A New Graph Grammar Formalism for Robust Syntactic Pattern Recognition

Peter Fletcher
School of Computer Science and Mathematics,
Keele University,
Keele,
Staffordshire, ST5 2BG, U.K.
E-mail: p.fletcher@keele.ac.uk

Please see also the accompanying paper *Mathematical Theory of Networks for Syntactic Pattern Recognition*.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

There are no conflicts of interest.

Abstract

I introduce a formalism for representing the syntax of recursively structured graph-like patterns. It does not use production rules, like a conventional graph grammar, but represents the syntactic structure in a more direct and declarative way. The grammar and the pattern are both represented as networks, and parsing is seen as the construction of a homomorphism from the pattern to the grammar. The grammars can represent iterative, hierarchical and nested recursive structure in more than one dimension.

This supports a highly parallel style of parsing, in which all aspects of pattern recognition (feature detection, segmentation, parsing, filling in missing symbols, top-down and bottom-up inference) are integrated into a single process, to exploit the synergy between them.

The emphasis of this paper is on underlying theoretical issues, but I also give some example runs to illustrate the error-tolerant parsing of complex recursively structured patterns of 50–1000 symbols, involving variability in geometric relationships, blurry and indistinct symbols, overlapping symbols, cluttered images, and erased patches.

Keywords

graph grammar, syntactic pattern recognition, graph parsing, error-tolerant parsing

1. Introduction

The ability to represent and process recursive symbol structures is often seen as fundamental to cognition, and its characteristic features of productivity, systematicity and compositionality have been especially celebrated by the 'classical' school of cognitive science [24,51]; and yet such symbolic processing is often *brittle*, i.e., unable to cope gracefully with noisy input, random variation, contradictory or incomplete data, and unexpected input. In contrast, nonsymbolic or subsymbolic methods [96], such as statistical pattern recognition and neural networks, are much more robust but have great difficulty representing recursively structured data. In the 1980s and 1990s this contrast was drawn in adversarial terms [51,53], but since then much effort has been devoted to reconciling the two by implementing symbolic processing in subsymbolic architectures [61,87,20].

I approach the problem from the other direction: the aim of my work is to make symbolic processing more robust. I choose as a suitable problem domain *geometric pattern recognition*, the problem of recognising structured spatial configurations. This is not computer vision: we are not concerned with perspective, binocular vision, motion, occlusion, colour, texture, shadow or illumination. We are concerned with issues that are common to pattern recognition in general, not any particular sense modality. The raw data from which the pattern is recognised I shall call the *image*, for want of a better word, although I intend it to be thought of as a spatial domain, not necessarily a visual one; where necessary I shall assume that it is two-dimensional.

Formal grammars will be used to specify the recursive structure of a class of patterns. By a *grammar* I mean any formal system that specifies how patterns can be composed out of parts, allowing for iteration and nested recursive structure; it may be expressed in the form of a set of production rules, a transition diagram, an accepting automaton, or by any other method. Since we are not limited to sequential patterns we need graph grammars rather than string grammars. A *graph grammar* is a formalism for representing the syntactic structure of systems of symbols connected by binary relations. The conventional type of graph grammar generates patterns by repeatedly applying production rules; I shall review these grammars and their difficulties in §2 and shall instead introduce a different kind of formalism that rejects production rules and represents syntactic structure directly and declaratively using *networks*, with parsing seen as a homomorphism between networks.

We begin by making a basic distinction between tokens and types. A symbol *token* is an occurrence of a symbol somewhere in the image; every symbol token has a *type*. E.g., a page of text may consist of hundreds of letter *tokens*, each with its own position, size, orientation, and relations to nearby letter tokens, but there is only ever a fixed set of 52 letter *types*, which have no position. The system of symbol tokens present in an image (with all their relationships) is called a *pattern*; the *grammar* is a system of symbol types, specifying the permitted structural and geometric relationships in patterns.

The grammars should be capable of representing complex patterns (of up to 1000 symbol tokens) related by iteration (not just in one dimension but two-dimensional iterations such as square grids), hierarchical structure, and nested recursive structure. Most syntactic pattern recognition research uses extremely limited grammars: e.g., a fixed compositional hierarchy (possibly allowing for absent parts) [68]; or AND-OR trees [81]; or special-purpose representations designed for a particular application, such as recognising the facades of buildings [93];

limited types of easy-to-parse graph grammars are used by [47,95]; specialised grammars are used for music notation [42,83] and for mathematical formulae [66,64,15,22,104].

The recognition algorithm must be able to tolerate all kinds of error and variability:

- variability in the geometric relationships between neighbouring symbol tokens, or between parts and wholes;
- blurry and indistinct symbol tokens, and cluttered images where the lowest-level symbol tokens cannot be detected purely by bottom-up processing;
- overlapping symbol tokens (where the image cannot be segmented by tracing boundaries or finding connected components);
- missing patches in the image, small blank regions where the recognition algorithm must fill in the missing symbol tokens or parts of symbol tokens using top-down information.

These requirements are extremely challenging for any existing pattern recognition algorithm. In my view the brittleness of syntactic pattern recognition is caused by the imposition of an unwarranted sequential order on what is conceptually a parallel process. In the first place, many graph parsing algorithms impose a sequential order on the graph and then traverse it in that order (see [48, §2] for a survey); a single error at an early stage may cause the whole parse to go astray. My algorithm will examine all parts of the image in parallel. Secondly, recognition should not involve taking early irrevocable decisions: the algorithm should generate alternative interpretations of (parts of) the image, which are then extended and combined in parallel, with all but one eliminated by the end. Thirdly, most recognition algorithms subdivide the task into a fixed sequence of phases. Recognising a pattern requires decisions on how the image is segmented into symbol tokens, the type of each symbol token present, which parts of the image are pixel noise (i.e., not part of any symbol token), the structural relationships between the symbol tokens, and the position, orientation, size and degree of stretching and shearing of each symbol token. It is widely recognised that these tasks are inter-dependent and that the synergy between them should be exploited. Tu *et al* [98] point out the need for segmentation, object detection and recognition, top-down parsing, and bottom-up parsing to work together; Tombre [97] argues that segmentation, recognition and higher-level semantic analysis need to interact; Casey & Lecolinet [21], in a survey of character segmentation, point out the inter-dependence of decisions on segmentation, shape similarity and contextual acceptability; see also [82,59]. Yet, despite all this, algorithms still separate these tasks and perform them in sequence. Many image recognition algorithms begin with a special segmentation method to identify the foreground object, find its bounding box, and transform it to a standard position, before the main parsing (identification of parts) begins, and there is commonly postprocessing after parsing [78,82]. For highly specialised applications, where the structure present is highly predictable, such as recognition of music scores, a top-down approach may be successful [83]. Other algorithms identify the lowest-level features first, constructing a graph from them, and then parse the graph; for example, Celik & Yanikoglu's mathematical formula recognition system [22] achieves very low recognition accuracy because any failure to recognise symbol tokens at the OCR stage cannot be corrected at the parsing stage. As Flasiński *et al.* emphasise [46], the conversion of the numeric input representation into a symbolic representation in terms of pattern primitives can lose essential information. Better results can be achieved if the recognition of pattern primitives can be guided by grammatical information from the higher-level stages [104]. Few algorithms can cope with overlapping objects, because bottom-up approaches to segmentation are inapplicable (a partial exception is [81]).

In my algorithm there is no division into phases; all aspects of recognition are integrated. During the recognition process the pattern is built up incrementally; symbol tokens are created, linked to other symbol tokens, merged, split and pruned; many rival pattern-fragments are in play simultaneously, competing to attract parts and to grow at one another's expense. Bottom-up and top-down influences apply simultaneously; indistinct, missing or overlapping symbol tokens can be 'hallucinated' using grammatical context.

The symbol tokens are arranged in spatial configurations, so some mathematical formalism is needed for representing geometric relationships and how they are permitted to vary in a pattern. In §5 I shall review the range of representational methods used and propose a more general concept of a *fleximap*, an elastic affine transformation relating one symbol token to another. The recognition process should be invariant under affine transformations, to accommodate different points of view on the same pattern; most existing invariant pattern recognition techniques assume that the image has already been segmented and apply only to a single object (with little or no occlusion).

Finally, the recognition process must also cope with symmetry; this is a neglected topic in the literature. A line segment has two-fold symmetry; the algorithm must understand that it may occur either way round as part of a complex symbol token and that the two ways are equivalent. A hexagonal grid has six-fold symmetry; the algorithm must recognise the six symmetry variants of a pattern as equivalent, not as rival interpretations. The symmetries of each symbol type must therefore be represented explicitly in the grammar.

This paper builds on my previous work [48], where regular graph grammars describing two-dimensional geometric patterns were parsed and learned from positive examples. Here I am using more powerful grammars, using the full geometry of the plane rather than a discrete grid, and adding error-tolerance. However, this paper considers only recognition of patterns, not learning of grammars.

The emphasis of this paper is on the new grammatical formalism. For reasons of space I give only a summary of the mathematical theory; see [49] and [50] for full details and proofs.

Section 2 reviews graph grammars in general and my reasons for being dissatisfied with the conventional approach based on production rules.

My new type of graph grammar is introduced informally in §3 and formally in §4, with the spatial aspects covered in §5. The problem of pattern recognition is specified in §6. The algorithm for recognition is developed in §§7–9. The core parts of the theory are provably correct, but peripheral parts rely on heuristic arguments (particularly in §7.4).

Three example grammars are constructed in §10, illustrating various types of iteration and recursion. Example runs with these three grammars (combined into a single grammar) are described in §11. These runs are intended purely as proof of concept and are not experiments: the emphasis of this paper is on investigating fundamental qualitative issues underlying all pattern recognition, in particular the difficulty of combining recursive representational capabilities with error-tolerance; I make no claims of applicability. Conclusions are drawn in §12.

2. Graph grammars

There is a well-established theory of grammars for representing 'strings' (sequences of symbols). String grammars contain *production rules* of the form $L \to R$, where L and R are strings, called the *left-hand side* and the *right-hand side* of the rule. Define a relation $\Rightarrow$ between strings as follows: for any strings S and T, $S \Rightarrow T$ iff S can be turned into T by replacing a substring matching L in S by R; in that case we say that the rule $L \to R$ has been *applied* to S to yield T. To *parse* a string S is to find a sequence of production rule applications $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \cdots \Rightarrow S$, which is called a *derivation* of S (where S_0 is the *start symbol*). The *language generated by* the grammar is the set of all strings S that have derivations.

To generalise all this to non-sequential patterns we replace the strings L and R by graphs, in which the nodes, and perhaps also the edges, bear labels. This gives us a *graph grammar*, which generates a language of graphs. (The node labels are symbol types and a labelled node in the generated graph is a symbol token.) However, several complications and perplexities arise in doing so:

(i) the complexity of the concept of applying a production rule, i.e, the $\Rightarrow$ relation;
(ii) the lack of a well-defined Chomsky hierarchy for graph grammars;
(iii) the difficulty of combining good expressive power with the existence of parse trees;
(iv) the imperspicuity of graph grammars;
(v) the intractability of parsing;
(vi) the difficulty of coping with noise, i.e., errors in the graph to be parsed.

I shall take these issues in turn.

2.1 The complexity of production rule application

The concept of 'applying' a production rule to a graph is not as obvious as in the string case. To apply the rule $L \to R$ to a graph G we must identify an isomorphic copy of L in G and replace it with an isomorphic copy of R, giving a new graph H. This may be expressed mathematically by graph endomorphisms (injective homomorphisms) $g{:}L \to G$ and $h{:}R \to H$. Thus $g(L)$ is removed from G and replaced by $h(R)$, giving H. But how exactly is $h(R)$ to be joined to $G \setminus g(L)$? This is called the *embedding problem*. There are three approaches in the literature. (I shall consider only *sequential* grammars, not *parallel* grammars, here.)

First approach: the context subgraph. In this method there is a subgraph of L that is shared with R, or in bijection with a subgraph of R; this subgraph is called K, the *context subgraph*. This can be expressed by endomorphisms $l{:}K \to L$ and $r{:}K \to R$. Often the bijection between the subgraphs $l(K)$ and $r(K)$ is expressed by marking the nodes of $l(K)$ and $r(K)$ with unique numeric labels. Then, when a rule is applied, $g(L)$ is removed from G and $h(R)$ is added to $G \setminus g(L)$ in such a way that $h(r(K))$ occupies the same position as its isomorphic copy $g(l(K))$ did before.

This can be expressed neatly in category-theoretic terms as a pair of push-outs [35,36]:

$$\begin{tikzcd} L \arrow[d, "g"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "d"] & R \arrow[d, "h"] \\ G & D \arrow[l, "c_1"'] \arrow[r, "c_2"] & H \end{tikzcd}$$

where all the arrows are graph endomorphisms. (Think of D as the part of G that is unchanged by the rule application (either because it is not involved in the operation or because it is the context part $g(l(K))$), and think of c_1, c_2 as inclusion endomorphisms.)

Given a rule $L \to R$ and a graph G, the rule can only be applied to G if a D exists that forms a push-out with K, L and G. This will be the case iff there are no edges in $G \setminus g(L)$ incident to nodes in $g(L \setminus l(K))$: this is called the *dangling-edge condition*.

In the most general formulation l, r, g, d, h, c_1, c_2 are not required to be injective, i.e., they are simply graph homomorphisms. In this case, the *identification condition* must also hold for the rule to be applicable to G: this states that whenever two nodes or edges x, x' in L have $g(x) = g(x')$ then x and x' are in $l(K)$.

(I have been assuming so far that the rule $L \to R$ is applied from left to right, i.e., that we interpret $G \Rightarrow H$ as meaning that G is transformed into H; but it is also possible to apply the rule from right to left, i.e., to regard it as transforming H into G; corresponding dangling-edge and identification conditions must be imposed to guarantee the existence of D forming a push-out with R, K and H.)

Examples of this approach are hyperedge replacement grammars [56,33]; positional grammars [25], which use plex-like structures with attaching points; Lichtblau's grammar [69]; reserved graph grammars [106] and variants of them such as spatial grammars [62].

Ferrucci *et al.* [44] consider this type of grammar restrictive, because the context subgraph K is fixed. The dangling-edge and identification conditions are also restrictive. There are some slight extensions that allow some variation in K. Layered graph grammars [84] and breeze graph grammars [67] allow the nodes in K to have 'wildcard' labels, each of which stands for one of a definite set of possible labels. In edge-based graph grammars [90,91], an edge in K may bear an asterisk, meaning that it represents any number of edges. In Blosetin's grammar [15] K may contain 'set nodes', each of which matches against as many nodes in G as possible. Other variants, which involve something similar to a context subgraph, are [101,102,25,88].

Second approach: connection relations. In this approach we still have the endomorphisms $g{:}L \to G$ and $h{:}R \to H$, but no context subgraph K. When $g(L)$ is removed from G, edges of $G \setminus g(L)$ that are incident to nodes of $g(L)$ must also be removed and replaced by new edges incident to nodes of $h(R)$. How this is done is specified by a *connection relation*.

The advantage of this method is that it gives more powerful grammars than the 'context subgraph' method – at least, this appears to be true in general and has been proved for particular classes of grammar [28].

A disadvantage is that descriptions of how the connection relation is used to connect the new edges are complicated [60,86] and often informal and incomplete [102,66]. This does not lend itself to a neat mathematical theory.

A second disadvantage is that application of a production rule may not be reversible, or not reversible in a unique way. In [4] the rules can be applied left-to-right or right-to-left, but the two directions are not inverses of each other. This is a problem for bottom-up parsing: the definition of a language involves left-to-right application, but bottom-up parsing involves right-to-left application.

The commonest type of grammar of this kind is the *node-replacement grammar* [41], in which the left-hand side L of each production rule is just a single node, n. Most commonly the nodes bear labels and the labels control the connection of new edges: this is called a *node-label controlled* (NLC) grammar [40]. The edges may bear labels too and may be directed. Rule application works as follows: $g(n)$ is replaced by $h(R)$, and if there is an edge e in G between $g(n)$ and a neighbour node u, it is replaced by new edges between $h(v)$ and u, for some nodes v

in R, depending on the labels of u and v (and e's label and direction if they exist), as directed by the *connection relation* that is provided with the grammar.

A slightly more general type of grammar is the *neighbourhood-controlled embedding* (NCE) grammar [41]. Here, the connection of new edges depends on the identity of the node v in R, not just its label. Relation grammars [43,44] are essentially equivalent to NCE grammars, presented in a different format. NCE grammars generate the same class of languages as NLC grammars, but that is not true when they are combined with some other extensions [40].

Alternatives to node-replacement grammars are *edge-label controlled grammars* (where the left-hand side L of a rule is one edge and its incident nodes) [75] and Adachi *et al.*'s context-sensitive NCE grammars, where L can be a graph of any size [4,5,3].

The most general type of production rule of this kind is Nagl's *set-theoretic approach* [79], in which new edges can be connected to $G \setminus g(L)$, not just to neighbour nodes of $g(L)$. A further generalisation is [14], where more general types of graph operation are used in place of production rules, the application of these operations being controlled by logical formulae and a tree grammar.

The 'context subgraph' and the 'connection relation' approaches should be seen as alternatives. Adachi *et al.* [4] use both in tandem, but in later work [5] abandon the context subgraph without affecting the generative power of the grammars.

Third approach: implicit edges. The problem we are considering is how to connect $h(R)$, the new part of the graph, to $G \setminus g(L)$, the old part, by edges. The simplest solution is to do without edges entirely. The graph G is simply a set of nodes, with geometric attributes such as position and size. The grammar contains 'constraints' between symbol types, which are essentially edges; if two nodes in G are related in a way that fits the constraints, there is implicitly a geometric relationship between them, but it is not represented explicitly as an edge. In the production rules L is a single node. The geometric attributes of the node in $g(L)$ are related to those of the nodes in $h(R)$; this means that new implicit geometric relationships are produced when the rule is applied, but there is no need for a context subgraph or a connection relation.

Examples of this approach are picture layout grammars [55] and constraint multiset grammars [23,76]. They are described in terms of 'multisets', but they are best thought of as graphs with implicit edges.

2.2 The lack of a well-defined Chomsky hierarchy

For string grammars there is a clear classification into levels according to generative power: type 0 (unrestricted grammars), type 1 (grammars where the left-hand side of each rule is no longer than the right-hand side, and parsing is guaranteed to halt), type 2 (context-free grammars, expressing iterative and nested structure), and type 3 (regular grammars, expressing iterative structure only). These types can be characterised in various ways (in terms of production rules, recognition automata, or alternative notations such as regular expressions), and are unaffected by minor differences in formulation.

Unfortunately the same cannot be said for graph grammars. One may still define type 0 grammars, having no restrictions on the production rules; there are inequivalent kinds of type 0 grammar, such as context-sensitive NCE grammars [3], edge-based graph grammars [90], reserved graph grammars [106], and the 'set-theoretic approach' [79]; reserved graph grammars are more expressive than edge-based grammars [109], whereas the set-theoretic approach seems to be the most expressive type. There are several concepts of type 1 grammar; the aim is always to impose a restriction that ensures that parsing is guaranteed to halt. The simplest restriction is that the number of nodes on the left-hand side of each rule is less than or equal to the number of nodes on the right-hand side [4]; or alternatively that the number of nodes and edges on the left is less than or equal to the number of nodes and edges on the right [35]. Layered graph grammars [84,3] impose a more complicated condition of the same kind, with node labels distinguished by 'layer'. Reserved graph grammars did the same at first [106], but more recent work drops the layering and reverts to the simpler condition [110].

There is a variety of inequivalent concepts of *context-free* grammar [39]. Some authors say that a context-free grammar is one where each rule has a single (labelled) node as left-hand side [60,106,102,91,52,72,25]. However Lange & Welzl [65] point out that this is not really context-free in the full sense: '"context-sensitivity" is hidden in the embedding mechanism of NLC grammars'. Hence context-freedom is often identified with a confluence or finite Church-Rosser property [16]. Definitions of this vary. The general idea is that 'the result of a derivation is independent of the order in which the productions are applied' [40]. One version is that if $G \Rightarrow_p H_1 \Rightarrow_q K_1$ and $G \Rightarrow_q H_2 \Rightarrow_p K_2$ (where '$\Rightarrow_p$' means 'derives by using production rule p') then K_1 is isomorphic to K_2. Another version is that if $G \Rightarrow_p H_1$ and $G \Rightarrow_q H_2$, where $p \neq q$, then $G \Rightarrow_p H_1 \Rightarrow_q K$ and $G \Rightarrow_q H_2 \Rightarrow_p K$ for some K. These are left-to-right versions; right-to-left versions can be defined by reversing the arrows.

Courcelle argues from a more abstract perspective that 'context-free' should be defined to mean confluent and associative [26].

These notions are heavily dependent on the surrounding formalism. In the 'set-theoretic approach' [79] and boundary NLC grammars [86], the context-free languages coincide with the context-sensitive ones, though this is not true for the grammars.

There are a few concepts of *regular* graph grammar in the literature [27,54,79,2,1,103,48]. These are heavily dependent on the details of the definition, especially on whether the right-hand side is allowed to be disconnected. There is no unique notion.

There has been little work to characterise classes of graph language by accepting automata. Some classes of NCE grammars (linear NCE and boundary NCE grammars) can be characterised by accepting automata, but this does not seem possible for arbitrary NCE grammars [17].

There has also been a little work on algebraic characterisations of classes of graph language. Courcelle [28] gives formal characterisations, using monadic second-order logic, of the set of languages generated by edge-labelled edge-directed NCE grammars and the set of languages generated by hyperedge-replacement grammars. Bauderon *et al.* [6,9] claim to provide an algebraic formulation of production rule application in NLC grammars as a pullback. But this cannot be true, as their graphs have no node labels. There is a kind of node labelling mechanism implicit in their 'unknown' homomorphisms [6, definition 3], but these labels are freely changed every time a production rule is applied, so this is not at all like what is normally understood as an NLC grammar. In their later work [7,8] an elaborate mechanism is specified for updating these labelling homomorphisms every time a rule is applied; this, along with the pullback, specifies how rule application works. There is also an element of parallel application of rules. It is now acknowledged that this generates a larger class of languages than node-replacement grammars.

2.3 Perspicuity, parse trees, expressive power

In a context-free string grammar the production rules (e.g., *Sentence* $\rightarrow$ *NP verb NP*) express something about the structural possibilities of the language. After parsing a sentence, a parse tree is produced, which makes explicit the grammatical structure of the sentence. It is the central function of grammars to do this. The parse tree provides a declarative view of the structure of the sentence, abstracting away from the operational details of the derivation. A parse tree may also be thought of as an algebraic expression that is evaluated to produce the sentence [34]. Unfortunately, with more complex types of grammar, including most graph grammars, the production rules are merely engines for generating sentences and do not convey any grammatical information. Parse trees are often not possible, so there is no representation of the grammaticality of the sentence more abstract than the sequence of production rule applications.

When is a parse tree possible in a graph grammar? If the production rules have a single node on the left-hand side then one may construct a tree of the nodes involved in the derivation; however, the edges will cut across between the branches in a tangle [55,29]. Useful parse trees can be produced if the grammar is confluent. This is done for hyperedge-replacement grammars [33], constraint multiset grammars [23], and relation grammars [43,44]. In [60] no confluence condition is imposed and the parse trees are dependent on the order of application of the production rules, so they do not express the syntactic structure in a fully abstract way. Parse trees are also possible for the graph expansion grammars of [34], which are essentially an extension of hyperedge replacement grammars.

Sometimes a nested graphical representation of syntactic structure is used, which is equivalent to a parse tree [102,17]. I believe this only works well if the grammar is confluent.

Thus it seems that parse trees are possible only for context-free graph grammars. Unfortunately, these grammars are surprisingly weak in their expressive power. One would expect context-free grammars to be able to represent iterative and nested structure, but, as Zou *et al.* say, 'context-free graph grammars have difficulty in specifying a large portion of graphical VPLs [visual programming languages]' [110]. Even in the weakest sense of context-free, 'Not all, nor even the majority, of VL [visual language] formalisms are context-free even in this loose sense' [102]. Even an iterative pattern in two dimensions, which one would expect to be captured by a regular graph grammar, is beyond the reach of context-free graph grammars: Schuster [89,16] showed that the set of quadratic graphs (rectangular grids) cannot be generated by any graph grammar with the finite Church-Rosser property.

Bauderon *et al.*'s pullback grammars can generate the set of square grids [7], but this is a very much more powerful type of grammar, with an elaborate mechanism for changing labels at every derivation step, and depending essentially on parallel application of rules; so the rules themselves do not fully express the grammatical knowledge.

This is the key limitation of graph grammars: to obtain decent expressive power we have to go beyond the context-free grammars, but then we lose the ability to represent syntactic structure through parse trees. Thus we have a purely operational concept of what it means to belong to the language. There is no description of how the graph fits the grammar simpler than the sequence of production rule applications.

2.4 Parsing algorithms

Graph grammars are very powerful mechanisms, and so parsing is necessarily intractable. Type 0 grammars in the 'set-theoretic approach' [79] and the 'algebraic approach' [35] and for edge-label controlled grammars [75] can generate any recursively enumerable set of graphs, so the membership problem (deciding whether a given graph is in the language) is undecidable.

Type 1 restrictions, which all amount to saying that the left-hand side is no larger than the right-hand side, make the membership problem decidable [35].

Even in the case of context-free grammars the membership problem is still intractable. NLC grammars can generate PSPACE-complete graph languages [16]. Hyperedge-replacement grammars [33], linear edge-replacement grammars [33], confluent edge-labelled edge-directed NCE graph grammars [41], and even regular NLC grammars [1] can generate NP-complete graph languages. This is unlike context-free string grammars, which can be parsed in cubic time using the standard Cocke-Younger-Kasami algorithm. The reason is that a string has quadratically many substrings, whereas a graph has exponentially many subgraphs [13].

Some improvement in efficiency is possible by clever optimisation to reduce backtracking [111]; the worst-case time-complexity however remains exponential in the size of the input graph, for a fixed grammar. Hence much research effort is devoted to identifying subclasses of grammar that can be efficiently parsed. Polynomial-time parsing is possible if

- the grammar has the finite Church-Rosser property,
- the grammar generates only connected graphs,
- the grammar generates graphs of bounded degree.

If any of these three conditions is dropped then a grammar exists for which the membership problem is NP-hard [16]. Note however that even where parsing is possible in polynomial time the degree of the polynomial is very high, so tailor-made efficient algorithms are still needed for particular classes of grammar [86,43].

If the grammar is confluent in the right-to-left direction then bottom-up parsing without backtracking, called *selection-free* parsing, is possible; this gives polynomial time complexity [110,106,72]. This is a very restrictive condition, however.

Precedence graph grammars can be parsed in linear time [35]. Graph languages generated by evaluating algebraic expressions generated by tree grammars can parsed in polynomial time if certain restrictions are imposed [14,12]. Other fast parsing algorithms for restricted classes of graph grammar are [31,32,13,54,103].

2.5 Coping with noise

Graph parsing algorithms have little ability to cope with errors in the graph (false or missing nodes, and errors in labels, edges or structure).

Many graph parsing algorithms are based on exhaustive enumeration of all possibilities, either by backtracking or by carrying forward a set of possible parses, e.g., algorithms based on the Cocke-Younger-Kasami algorithm [86,55,16] or on the Earley algorithm [43,101,18,19,69]. With noise the possibilities proliferate and become unmanageable.

Most algorithms work by traversing the graph, i.e., by visiting each node or edge one at a time [69,1,25,101]. This is unsuitable for non-Eulerian graphs and is error-intolerant, since an error in the first node visited may throw the process onto the wrong track, and the erroneous node cannot be corrected later because it is never revisited.

In addition, many pattern recognition systems work in a fixed sequence of phases, or in a fixed direction (top-down or bottom-up), where misinterpretations made in one phase cannot be corrected in later phases.

Some systems have a limited ability to handle simple types of error. *Error-correcting grammars* can, in principle, model the errors using production rules and make the correction of error part of the parsing process; however, these have only been used for the simplest sorts of error. Skomorowski's grammar [95] can correct errors in the label of a node or edge, but not structural errors. In Fahmy & Blostein's music recognition system [42], the symbols can have ambiguous labels; one node is created for each possible value of the label. Thus labelling errors (and stray dots) can be corrected. Liu & Yang's [72] production rules include 'uncertain edges', which are excess edges that it is anticipated may occur in the graph. If such edges occur they are removed during bottom-up parsing. Sánchez *et al.*'s grammar [88] recognises tessellations of rectangles and octagons. There are production rules that add special 'inserted' nodes where rectangles or octagons are expected at the boundary of the tessellation; at the end of parsing other production rules either remove each 'inserted' node or relabel it as a 'cut' symbol or a 'split' symbol. Mas *et al.*'s grammar [76] recognises hand-drawn diagrams; it includes production rules for eliminating errors where one stroke has been misconstrued as two.

Error-correcting grammars are only suitable for simple, explicitly anticipated types of error. It would not be feasible to use them for general error correction, where any symbol or relation may be spurious, missing, mislabelled, misinterpreted as two, or conflated with another.

Instead of modelling the errors in the grammar, an alternative is an error-tolerant parser, which finds the closest match between the graph and what the grammar would accept (for example [74], which matches a tree against a tree automaton). There are also error-tolerant parsing algorithms for matching a graph against a fixed set of graphs (rather than a grammar) by building up partial matches or by using edit distance [77,85].

3. Networks and homomorphisms

3.1 Requirements

From the literature survey in the previous section I draw the following conclusions for my own work.

- I require a grammatical formalism that can represent iterative and nested structure (including iterations in two dimensions such as tessellations).
- The syntactic structure in an image should be represented explicitly and declaratively, in a structure called a *pattern*, like the way a parse tree represents the structure of a string, rather than by a sequence of production rule applications. I shall do this by abandoning production rules altogether. The grammar and the pattern will each be a graph-like structure called a *network*, and parsing will be a matter of constructing a homomorphism from the pattern to the grammar.
- The grammatical formalism should be amenable to mathematical treatment. There should be no complicated semi-formal prose definitions. All definitions will be expressed algebraically.

- All phases of pattern recognition should be simultaneous and interdependent. Pattern recognition involves both constructing the pattern from the given image and parsing the pattern. Top-down and bottom-up parsing occur simultaneously. We break free completely from sequential ideas drawn from string parsing: there is no concept of traversal. Error-tolerant parsing must work opportunistically, starting with the most clear-cut parts of the image and recognising the ambiguous parts later in the light of top-down evidence. Many alternative interpretations of the image will be held in the same pattern and worked on in parallel, with one interpretation chosen by the end of parsing.

In this section I am considering only the structural aspects of the pattern: the geometric relationships will be considered in §5.

3.2 Networks

Consider the example in figure 1. The raw data is called the *image*. The pattern recognition system recognises the presence of one symbol token, called $A1$, of type A, and three symbol tokens, $line1$, $line2$, $line3$, of type $line$, which are its parts.

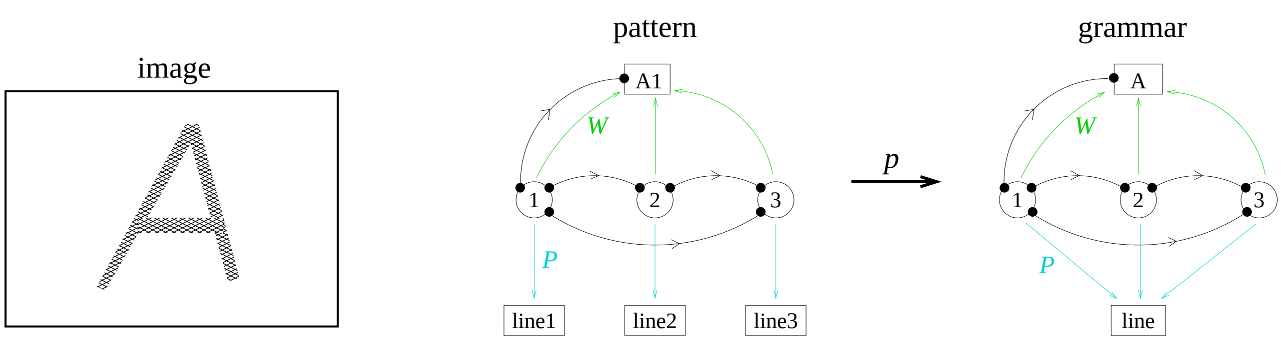


Figure 1. An image, the corresponding pattern, and the grammar against which it is parsed. Rectangles are symbols, circles are nodes, small filled discs are hooks, and black curves are edges. The W and P functions are depicted by green and blue arrows.

The symbol tokens are connected together to make the pattern. Part-whole relationships are represented by *node tokens*. Whenever one symbol token, e.g., $line1$, is a part of another, $A1$, there is a node n between them. Two functions W, P represent the relationship: $W(n) = A1$ and $P(n) = line1$. The node represents the role that the part plays in the whole; for example, node 1 = left stroke, node 2 = cross-bar, node 3 = right stroke. (In all my figures, symbols are depicted by rectangles and nodes by circles; the W and P functions are depicted by green and blue arrows.) When two parts of a whole are related to each other they are called *siblings*, and their relationship is represented by an *edge token* between the nodes. For example, the left stroke of an A must be joined at its top to the right stroke, and so there is an edge token between node 1 and node 3. In this example all three parts are one another's siblings. (In the figures edge tokens are depicted by black curves, sometimes directed.) Actually, an edge token does not run between two node tokens but between two *hook tokens*, which are attached to node tokens. (Hook tokens are depicted in the figures as small filled discs.) Note that there is also an edge token between node token 1 and the symbol token $A1$.

The grammar is similarly constructed, except that it is composed of symbol types, node types, hook types and edge types. Parsing consists of constructing a homomorphism p from the pattern to the grammar, under which each symbol, node, hook and edge token maps to its type, preserving incidence relations. The entire recognition process is, given an image and a grammar, to construct the pattern and the homomorphism.

This illustrates the main features of networks, albeit in an uninteresting example. Figure 2 shows an example with iteration (the image is omitted). An *alkane* is a hydrocarbon molecule consisting of a carbon chain of arbitrary length surrounded by hydrogen atoms. Propane is the case where there are three carbon atoms. Diagrammatic conventions are as before; under p, each symbol token or node token maps to the symbol type or node type with the same alphanumeric label, except that *propane* maps to *alkane*. (These symbol and node labels are just a diagrammatic convention for conveying how p works; they are not really there in the network.)

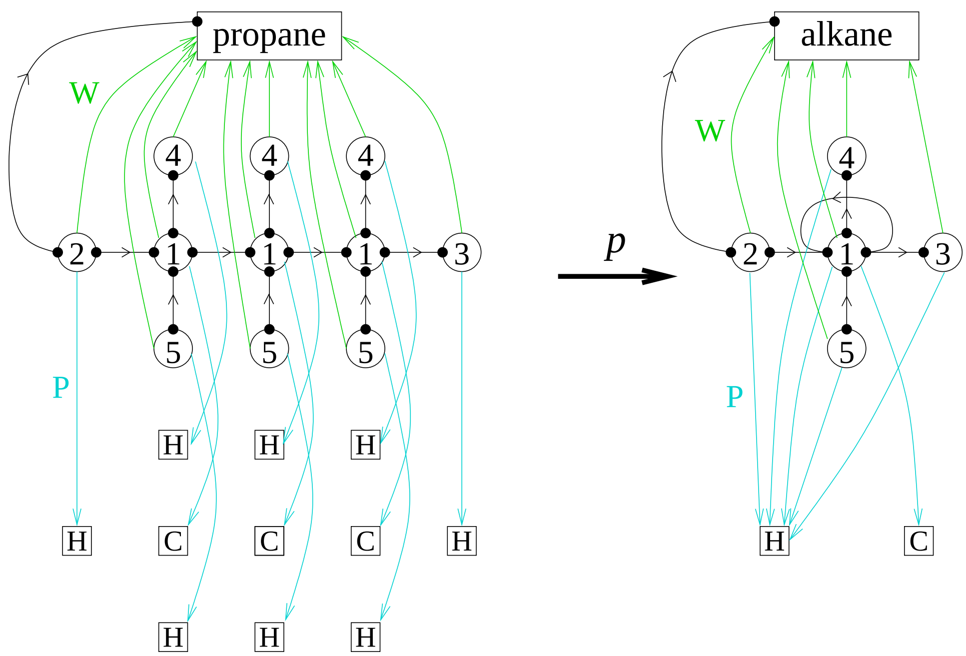


Figure 2. On the right, the grammar for alkanes. On the left, one possible alkane pattern.

There is a rule that each hook token in the pattern must have exactly one incident edge token. As a consequence, the *propane* token must be connected to one ‘2’ node token, which must be connected to one ‘1’ node token. Each ‘1’ node token must be connected to one ‘4’ and one ‘5’ token, and to either a ‘2’ or a ‘1’ token (by its left hook), and to either a ‘3’ or a ‘1’ token (by its right hook). This permits unbounded iteration of ‘1’ node tokens, and hence any number of carbon atoms. Thus, in the grammar, a hook with several incident edges represents a one-from-n choice, and allows for grammatical alternatives and iteration, which can be in one or more dimensions.

During the recognition process, while the pattern is under construction, there may be any number of edge tokens incident to a hook token (either because no edge token has been added yet or because several alternatives are being considered in parallel). Thus alternative interpretations for various parts of the image may be entertained in parallel, in the same pattern, without being confused. By the end, one incident edge token must be chosen for every hook token.

3.3 Subsymbols

One further type of grammatical constraint is needed. Consider figure 3, which shows (a) a hexagonal grid image, (b) part of the pattern corresponding to the portion of the image encircled in red, with symbol tokens (the grid itself and the component lines) omitted, and (c) the relevant portion of the grammar. (This is a simplified version of the hexagonal grid grammar of §10.1.) As usual, each node and edge in the pattern is mapped under p to the node and edge with the same label in the grammar.

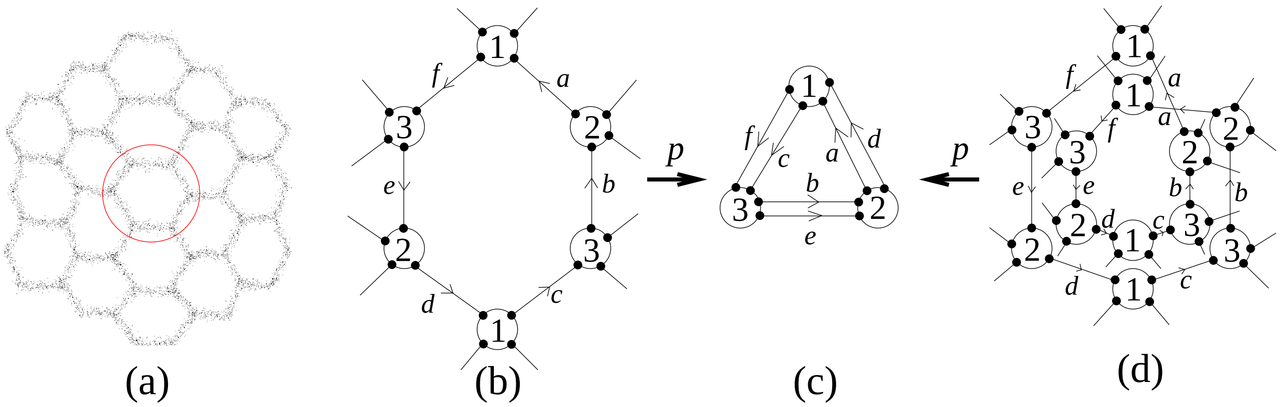


Figure 3. (a) An image. (b) Part of the pattern (without symbol tokens). (c) The relevant part of the grammar. (d) An unwanted pattern.

There is a problem with this. The grammar appears to allow 'nonstandard models', unintended patterns that could map homomorphically to the grammar just as well (see (d)). To cure this problem we need to treat a hexagon (i.e., the portion of the hexagonal grid shown in figure 3(b)) as a symbol in its own right. The hexagon is said to be a *subsymbol* of the hexagonal grid, meaning that all the parts of the hexagon (the six line symbol tokens) are parts of the hexagonal grid. Figure 4 shows, in part (a), the relevant portions of the pattern: the $hexagon1$ symbol token and all its nodes, hooks and edges; the $hex1$ hexagonal grid symbol token (only the relevant nodes, hooks and edges are shown); and the six parts of $hexagon1$, the symbol tokens $line1$–$line6$, which are also parts of $hex1$. Part (b) of the figure shows the grammar. As usual, under the parsing homomorphism p, each node and edge token in the pattern maps to the node or edge type with the same label in the grammar.

To indicate the subsymbol relationship, $hexagon1$ is 'glued' to $hex1$ by a gluing relation G; formally we write $G(hex1, hexagon1)$. We say $hexagon1$ is a *subsymbol* of $hex1$ and $hex1$ is a *supersymbol* of $hexagon1$. The corresponding nodes are also glued: $G(1,1'), G(1,1''), G(2,2')$, $G(2,2''), G(3,3'), G(3,3'')$. We say node $1'$ is a *subnode* of node 1 and node 1 is a *supernode* of node $1'$. Corresponding hooks and edges are also glued: $G(a,a'), G(b,b'), \ldots$. This is done in both the pattern and the grammar; the gluings in the pattern must match those in the grammar. Some of the gluings are shown in red in the figure.

One final syntactic feature is needed. The grammar must specify where *hexagon* subsymbols should occur in a *hex* grid. This is done by giving some of the edge types one or more *facets* (these are depicted as small red crosses in the figure). Some of the facets in the grammar are glued together in pairs. The edge tokens in the pattern have facets similarly. There is a new grammatical rule: in the pattern, each facet must be glued to *one* other facet

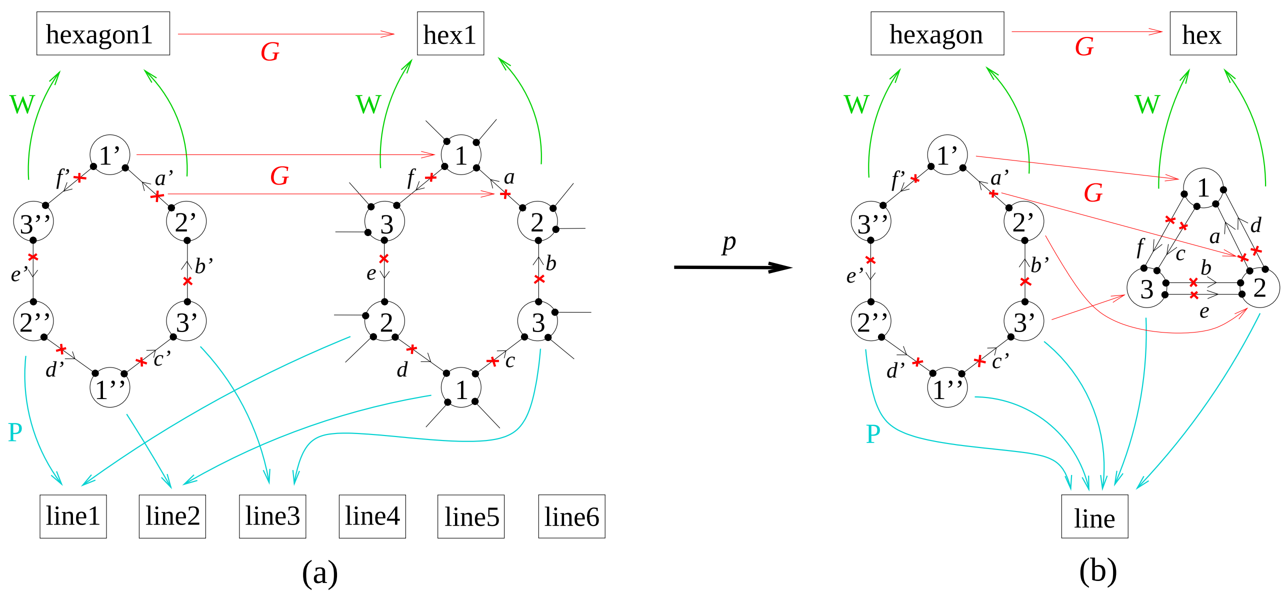


Figure 4.(a) Part of a pattern: a hexagonal grid symbol token *hex*1 and some its nodes, hooks, edges; and a subsymbol *hexagon*1, with all its nodes, hooks and edges and parts *line*1–*line*6. (b) The grammar, including the hexagonal grid symbol type *hex* and its subsymbol type *hexagon*. Only a few of the W and P arrows are shown. The gluing relation G is shown in red (only a few of the gluings are shown). Facets are shown as small red crosses.

(in a way consistent with the grammar). When two facets are glued this forces their edges to be glued, and hence their incident hooks are glued, hence the nodes to which they are attached are glued, hence the whole symbol tokens above them are glued.

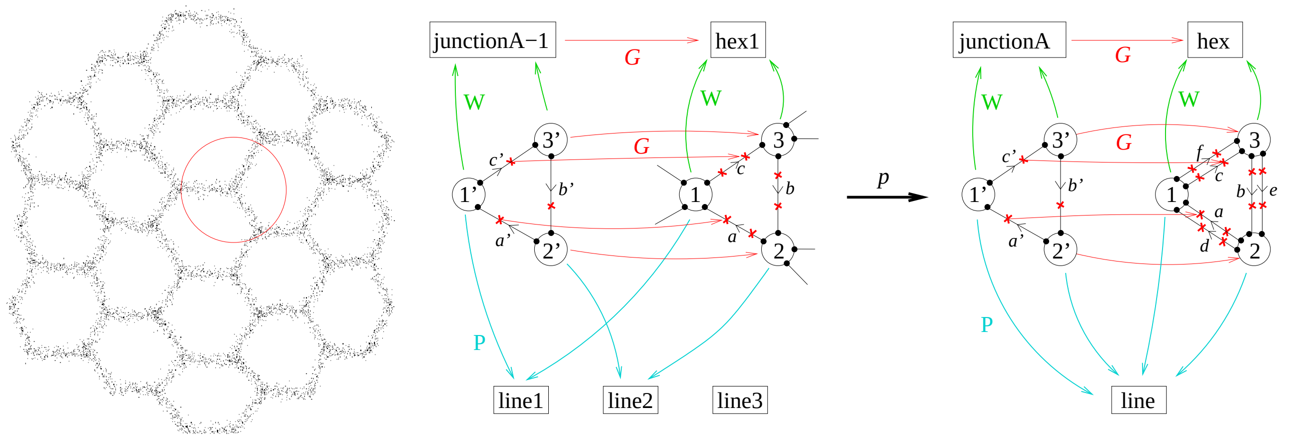
As another example, the three lines meeting at a vertex or junction of the hexagonal grid also form a subsymbol. There are two sorts of junction subsymbol: those in which the three lines are oriented towards the junction and those in which the three lines are oriented away from the junction, so we need two subsymbol types, *junctionA* and *junctionB*. Figure 5 shows the *junctionA* subsymbol type and one token *junctionA*-1 of that type. We give the edges a–f in the grammar and pattern a second facet: this means that each such edge will be part of two subsymbols: a junction and a hexagon. The *junctionA* edges a', b', c' only need one facet. As before, in the pattern each facet must be glued to one other facet (in a way consistent with the grammar). This will force edge tokens a', b', c' to be glued to a, b, c, respectively, in the pattern. This will force their incident hook tokens to be glued; this will force the node tokens $1', 2', 3'$ to be glued to $1, 2, 3$; and so *junctionA*-1 will be glued to *hex*1.

Further example grammars are given in §10. In the next section the concepts of this section will be defined formally.

4. The structural part of the theory

4.1 Preliminaries

We shall deal with sets, functions and binary relations. I use standard set and function notation.



(a) (b) (c)

Figure 5. (a) The image, with the relevant portion encircled. (b) The pattern: the *junctionA*-1 subsymbol token and the relevant portion of the *hex*1 symbol token. (c) The grammar: the *junctionA* and *hex* symbol types. A few of the P and W arrows and a few of the gluings are shown. Every node or edge with a primed label is glued to the one with the unprimed label. Every node or edge in the pattern is mapped under p to the one with the same label in the grammar.

Definition. The *domain* dom(R) and *range* ran(R) of a relation R are defined by

$$\mathrm{dom}(R) = \{\, a \mid \exists b\ R(b,a)\,\}, \qquad \mathrm{ran}(R) = \{\, b \mid \exists a\ R(b,a)\,\}.$$

Definition. A relation R is said to be *on* a set A iff $\mathrm{dom}(R) \subseteq A$ and $\mathrm{ran}(R) \subseteq A$.

Definition. For any set A, the *identity* relation id_A is defined by

$$\forall a, b \quad (id_A(a,b) \iff a = b \in A).$$

Definition. The empty relation $\perp$ is defined by $\forall a, b\ \neg \perp(a,b)$.

Definition. For any relations R and S,

$$\begin{aligned} R = S &\iff \forall a,b\ (R(a,b) \Leftrightarrow S(a,b)) \\ R \subseteq S &\iff \forall a,b\ (R(a,b) \Rightarrow S(a,b)) \\ R \subset S &\iff R \subseteq S \wedge R \neq S. \end{aligned}$$

Definition. For any relations R and S, the relations $R \cap S$, $R \cup S$ and $R \backslash S$ are defined by

$$\forall a,b \begin{cases} (R \cap S)(a,b) \Leftrightarrow R(a,b) \wedge S(a,b), \\ (R \cup S)(a,b) \Leftrightarrow R(a,b) \vee S(a,b), \\ \ (R \backslash S)(a,b) \Leftrightarrow R(a,b) \wedge \neg S(a,b). \end{cases}$$

Definition. For any relations R and S, the *composed* relation $R \circ S$ is defined by

$$\forall a,c \quad ((R \circ S)(a,c) \iff \exists b\ (R(a,b) \wedge S(b,c))).$$

Definition. For any relation R, the *inverse* relation R^{-1} is defined by

$$\forall a,b \ \ (R^{-1}(b,a) \ \Leftrightarrow \ R(a,b)).$$

Definition. The *graph* of a function f is the relation $\overline{f}$ defined by

$$\forall a,b \ \ (\overline{f}(b,a) \ \Leftrightarrow \ a \in \mathrm{dom}(f) \wedge b = f(a)).$$

Note that, for any f,g, $\overline{f \circ g} = \overline{f} \circ \overline{g}$.

Definition. A relation R is *finite* iff there are finitely many pairs (a,b) such that $R(a,b)$ holds.

Definition. The 'not-equal-to' relation NE is defined by $NE(a,b) \Leftrightarrow a \neq b$.

Definition. A finite relation R is *acyclic* iff

$$\neg\, \exists R^* \ (\bot \neq R^* \subseteq R \ \wedge \ R^* \subseteq (R^* \circ NE \cap NE \circ R^*)).$$

(Informally, R is acyclic iff every non-empty subrelation R^* has an element of valency 1, i.e., an element related to just one element by R^* or to just one element by R^{*-1}. This is equivalent to the non-existence of a finite cyclic sequence $x_1, \ldots x_n, x_1$, with n even and $n > 2$, where the terms are all different and are related by $x_1 \xrightarrow{R} x_2 \xleftarrow{R} x_3 \xrightarrow{R} x_4 \xleftarrow{R} x_5 \xrightarrow{R} \cdots \xrightarrow{R} x_n \xleftarrow{R} x_1$.)

Definition. If $f{:}X \to Y$ and R is a relation on X then R is *connected* relative to f iff, for any set Z and function $g{:}X \to Z$ such that $\overline{g} \circ R \subseteq \overline{g}$, there exists a function $i{:}f(X) \to Z$ such that $i \circ f = g$.

(Informally, R is connected relative to f iff, for any $x,x' \in X$ such that $f(x) = f(x')$, there exists a finite sequence $x = x_0, x_1, \ldots x_n = x'$ such that, for each $i \in \{1, \ldots n\}$, $R(x_{i-1}, x_i)$ or $R(x_i, x_{i-1})$ holds.)

Definition. If $f{:}X \to Y$ and R is a relation on X then R is *minimal* relative to f iff

$$\forall R^* \subseteq R \ \ (\overline{f} \circ R^* = \overline{f} \circ R \Rightarrow R^* = R).$$

(Informally, R is minimal relative to f iff, for any $x \in X$ and $y \in Y$, there exists at most one $x' \in X$ such that $f(x') = y$ and $R(x', x)$.)

I shall also use some basic concepts of category theory, namely pullbacks, sums and coequalisers.

4.2 Networks and homomorphisms

Definition. A *network* is a 12-tuple $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, where Σ, N, H, E, K are disjoint finite sets; $W{:}N \cup \Sigma \to \Sigma$, $P{:}N \to \Sigma$, $A{:}H \to N \cup \Sigma$, $F,S{:}E \to H$ and $C{:}K \to E$ are functions such that $\forall \sigma \in \Sigma \ W(\sigma) = \sigma$; and G is a relation on $\Sigma \cup N \cup H \cup E \cup K$ such that

(1) $id_\Sigma \circ G = G \circ id_\Sigma$, $id_N \circ G = G \circ id_N$, $id_H \circ G = G \circ id_H$, $id_E \circ G = G \circ id_E$, $id_K \circ G = G \circ id_K$;
(2) $\overline{W} \circ G = G \circ \overline{W}$, $\overline{P} \circ G \subseteq \overline{P}$, $\overline{A} \circ G \subseteq G \circ \overline{A}$, $\overline{F} \circ G \subseteq G \circ \overline{F}$, $\overline{S} \circ G \subseteq G \circ \overline{S}$, $\overline{C} \circ G \subseteq G \circ \overline{C}$;
(3) G_H and G_H^{-1} are minimal relative to A; G_K and G_K^{-1} are minimal relative to C; G_E and G_E^{-1} are minimal relative to F and S (where $G_H = G \circ id_H$, $G_K = G \circ id_K$ and $G_E = G \circ id_E$);
(4) $G \circ G = \bot$.

The elements of Σ, N, H, E, K are called *symbols*, *nodes*, *hooks*, *edges* and *facets*, respectively. G is called the *gluing relation*; $G(x,y)$ means that y is a subsymbol, subnode, subhook, subedge or subfacet of x, i.e., x is a supersymbol, supernode, superhook, superedge or superfacet of y. The functions W, P, A, F, S, C express the incidence relations: a node n connects a part $P(n)$ to a whole $W(n)$; a hook h is attached to a node (or possibly a symbol) $A(h)$; an edge e runs from its *first* hook $F(e)$ to its *second* hook $S(e)$; a facet k belongs to the edge $C(k)$.

Condition (1) means the gluing relation G may be considered as the disjoint union of a relation $G_\Sigma = G \circ id_\Sigma$ on symbols, a relation $G_N = G \circ id_N$ on nodes, a relation $G_H = G \circ id_H$ on hooks, a relation $G_E = G \circ id_E$ on edges, and a relation $G_K = G \circ id_K$ on facets.

Condition (2) says that G preserves incidence: e.g., if a hook h_1 is glued to a hook h_2 then the node $A(h_1)$ is glued to the node $A(h_2)$.

Condition (3) means that a hook is glued to at most one hook of any given node; a facet is glued to at most one facet of any given edge; and an edge is glued to at most one edge of any given hook (for each edge direction).

Condition (4) says that a subsymbol, subnode, subhook, subedge or subfacet cannot be also be a supersymbol, supernode, superhook, superedge or superfacet.

Definition. If $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ and $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ are networks, a *homomorphism* $p{:}\,\mathcal{N}_1 \to \mathcal{N}_0$ is a function from $\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1$ to $\Sigma_0 \cup N_0 \cup H_0 \cup E_0 \cup K_0$ such that

(1) $p(\Sigma_1) \subseteq \Sigma_0, \quad p(N_1) \subseteq N_0, \quad p(H_1) \subseteq H_0, \quad p(E_1) \subseteq E_0, \quad p(K_1) \subseteq K_0;$

(2) $W_0 \circ p = p \circ W_1, \quad P_0 \circ p = p \circ P_1, \quad F_0 \circ p = p \circ F_1, \quad S_0 \circ p = p \circ S_1;$

(3) $\begin{tikzcd} N_0\cup\Sigma_0 & N_1\cup\Sigma_1 \arrow[l, "p|_{N_1\cup\Sigma_1}"'] \\ H_0 \arrow[u, "A_0"] & H_1 \arrow[l, "p|_{H_1}"] \arrow[u, "A_1"'] \end{tikzcd}$ and $\begin{tikzcd} E_0 & E_1 \arrow[l, "p|_{E_1}"'] \\ K_0 \arrow[u, "C_0"] & K_1 \arrow[l, "p|_{K_1}"] \arrow[u, "C_1"'] \end{tikzcd}$ are pullbacks in the category of sets;

(4) $\overline{p} \circ G_1 = G_0 \circ \overline{p};$

(5) G_1 is minimal relative to p.

Condition (1) means that p maps symbols to symbols, nodes to nodes, hooks to hooks, edges to edges, and facets to facets.

Condition (2) means that p preserves the W, P, F, S incidence functions.

Condition (3) means that p maps the hooks of any node n bijectively onto the hooks of $p(n)$, and maps the facets of any edge e bijectively onto the facets of $p(e)$.

Condition (4) means that p preserves gluing (i.e., if $G_1(x,y)$ then $G_0(p(x),p(y))$), and if $p(x)$ is a subsymbol, subnode, subhook, subedge or subfacet then so is x.

Condition (5) says that the gluings in $\mathcal{N}_1$ are induced by those in $\mathcal{N}_0$, i.e., two things are glued in $\mathcal{N}_1$ only if they are forced to be by condition (4).

The composition of two homomorphisms is a homomorphism and the inverse of a bijective homomorphism is a homomorphism [50, theorems 15,16].

Definition. If $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is a network, a *subnetwork* of $\mathcal{N}$ is a 12-tuple $(\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, where

$$\Sigma' \subseteq \Sigma, \quad N' \subseteq N, \quad H' \subseteq H, \quad E' \subseteq E, \quad K' \subseteq K$$

$$W(N') \subseteq \Sigma', \quad P(N') \subseteq \Sigma', \quad H' = A^{-1}(N' \cup \Sigma'), \quad F(E') \subseteq H', \quad S(E') \subseteq H', \quad K' = C^{-1}(E')$$

$$W' = W|_{N'\cup\Sigma'}, \quad P' = P|_{N'}, \quad A' = A|_{H'}, \quad F' = F|_{E'}, \quad S' = S|_{E'}, \quad C' = C|_{K'}$$

$$G \circ id_{\Sigma'\cup N'\cup H'\cup E'\cup K'} \subseteq id_{\Sigma'\cup N'\cup H'\cup E'\cup K'} \circ G, \quad G' = id_{\Sigma'\cup N'\cup H'\cup E'\cup K'} \circ G \circ id_{\Sigma'\cup N'\cup H'\cup E'\cup K'}.$$

If $\mathcal{N}'$ is a subnetwork of $\mathcal{N}$ then $\mathcal{N}'$ is a network and the inclusion function from $\mathcal{N}'$ into $\mathcal{N}$ is a homomorphism [50, theorem 24].

A *proper* subnetwork of $\mathcal{N}$ is a subnetwork $\mathcal{N}'$ such that $\mathcal{N}' \neq \mathcal{N}$.

4.3 Semi-definite and definite networks

The grammar is required to be a *semi-definite network* and the pattern to be a *definite* network. (While the pattern is under construction it is not definite.) First we need the concept of a *minimal* gluing relation.

Definition. If $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is a network then its gluing relation G is *minimal* relative to $\mathcal{N}$ iff

$$id_K \subseteq G_K \circ G_K^{-1} \cup G_K^{-1} \circ G_K$$

where $G_K = G \circ id_K$; and, for any relation $G^* \subseteq G$ such that

- $\overline{W} \circ G^* \subseteq G^* \circ \overline{W}$, $\overline{P} \circ G^* \subseteq \overline{P}$, $\overline{A} \circ G^* \subseteq G^* \circ \overline{A}$, $\overline{F} \circ G^* \subseteq G^* \circ \overline{F}$, $\overline{S} \circ G^* \subseteq G^* \circ \overline{S}$, $\overline{C} \circ G^* \subseteq G^* \circ \overline{C}$,
- $id_K \subseteq G^*_K \circ {G^*_K}^{-1} \cup {G^*_K}^{-1} \circ G^*_K$ (where $G^*_K = G^* \circ id_K$),

we have $G^* = G$.

(Informally, the condition that G be minimal relative to $\mathcal{N}$ means that every facet is glued to another facet and, subject to this constraint, G is as small as it can be.)

Definition. A network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is *semi-definite* iff

(1a) $W \circ A \circ F = W \circ A \circ S$,
(3) G_N is acyclic,
(4a) $H = F(E) \cup S(E)$,
(5a) $G_K \circ G_K^{-1} \subseteq id_K$,
(6) G is minimal relative to $\mathcal{N}$,
(7a) $\overline{F} \circ G = G \circ \overline{F}$ and $\overline{S} \circ G = G \circ \overline{S}$,
(8a) $\forall R \;\; (\overline{A}^{-1} \circ R = (\overline{F} \cup \overline{S}) \circ R \;\Rightarrow\; G_\Sigma \circ R \subseteq \overline{W} \circ G_N \circ R \;\cup\; id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1} \circ R)$.

Definition. A network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is *definite* iff

(1b) $E \underset{A \circ S}{\overset{A \circ F}{\rightrightarrows}} N \cup \Sigma \xrightarrow{W} \Sigma$ is a coequaliser diagram in the category of sets,
(2b) G_N is connected relative to P,
(3) G_N is acyclic,
(4b) $E \xrightarrow{F} H \xleftarrow{S} E$ is a sum diagram in the category of sets,
(5b) $id_K = G_K \circ G_K^{-1} \cup G_K^{-1} \circ G_K$,
(6) G is minimal relative to $\mathcal{N}$.

Condition (1a) says that the two nodes at the ends of any edge belong to the same whole. The stronger condition (1b) says that the nodes and edges belonging to any whole form a *connected* graph.

Condition (2b) means that two nodes share the same part only when they are glued together (directly or indirectly). (This condition prevents the same symbol from being interpreted as part of two unrelated wholes.)

Condition (3) means that there is no cyclic sequence of gluings. This is a technical condition for ensuring that condition (2b) holds in the pattern at the end of recognition [50, §3.6].

Condition (4a) says that every hook has at least one incident edge. The stronger condition (4b) says that every hook has exactly one incident edge.

Condition (5a) says that every facet has at most one superfacet. The stronger condition (5b) says that every facet has exactly one subfacet or superfacet.

Condition (6) says that every facet is glued to another facet, but the gluings are minimal subject to this constraint.

Condition (7a) says that, if any hook is a subhook, then all its incident edges are subedges.

Condition (8a) means roughly that, whenever a subsymbol is glued to a supersymbol, there are sufficiently many nodes and hooks belonging to the subsymbol that are glued to nodes and hooks belonging to the supersymbol. This is another technical condition for ensuring that condition (2b) holds in the pattern at the end of recognition (it is used directly in [50, theorem 22]).

Every definite network is semi-definite [50, theorem 18].

4.3 The recognition problem

We can now make our first formal statement of the recognition problem: given a semi-definite network $\mathcal{N}_0$ (the *grammar*) and an image, the task is construct a definite network $\mathcal{N}_1$ (the *pattern*) and a homomorphism $p{:}\,\mathcal{N}_1 \to \mathcal{N}_0$ (the *parse*).

This statement will be refined in §6.3.

5. The geometric part of the theory

This section is concerned with the spatial aspects of the pattern, in particular the following five issues:

- the geometric relationships between one symbol token and another;
- the *embedding* or 'pose' of a symbol token in the image, i.e., the transformation that maps the symbol token into the image at a certain position, orientation and size;
- how to represent the variability in the embeddings and relationships;
- invariance of the whole pattern under affine transformations;
- symmetries of the pattern.

I shall review the available approaches in the literature and introduce my own concept of a *fleximap*, a variable affine transformation.

5.1 Representing geometric relationships and embeddings

Some authors represent the relation between one symbol token and another using qualitative relationships (i.e., special named relationships) such as horizontal and vertical neighbourhood [94,15,63]; *above*, *below*, *before*, *after*, *overlap*, etc. [100,104]; closeness [42]; or topological relationships such as *hinged*, *butting*, *collinear*, *parallel*, *attached*, *concentric*, *radial*, *contained* [71] or *parallel*, *incidence*, *perpendicular*, *intersection*, *neighbour*, *inclusion* [76].

A more versatile approach is to represent relationships between symbol tokens in terms of geometric attributes, such as vanishing points and orientation angles [57]; distance and relative size [100]; relative angles [47]; distance and relative angle [70]; relative position, angle and size [92,22]; internal angles and ratios of lengths [108,107]; curvature and direction of bending [73]; closeness of attachment points, ratios of widths, curvature and angles of tilt [80]; and perpendicular distance from an endpoint of a line to another line [11].

Embedding transformations are sometimes represented as sequences of simple transformations, e.g., as *translation·rotation·dilation* [11] (where the dot denotes function composition), or *stretch · rotation · shear* [98], or via a singular value decomposition as *translation · rotation · dilation · stretch · rotation* [10].

5.2 Representing variability

Since symbol tokens are not likely to be perfectly positioned, some way is needed to allow for variability in their embeddings and relationships. The crudest way of allowing for variability is by rounding of geometric parameters (such as position, angle and size) into a small discrete set of possible values. A small variation in position, angle or size is tolerated, provided it does not cross over a rounding boundary [100,47]. A better-behaved approach is to allow geometric parameters to vary in a certain range [15,58]. These are 'hard' constraints: they either hold or do not.

Other authors use 'soft' constraints: deviations of the geometric parameters from their nominal values are penalised by a 'penalty' or 'affinity score' [42]. Sometimes fuzzy conditions are used [80]. Soft constraints have the advantage that they allow several alternative interpretations of a symbol token to be scored and a choice made between them. Most commonly, a quadratic penalty function is used to penalise deviations of the attributes from their nominal values [99,57,108,11,30]; an 'energy' or 'match' function is calculated by summing the penalties of all geometric relationships along with other penalty terms (and in some papers the energy is exponentiated to give a Gaussian probability density [81]). The energy can then be minimised, giving the optimal match of the symbols with the image. The advantage of using an energy function is that a high deviation in one symbol-pair relationship can be traded off against low deviations in other relationships to give the overall best match. In this way the pattern recognition process becomes sensitive to context.

Particularly interesting is the use of 'springs' to connect pairs of symbol tokens [45]. This allows variability in the relative position of the symbols; a different device is used for controlling relative orientation and length. This is similar to deformable templates, such as [105], in which a human eye is modelled with parabolic curves.

Each of these methods represents the variability of a relationship in a fixed way, for a particular purpose. For example, in [99] the variability is always analysed into rotations around the centre point, dilations from the centre point, and translations. What if one wanted a different decomposition, such as rotation or dilation about a different point? For example, in figure 6 we have three 'line' symbol tokens, which are parts of an 'A' symbol token. We may want to allow the crossbar part σ_3 to rotate around its end-point, which is 40% up the length of σ_2, while σ_3 may also dilate relative to σ_2 (keeping one end in contact with σ_2); σ_3 may be translated up or down σ_2 from the nominal 40% position; it may also overshoot or undershoot σ_2 by a tiny amount. Each of these kinds of variation are to be penalised by different coefficients. The total penalty should be a weighted sum of the squares of these variations.

This is accomplished by my concept of a *fleximap*, a variable affine transformation in which variations in relative position, orientation, size, stretching and shearing are treated in a uniform formalism. Assume that every symbol type has a *template*, depicting an ideal

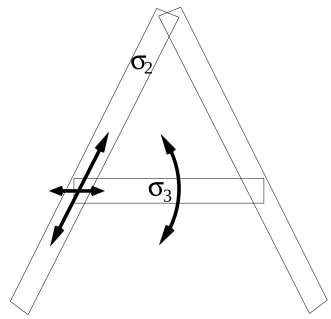


Figure 6. The relationship between two symbol tokens, and some of its dimensions of variability

token of that type, at a standard position, size and orientation. Every symbol token has an *embedding*, an affine transformation mapping from the plane of the template into the image plane. Figure 7 shows the templates for the 'A' and '*line*' symbol types, and embeddings for one symbol token σ_1 of type A and three symbol tokens $\sigma_2, \sigma_3, \sigma_4$ of type *line*. The embeddings are called $u(\sigma_1), u(\sigma_2), u(\sigma_3), u(\sigma_4)$.

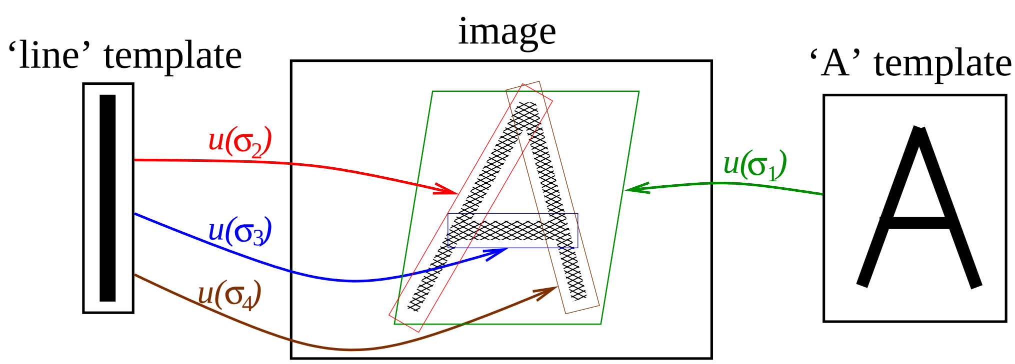


Figure 7. This shows an image and the templates for two symbol types *line* and A. In colour are shown the embedding transformations for four symbol tokens, σ_1 (of type A) and $\sigma_2, \sigma_3, \sigma_4$ (of type *line*).

Now consider the geometric relationship between σ_2 and σ_3: this is defined as $u(\sigma_3)^{-1} \cdot u(\sigma_2)$, which is a member of $\mathcal{G}$, the group of affine transformations (if the image is a plane this is a six-dimensional Lie group).

The relationship between σ_2 and σ_3 may be expressed in the form

$$u(\sigma_3)^{-1} \cdot u(\sigma_2) = F \cdot \exp(A) \tag{1}$$

where $F \in \mathcal{G}$ is the *nominal* or *ideal* value of the relationship, A is a member of $\mathcal{A}$, the six-dimensional Lie algebra of $\mathcal{G}$, and $\exp\colon \mathcal{A} \to \mathcal{G}$ is the exponential function. If you would like a concrete representation, think of a point x in the plane, an affine transformation $G \in \mathcal{G}$, and a member $A \in \mathcal{A}$ of the Lie algebra in matrix form using projective coordinates:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}, \quad G = \begin{pmatrix} g_{11} & g_{12} & s_1 \\ g_{21} & g_{22} & s_2 \\ 0 & 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} h_{11} & h_{12} & t_1 \\ h_{21} & h_{22} & t_2 \\ 0 & 0 & 0 \end{pmatrix}.$$

Then G may be applied to x by matrix multiplication, and exponentiation is defined by

$$\exp(A) = \sum_{n=0}^{\infty} \frac{A^n}{n!}.$$

A is called the *deviation* of $u(\sigma_3)^{-1} \cdot u(\sigma_2)$ from its nominal value. This deviation is quadratically penalised using a metric tensor.

Definition. A *metric tensor* is a function $g: \mathcal{A} \to (\mathcal{A} \to \mathbb{R})$ such that

- $\forall c_1, c_2 \in \mathbb{R}\ \forall A, B_1, B_2 \in \mathcal{A}\ \ g(A)(c_1 B_1 + c_2 B_2) = c_1 g(A)(B_1) + c_2 g(A)(B_2)$,
- $\forall A, B \in \mathcal{A}\ \ g(A)(B) = g(B)(A)$,
- $\forall A \in \mathcal{A}\ (A \neq 0 \Rightarrow g(A)(A) > 0)$.

A fleximap is simply the combination of the nominal transformation and the metric tensor.

Definition. A *fleximap* is a pair (F, g), where $F \in \mathcal{G}$ and g is a metric tensor.

The *penalty* of the affine transformation $u(\sigma_3)^{-1} \cdot u(\sigma_2)$ relative to the fleximap (F, g) is defined as $g(A)(A)$, where A is given by equation (1) above. To state it in general,

Definition. For any fleximap $\tau = (F, g)$ and any affine transformation G, define the *penalty* $E_\tau(G)$ of G relative to τ by

$$E_\tau(G) = g(A)(A)$$

where $G = F \cdot \exp(A)$.

If we choose a basis $(a_1, \ldots a_6)$ for $\mathcal{A}$, A may be expressed as $A = \sum_{i=1}^{6} A^i a_i$, where $A^1, \ldots A^6$ are real coefficients, and g may be represented as a positive-definite, symmetric real 6×6 matrix (g_{ij}), with $g(A)(A) = \sum_{i,j=1}^{6} g_{ij} A^i A^j$.

By a suitable choice of basis $(a_1, \ldots a_6)$, g may be put into diagonal form, $g = diag(g_1, g_2, g_3, g_4, g_5, g_6)$, where $g_1, g_2, g_3, g_4, g_5, g_6 > 0$. Then the penalty is $g(A)(A) = \sum_{i=1}^{6} g_i (A^i)^2$. We can choose g so that $a_1, \ldots a_6$ are the six desired dimensions of variation (a_1 = rotation about a certain point, a_2 = dilation about a certain point, a_3 = translation in a certain direction, etc.), and $g_1, g_2, g_3, g_4, g_5, g_6$ are the penalties for each of these types of deviation. Hence $g(A)(A)$ is a weighted sum of squares of deviations, as desired.

Thus the deviation A may be decomposed into any six (linearly independent) dimensions of variation that we like, and each dimension may be penalised as much as we like. For the full and rigorous theory of fleximaps see [49, §3,§5].

In the grammar, relationships between siblings are represented by edges; so each edge type will have a fleximap τ describing the geometric relationship. In the pattern, each edge token has the actual value of the relationship, e.g., $u(\sigma_3)^{-1} \cdot u(\sigma_2)$. The penalty is calculated for the actual relationship relative to the fleximap, and equals $E_\tau(u(\sigma_3)^{-1} \cdot u(\sigma_2))$.

Relationships between part and whole, e.g., between σ_2 and σ_1 in figure 7, are represented by nodes; each node type in the grammar will have a fleximap constraining this relationship. A penalty is calculated for the actual relationship $u(\sigma_1)^{-1} \cdot u(\sigma_2)$ relative to the fleximap.

All these penalties are summed (multiplied by -1), along with other penalty terms, to give the total *match function* (see §6.2,§7.2), which is to be maximised.

5.3 Invariance of the whole pattern under affine transformations

Whichever method is used to represent geometric transformations, some allowance needs to be made for invariance of the pattern under change of frame of reference. Qualitative spatial relationships such as *leftOf*, *aboveOf* are invariant under translation but not rotation; they are appropriate for applications where the image has already been normalised for orientation and spatial directions have a special significance, such as recognition of mathematical formulae [15] or musical scores [42].

Topological qualitative relationships [71] are invariant under homeomorphisms. The qualitative relationships in [76] are a mixture of topological invariants, affine invariants and similarity invariants.

Where geometric relationships are expressed in terms of angles and ratios of lengths [100,92,22,108,107] they are invariant under similarities; others are invariant under isometries [11].

My fleximap formalism is invariant under affine transformations. If the whole pattern is subjected to an affine transformation f then the embedding $u(\sigma)$ of every symbol token σ is transformed by $u(\sigma) \mapsto f \cdot u(\sigma)$, so the relationship $u(\sigma_2)^{-1} \cdot u(\sigma_1)$ between two symbol tokens σ_1 and σ_2 is unchanged, so the penalty value is unchanged.

5.4 Symmetry

The final issue to be considered is symmetry of the symbols and the whole pattern. A hexagonal grid pattern, for example, has six-fold rotational symmetry: every interpretation of the image has six symmetry variants. A pattern recognition system must recognise that these are equivalent, rather than treating them as six alternative, competing hypotheses, all of precisely equal merit. Thus symmetries must be represented explicitly. Every symbol type has a symmetry group, which we assume to be finite.

For example, the *line* symbol type has two symmetries: the identity transformation and rotation by π. Look back at figure 7. The symbol token σ_1 of type A expects its three line parts to be oriented in a certain way relative to it; for example, it may expect the crossbar to be oriented left-to-right; yet the line token σ_3 that plays the role of the crossbar may actually be oriented the other way round. If so, then if we calculated the penalty of $u(\sigma_3)^{-1} \cdot u(\sigma_1)$ against the fleximap we would get a spuriously high value. We need a way of saying that rotations by π are to be disregarded when calculating the penalty. To allow for this, it is convenient to give node tokens embeddings as well as symbol tokens. Recall that the part-whole relationship between σ_3 and σ_1 is represented by a node token n with $P(n) = \sigma_3$ and $W(n) = \sigma_1$. We give n an embedding $u(n)$, which is the same as $u(\sigma_3)$ except that it is oriented the expected way round, considered as a part of σ_1. That is, $u(\sigma_3) = u(n) \cdot s$, where s is either the identity or rotation by π. It is $u(\sigma_3)^{-1} \cdot u(n)$, rather than $u(\sigma_3)^{-1} \cdot u(\sigma_1)$, that we take as the part-whole relationship in the calculation of the penalty relative to the part-whole fleximap.

To specify a symmetry we must provide

- an automorphism $a : \mathcal{N}_0 \to \mathcal{N}_0$ of the grammar (i.e., an isomorphism onto itself),
- an affine transformation $s(\sigma)$ to be applied to every symbol type σ (it must be one of σ's symmetries),
- an affine transformation $s(n)$ to be applied to every node type n (it must be one of $P(n)$'s symmetries),

such that the grammar is unchanged when the automorphism is applied to it and all fleximaps are transformed using the $s(\sigma)$ and $s(n)$ functions. The symmetry itself is the pair (a, s). (See [49, §6.4,§7.5] for precise details.) For example, in one symmetry the hexagonal grid symbol type *hex* (see §3.3) is rotated by $\frac{\pi}{3}$, some of its nodes are rotated by π (but *line* is not), and the automorphism maps *hex* to *hex*, *line* to *line*, nodes 1,2,3 to nodes 2,3,1, and edges a, b, c, d, e, f

to b, c, a, e, f, d, respectively. Such symmetries of the grammar can then be applied to the pattern $\mathcal{N}_1$ with a parse $p: \mathcal{N}_1 \to \mathcal{N}_0$ and a set of embeddings; a *local* symmetry of $\mathcal{N}_1$ consists of applying a separate symmetry to each symbol, producing a new parse and a new set of embeddings. For example, a symmetry can be applied to a single token of type *hex* and its subsymbols, nodes and edges, leaving everything else unchanged [50, §4.3].

5.5 Embedding tokens and types

It is convenient to bundle together all the embedding transformations into a single mathematical object, the function $u: \Sigma_1 \cup N_1 \to \mathcal{G}$ (where $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is the pattern). This is called an *embedding token* for $\mathcal{N}_1$. It is subject to the constraint

$$\forall n, n^* \in N_1 \; (G_1(n, n^*) \Rightarrow u(n) = u(n^*))$$

i.e., if two nodes are glued then they have the same embedding.

It is also convenient to bundle together all the fleximaps that constrain these embedding transformations into a single mathematical object, called an *embedding type* for $\mathcal{N}_1$. This is a sextuple $v_1 = (sub_1, con_1, rel_1, symm_1, tem_1, in_1)$, consisting of functions $sub_1: \{ (\sigma, \sigma^*) \in \Sigma_1 \times \Sigma_1 \mid G_1(\sigma, \sigma^*) \} \to \mathit{Flex}$, $con_1: N_1 \to \mathit{Flex}$, $rel_1: E_1 \to \mathit{Flex}$, $symm_1: \Sigma_1 \to \mathit{Sub}(\mathcal{G})$, $tem_1: \Sigma_1 \to \mathit{Tem}$, and $in_1: \Sigma_1 \to \mathit{Flex}$, where *Flex* is the set of all fleximaps, $\mathit{Sub}(\mathcal{G})$ is the set of all subgroups of $\mathcal{G}$, and *Tem* is the set of all templates. These are interpreted as follows.

For any supersymbol σ and subsymbol σ^*, $sub_1(\sigma, \sigma^*)$ is the fleximap defining the variable geometric relationship between them.

For any node n, $con_1(n)$ is the fleximap defining the geometric relationship between the part $P(n)$ and the whole $W(n)$, or, more accurately, the relationship between the node and the whole, as explained in the previous subsection.

For any edge e, $rel_1(e)$ is the fleximap defining the geometric relationship between the nodes $A(F(e))$ and $A(S(e))$.

For any symbol σ, $symm_1(\sigma)$ is the symmetry group of σ, and $tem_1(\sigma)$ is its template; $in_1(\sigma)$ is a fleximap (I, g) whose nominal part I is just the identity affine transformation, but whose metric g is the *inertial* metric of σ, which is used to determine how σ moves in response to forces (see §7.3).

But where does this embedding type v_1 for $\mathcal{N}_1$ come from? We are given an embedding token $v = (sub, con, rel, symm, tem, in)$ for the grammar $\mathcal{N}_0$, which captures all the geometric information associated with the grammar. We can transfer it to the pattern using the homomorphism p, giving an embedding type $v_1 = v \circ p$ for $\mathcal{N}_1$, defined by

$$v \circ p = (sub_1, con \circ p, rel \circ p, symm \circ p, tem \circ p, in \circ p).$$

where

$$\forall \sigma, \sigma^* \in \Sigma_1 \quad (G_1(\sigma, \sigma^*) \;\Rightarrow\; sub_1(\sigma, \sigma^*) = sub(p(\sigma), p(\sigma^*))).$$

v_1 is called the *induced* embedding type on $\mathcal{N}_1$. It is v_1 that we compare with the embedding token u, as just described.

Embedding tokens are further constrained by a symmetry condition. If u is the embedding token for the pattern $\mathcal{N}_1$ and $v_1 = v \circ p = (sub_1, con_1, rel_1, symm_1, tem_1, in_1)$ is the embedding type for $\mathcal{N}_1$, the *symmetry condition* for $\mathcal{N}_1, u, v_1$ is

$$\forall n \in N_1 \quad u(P_1(n))^{-1} \cdot u(n) \in symm_1(P_1(n)).$$

This states that each node n is embedded in the image plane in the same way as the part $P_1(n)$, up to a symmetry of $P_1(n)$ (as explained in the previous subsection). This symmetry condition will be imposed throughout the recognition process.

6. Templates and the definite match function

6.1 Templates

So far I have said nothing about the relationship between the image and the pattern. It is only at this point that we need to make any assumptions about the nature of the raw data, which I have called the 'image'. Assume in this section that the image is a rectangular array of monochrome pixels. Formally, an *image* is a function $I\colon \mathbb{R}^2 \to [0,\infty)$. For any point $p \in \mathbb{R}^2$, $I(p)$ is the image intensity at the pixel p. The domain of I is called the *image plane*.

Recall figure 7 in §5.2. Formally, a template is a differentiable function $T\colon \mathbb{R}^2 \to [0,\infty)$ such that the set $\{\, x \in \mathbb{R}^2 \mid T(x) > 0 \,\}$ is bounded. The domain of T is called the *template plane*. Each token σ of this type has an embedding transformation $G = u(\sigma)$, mapping from the template plane to the image plane.

We shall define a measure $\rho_{I,T}(G)$, called the *correlation function*, of how well a template T matches the image (transferred into the template plane) $I \circ G$, and will seek to choose G to maximise it. Define

$$\rho_{I,T}(G) = |det(G)| \int T(u)\,(I(G(u)) - I_0)\,\mathrm{d}^2u = \int T(G^{-1}(x))\,(I(x) - I_0)\,\mathrm{d}^2x$$

where I_0 is a positive real constant associated with T. The integrals are over the whole of $\mathbb{R}^2$, or equivalently over a large enough region to include in its interior all the points where T is non-zero.

Each symbol type has a template (this is provided as part of the embedding type v, see §5.5). This is transferred across to the pattern when we form $v \circ p$, so now we can say that each symbol token σ has a template, $tem(\sigma)$, which is equal to the template of its symbol type.

There is one complication. We wish to prevent two identical symbols from forming at the same place in the image, or more generally to discourage two or more symbols from claiming credit for the same patch of the image. If two or more symbol tokens' templates overlap in the image plane, i.e., if there are points x in the image plane where $tem(\sigma)(u(\sigma)^{-1}(x)) > 0$ for two or more symbol tokens σ, then the point x should become *saturated* and the contribution it makes to the correlation function should be reduced to penalise the overlapping symbol tokens. For this it is necessary to calculate the saturation $sat(x)$ of each point x, which is roughly the sum of $tem(\sigma)(u(\sigma)^{-1}(x))$ over each symbol token σ. (However, this needs modification for subsymbols to avoid double-counting.) The formal definition of saturation follows.

Suppose we are given a pattern $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$, an embedding token u for $\mathcal{N}_1$, and a function $tem\colon \Sigma_1 \to Tem$, then $sat\colon \mathbb{R}^2 \to \mathbb{R}$, is defined by

$$\forall x \in \mathbb{R}^2 \quad sat(x) = \sum_{\sigma \in \Sigma_1} (1 - g_\sigma) \cdot tem(\sigma)(u(\sigma)^{-1}(x)).$$

where $g_\sigma = \left|\{\, \sigma^* \in \Sigma \mid G_1(\sigma^*, \sigma) \,\}\right|$. We call $sat(x)$ the *saturation at* the image point x.

The correlation function is now modified to take account of saturation:

$$\rho_{I,T,sat}(G) = |det(G)| \int w(sat(G(u)))\, T(u)\, (I(G(u)) - I_0)\, \mathrm{d}^2u = \int w(sat(G))\, T(G^{-1}(x))\, (I(x) - I_0)\, \mathrm{d}^2x$$

where w is a weighting function that suppresses the integrand at points x where $sat(x)$ is above 1. A suitable definition of w is

$$\forall s \in \mathbb{R} \quad w(s) = \begin{cases} 1 & \text{if } s \leq 1, \\ (1.6 - s)/0.6 & \text{if } 1 < s < 1.6, \\ 0 & \text{if } 1.6 \leq s. \end{cases}$$

We also define *mass*, m, of a symbol token, with template T, and which is embedded in the image I by G, by

$$m = |det(\overline{G})| \int T(u)\, \mathrm{d}^2u = \int T(G^{-1}(x))\, \mathrm{d}^2x.$$

This is used when calculating how much the symbol token moves in response to a force (see §7.3 below).

6.2 The definite match function

We are now in a position to state the *definite match function*, which measures how well a definite pattern $\mathcal{N}$ matches an image I, and how well $\mathcal{N}$'s embedding token u matches an embedding type v for $\mathcal{N}$. It is the aim of recognition to construct a definite pattern that maximises the definite match function.

The definite match function DM is defined by

$$\begin{aligned} DM(I, \mathcal{N}, u, v) = &\sum_{\sigma \in \Sigma} \rho_{I,tem(\sigma),sat}(u(\sigma)) \;-\; \theta|\Sigma \backslash P(N)| \;-\; \sum_{(\sigma,\sigma^*)|G(\sigma,\sigma^*)} E_{sub(\sigma,\sigma^*)}\big(u(\sigma)^{-1} \cdot u(\sigma^*)\big) \\ &- \sum_{n \in N} E_{con(n)}\big(u(W(n))^{-1} \cdot u(n)\big) \;-\; \sum_{e \in E} E_{rel(e)}\big(u(A(S(e)))^{-1} \cdot u(A(F(e)))\big) \end{aligned}$$

where $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, $v = (sub, con, rel, symm, tem, in)$, θ is a positive real constant and sat is the saturation function, as defined in the previous section.

The first term on the right-hand side is a sum over all symbols; $tem(\sigma)$ is the template for σ, and $\rho_{I,tem(\sigma),sat}(u(\sigma))$ measures the correlation between the template (embedded in the image using $u(\sigma)$) and the image I.

The second term on the right-hand side applies a fixed penalty of θ for each 'bare' symbol, i.e., each symbol that is not a part of another symbol. This term encourages the symbols to connect themselves together rather than remaining separate.

The third term measures how well each subsymbol matches its supersymbol geometrically. The summation is over all pairs (σ, σ^*) such that σ is a supersymbol of σ^*; $sub(\sigma, \sigma^*)$ is a fleximap that defines what the geometric relationship between σ^* and σ should be; $u(\sigma)^{-1} \cdot u(\sigma^*)$ is the actual relationship between them; the quadratic penalty function $E_\tau(G)$ calculates the penalty for the deviation between an affine transformation G and a fleximap τ, as defined in §5.2.

The fourth term measures how well each part matches its whole geometrically. The summation is over all nodes n; $P(n)$ is the part symbol and $W(n)$ is the whole symbol. The node n has its own embedding $u(n)$, which equals $u(P(n))$ up to a symmetry; $u(W(n))^{-1} \cdot u(n)$ is the actual geometric relationship between the part (or rather the node) and the whole; $con(n)$ is a fleximap specifying what the relationship should be.

The final term measures how well each pair of siblings match geometrically. The sum is over every edge e, representing a sibling relationship between two nodes $A(F(e))$ and $A(S(e))$; $u(A(S(e)))^{-1} \cdot u(A(F(e)))$ is the actual relationship and $rel(e)$ is the fleximap specifying what the relationship should be.

All these fleximaps are provided by the embedding type v; the final component in of v is not used yet.

The DM function is invariant under application of affine transformations to the symbols' internal frames of reference, under local symmetries, and under affine transformations of the image of determinant 1 [50, theorems 28-31].

The DM function is used as follows. We are given an embedding type v for the grammar, we transfer it across to an embedding type $v \circ p$ for the pattern, and DM allows us to compare this with the embedding token u that we have constructed for the pattern, while also comparing the pattern $\mathcal{N}_1$ with the image I.

6.3 Full statement of the recognition problem

We can now refine our statement of the recognition problem in §4.3. Given a semi-definite network $\mathcal{N}_0$ (the *grammar*), an embedding type v for $\mathcal{N}_0$, and an image I, the task is construct a definite network $\mathcal{N}_1$ (the *pattern*), an embedding token u for $\mathcal{N}_1$ (specifying how everything in the pattern is embedded in the image), and a homomorphism $p\colon \mathcal{N}_1 \to \mathcal{N}_0$ (the *parse*), maximising $DM(I, \mathcal{N}_1, u, v \circ p)$, subject to the symmetry condition for $\mathcal{N}_1, u, v \circ p$.

7. Inclusion functions and the indefinite match function

We now turn to the algorithm for solving the recognition problem. A pattern $\mathcal{N}_1$ is constructed incrementally. While it is under construction it is indefinite, meaning that alternative grammatical interpretations co-exist for different portions of the image. Some alternatives are evaluated as better than others. Each symbol token σ has an *inclusion value* $i(\sigma) \in [0,1]$, which is the algorithm's degree of confidence in σ, where $i(\sigma) = 1$ means that σ is definitely correct and $i(\sigma) = 0$ means that σ is definitely wrong and will be pruned. Similarly each node token n and each edge token e has an inclusion value $i(n)$ and $i(e)$. The function i is called an *inclusion function* and is accompanied by a second inclusion function j, explained below.

By the end of the recognition process, all inclusion values have been driven to the extremes, 0 or 1, and everything with inclusion value 0 has been pruned.

7.1 Definition of inclusion functions

Definition. A pair of *inclusion functions* on a network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is defined as (i,j), where $i\colon \Sigma \cup N \cup E \to [0,1]$ and $j\colon \Sigma \cup N \cup H \cup (K \backslash dom(G)) \to [0,1]$, such that

$$\forall \sigma \in \Sigma \quad i(\sigma) = j(\sigma) + \sum_{n \in P^{-1}(\{\sigma\})} (1 - g_n) i(n), \qquad \text{where } g_n = \left|\{\, n^* \in N \mid G(n^*, n) \,\}\right| \tag{1}$$

$$\forall n \in N \quad i(W(n)) = j(n) + i(n) \tag{2}$$

$$\forall h \in H \quad i(A(h)) = j(h) + \sum_{e \in F^{-1}(\{h\})} i(e) + \sum_{e \in S^{-1}(\{h\})} i(e) \tag{3}$$

$$\forall k \in K \backslash dom(G) \quad i(C(k)) = j(k) + \sum_{k^* \mid G(k,k^*)} i(C(k^*)). \tag{4}$$

(The summation notation in (4) means a sum over all k^* such that $G(k,k^*)$.)

This definition may be interpreted informally as follows. (In the following explanation I shall say 'should' or 'is correct' to state what will hold or exist at the end of recognition when the pattern is definite, and a simple 'is' for what is true during recognition when the pattern is indefinite.)

Line (1). Each symbol σ should either be a 'bare' symbol (with $P^{-1}(\{\sigma\}) = \emptyset$) or a part of one larger symbol (with $|P^{-1}(\{\sigma\})| = 1$). Hence the nodes presently in $P^{-1}(\{\sigma\})$ are in competition with one another; at most one can be correct. $i(\sigma)$ is interpreted as the degree of confidence in σ, $j(\sigma)$ is the degree of confidence that σ should be a bare symbol, and $i(n)$ is the degree of confidence in n. An exception to this competition is that if n is a subnode of n^* (i.e., $G(n^*,n)$) then they may both be correct; the $1-g_n$ factor allows for this co-existence.

In line (2), $i(W(n))$ is the degree of confidence in $W(n)$, $j(n)$ is the degree of confidence that $W(n)$ is correct but n is not, and $i(n)$ is the degree of confidence in n.

Line (3) expresses the fact that each hook h should have a single edge incident to it (i.e., $|F^{-1}(\{h\})| + |S^{-1}(\{h\})| = 1$); hence the edges presently in $F^{-1}(\{h\})$ and $S^{-1}(\{h\})$ are in competition with one another. Thus $i(A(h))$ is the degree of confidence in the node or symbol $A(h)$, $i(e)$ is the degree of confidence in e, and $j(h)$ is the degree of confidence that $A(h)$ is correct but that none of its present edges is (i.e., the correct edge has yet to be created).

Line (4) expresses the fact that each facet k that is not itself a sub-facet should be glued to a single sub-facet. Hence if k is presently glued to several sub-facets then they are in competition with one another; $i(C(k^*))$ is the degree of confidence in the edge $C(k^*)$, $j(k)$ is the degree of confidence that $C(k)$ is correct but that none of k's present sub-facets is.

At the end of recognition all surviving symbols, nodes and edges will have inclusion values 1 and the pattern will be definite.

Theorem 32 (from [50]). If $\mathcal{N} = (\Sigma,N,H,E,K,W,P,A,F,S,C,G)$ is a definite network and (i,j) is a pair of inclusion functions on $\mathcal{N}$ satisfying $\forall x{\in}\Sigma\cup N\cup E\ i(x) = 1$, then

$$\forall\sigma{\in}\Sigma\ j(\sigma) = \begin{cases} 0 & \text{if } \sigma\in P(N) \\ 1 & \text{otherwise} \end{cases}, \quad \forall n{\in}N\ j(n) = 0, \quad \forall h{\in}H\ j(h) = 0, \quad \forall k{\in}K\setminus\mathrm{dom}(G)\ j(k) = 0.$$

7.2 The indefinite match function, IM

We can now generalise the definite match function DM to a function applicable to indefinite patterns. First we generalise the (definite) saturation function defined in §6.1 to the *indefinite* saturation function $sat\colon\mathbb{R}^2\to\mathbb{R}$, defined by

$$\forall x{\in}\mathbb{R}^2\quad sat(x) = \sum_{\sigma\in\Sigma}(1-g_\sigma)\,i(\sigma)\,tem(\sigma)(u(\sigma)^{-1}(x)).$$

(The difference is that every symbol σ is now weighted by $i(\sigma)$.) Then the indefinite match function IM is defined by

$$\begin{aligned}
&IM(I,\mathcal{N},u,v,i,j,B) = \\
&\quad\sum_{\sigma\in\Sigma}\big(i(\sigma)\rho_{I,tem(\sigma),sat}(u(\sigma)) - j(\sigma)B(\sigma)\big) - \sum_{(\sigma,\sigma^*)|G(\sigma,\sigma^*)} i(\sigma^*)E_{sub(\sigma,\sigma^*)}(u(\sigma)^{-1}\cdot u(\sigma^*)) \\
&\quad-\sum_{n\in N} i(n)E_{con(n)}(u(W(n))^{-1}\cdot u(n)) - \sum_{h\in H} j(h)B(h) - \sum_{k\in K\setminus\mathrm{dom}(G)} j(k)B(k) \\
&\quad-\sum_{e\in E} i(e)\big(E_{rel(e)}(u(A(S(e)))^{-1}\cdot u(A(F(e)))) + E_{in(W(A(F(e))))}(u(W(A(S(e))))^{-1}\cdot u(W(A(F(e)))))\big).
\end{aligned}$$

Here, I is the image, $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is the network we are comparing with the image, u is the embedding token that specifies how $\mathcal{N}$ is embedded in I, $v = (sub, con, rel, symm, tem, in)$ is the embedding type against which u is evaluated, (i,j) are the inclusion functions, and B is called the *bareness function*.

The terms of IM are much like the terms of DM, except that everything is weighted by its inclusion value. The fixed penalty θ levied for each bare symbol is replaced by a variable penalty $B(\sigma)$; it is weighted by $j(\sigma)$ because $j(\sigma)$ is the degree of confidence that σ is bare. There is a new penalty $B(h)$ for a bare hook (i.e., a hook with no incident edges); this is weighted by the degree of confidence $j(h)$ that h is bare. Similarly there is a new penalty $B(k)$ for a bare facet (a facet that should have a sub-facet but does not); this is weighted by the degree of confidence $j(k)$ that k is bare.

All these variable penalties are specified by the bareness function $B: \Sigma \cup H \cup (K \setminus \mathrm{dom}(G)) \to [0, \infty)$.

The final term $(E_{in(W(A(F(e))))}(\ldots))$ is new. Normally, for any edge e, we have $W(A(F(e))) = W(A(S(e)))$, which means that the two nodes at each end of e belong to the same whole (this is called a *coherent* edge). However, incoherent edges are temporarily permitted during recognition. They may occur, for example, with the hexagonal grids in §10.1. There may be genuine uncertainty about whether two overlapping hexagonal grids should be merged or kept separate. Incoherent edges may form between a node of one grid and a node of the other. If so, this is penalised in IM by using a metric tensor $in(\sigma)$, where in is taken from v and σ is one of the two grids. This term penalises incoherent edges; it is 0 if the edge is coherent. As we are about to see, each term in IM gives rise to forces that change the symbols' embeddings. This final term exerts a force that draws the two wholes together; eventually, either they will come close enough to be merged or the incoherent edges will be removed. At the end all edges will be coherent.

The next theorem shows that DM is a special case of IM.

Theorem 33 (from [50]). If

(a) $\mathcal{N}$ is definite,

(b) $\forall x \in \Sigma \cup N \cup E \; i(x) = 1$,

(c) $\forall \sigma \in \Sigma \setminus P(N) \; B(\sigma) = \theta$,

then $IM(I, \mathcal{N}, u, v, i, j, B) = DM(I, \mathcal{N}, u, v)$.

When we use IM during recognition, the network we take is the pattern $\mathcal{N}_1$, and the embedding type we take is $v \circ p$, i.e., the given embedding type v for the grammar $\mathcal{N}_0$ transferred across to $\mathcal{N}_1$ using the parse $p: \mathcal{N}_1 \to \mathcal{N}_0$. So it is $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$ that we shall be maximising. At the end of the recognition process, when the conditions of theorem 33 will hold, we shall have maximised $DM(I, \mathcal{N}_1, u, v)$, as required.

7.3 Adjustment of the embedding token

During recognition the embeddings of all the symbols and nodes are continually adjusted to maximise $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$. Consider making a small change to the embedding token $u \mapsto u \cdot \Delta u$, where

$$\forall \sigma \in \Sigma \quad \Delta u(\sigma) = \exp(\varepsilon V_\sigma)$$

$$\forall n \in N \quad \Delta u(n) = \exp(\varepsilon V_n)$$

where the increments $V_\sigma, V_n \in \mathcal{A}$ (recall that $\mathcal{A}$ is the Lie algebra of the affine group). Then, developing the change in the value of IM to first order,

$$IM(I,\mathcal{N},u\cdot\Delta u,v,i,j,B) = IM(I,\mathcal{N},u,v,i,j,B) + \varepsilon \sum_{\sigma\in\Sigma} F_\sigma(V_\sigma) + o(\varepsilon).$$

Here $F_\sigma \in \mathcal{F}$, the dual vector space to $\mathcal{A}$. (The theory of $\mathcal{A}$ and $\mathcal{F}$ is developed fully in [49].) Elements of $\mathcal{F}$ are called *forces*. Note that the change depends only on the V_σ increments, not also on the V_n increments, because they are tightly related by the symmetry condition.

The forces can be understood informally as follows. Each term in IM generates (through its derivative) a force at a symbol, node or edge, which propagates through the network; the forces reaching any symbol σ are summed to give the total force F_σ acting on σ (see [49, §7.6] and [50, §6.5]).

We then apply gradient ascent, choosing our increment Δu to increase the value of IM. To make this well-defined we must specify the cost of making the adjustment; this requires a choice of an *inertial* metric tensor $in(\sigma)$. The cost is defined as $\frac{1}{2}\varepsilon\sum_{\sigma\in\Sigma} m_\sigma\, in(\sigma)(V_\sigma)(V_\sigma)$, where m_σ is the mass of σ, defined in §6.1. It is a simple exercise [49, §7.7] to show that the optimal choice of increment is

$$\forall\sigma\in\Sigma \quad V_\sigma = \frac{1}{m_\sigma} in(\sigma)^{-1}(F_\sigma).$$

Thus the role of the inertial metric $in(\sigma)$ is to convert (covariant) forces into (contravariant) increments. The increments V_n for nodes n are then determined by the symmetry condition. This determines the change $u \mapsto u\cdot\Delta u$ in the embedding token (up to an arbitrary choice of the positive constant ε). This incremental adjustment, continually repeated, optimises the embedding of the pattern in the image.

The inertial metrics $in(\sigma)$ are determined by the function in, provided as part of the embedding type. (As we saw in the last section $in(\sigma)$ also has a secondary use, in the penalty for incoherent edges in the IM function; this is separate from its primary use here.)

7.4 How the inclusion functions are determined

The inclusion functions i,j are continually recalculated during the recognition process by a simulated annealing process, governed by a *temperature* parameter that varies across the pattern and with time. Where the structure of the pattern is changing, the temperature is high, and this makes the inclusion functions take on mid-range values, so several alternative interpretations can co-exist in parallel; when structural changes stop, the temperature declines and the inclusion functions are pushed towards 0 or 1, and so a choice is forced between alternatives. Every symbol token, node token or edge token with inclusion value 0 is pruned. Eventually everything has inclusion value 1 and the pattern becomes definite.

For the purposes of this section only, we reformulate i,j and the constraints on them in a vector notation in order to emphasise their linear nature. Given a pair of inclusion functions (i,j) on a network $\mathcal{N} = (\Sigma,N,H,E,K,W,P,A,F,S,C,G)$, we shall convert (i,j) into the *inclusion vector* $\boldsymbol{i}$ on $\mathcal{N}$, with components $\boldsymbol{i}_x$, for all $x\in X$, where

$$X = \Sigma\times\{0,1\} \ \cup\ N\times\{0,1\}\ \cup\ H\ \cup\ E\ \cup\ K\backslash dom(G).$$

Each component of $\boldsymbol{i}$ represents one value of i or j, as follows.

$$\forall \sigma \in \Sigma \quad \boldsymbol{i}_{(\sigma,0)} = i(\sigma), \quad \boldsymbol{i}_{(\sigma,1)} = j(\sigma)$$
$$\forall n \in N \quad \boldsymbol{i}_{(n,0)} = i(n), \quad \boldsymbol{i}_{(n,1)} = j(n)$$
$$\forall h \in H \quad \boldsymbol{i}_h = j(h)$$
$$\forall e \in E \quad \boldsymbol{i}_e = i(e)$$
$$\forall k \in K \backslash dom(G) \quad \boldsymbol{i}_k = j(k)$$

The constraints on i and j in §7.1 are linear and so can be expressed as a set of linear conditions on $\boldsymbol{i}$ of the form

$$\forall y \in Y \quad \boldsymbol{c}^y \cdot \boldsymbol{i} = \sum_{x \in X} \boldsymbol{c}^y_x \boldsymbol{i}_x = 0$$

using a set of vectors $\boldsymbol{c}^y$, for all $y \in Y$, where

$$Y = \Sigma \cup N \cup H \cup K \backslash dom(G)$$

and $\boldsymbol{c}^y$ has components $\boldsymbol{c}^y_x$ for $x \in X$.

The *IM* function is also linear in $\boldsymbol{i}$ (if we disregard the dependence of the saturation function on $\boldsymbol{i}$) and so can be written in the form

$$IM(I, \mathcal{N}, u, v, i, j, B) = \boldsymbol{i} \cdot \boldsymbol{m} = \sum_{x \in X} \boldsymbol{i}_x \boldsymbol{m}_x$$

where $\boldsymbol{m}$ is a vector whose components $\boldsymbol{m}_x$ depend on $I, \mathcal{N}, u, v, B$.

The vector $\boldsymbol{i}$ is determined by maximising the expression

$$E = \sum_{x \in X} \frac{\boldsymbol{i}_x \boldsymbol{m}_x}{T_x} - \sum_{x \in X} \big(\boldsymbol{i}_x \ln \boldsymbol{i}_x + (1 - \boldsymbol{i}_x) \ln(1 - \boldsymbol{i}_x)\big)$$

subject to the constraints $\forall y \in Y \ \boldsymbol{c}^y \cdot \boldsymbol{i} = 0$, where each T_x is a positive number, known as the *temperature* of x. To be precise, every symbol, node, hook, edge and facet has a temperature, and we define $T_{(\sigma,0)} = T_{(\sigma,1)} = T_\sigma$ and $T_{(n,0)} = T_{(n,1)} = T_n$. The solution is

$$\boldsymbol{i}_x = sig\left(\frac{\boldsymbol{m}_x}{T_x} + \sum_{y \in Y} \lambda_y \boldsymbol{c}^y_x\right)$$

where the sigmoid function $sig\colon \mathbb{R} \to (0,1)$ is defined by $\forall u \in \mathbb{R} \ sig(u) = \frac{1}{1+e^{-u}}$ and the λ_y parameters are unknown Lagrange multipliers. By examining the second derivatives it can be determined that this solution is a local maximum of E.

We can split each $\boldsymbol{c}^y$ vector into two vectors $\boldsymbol{c}^{y+}$ and $\boldsymbol{c}^{y-}$ by separating positive and negative components:

$$\forall y \in Y \ \forall x \in X \quad \boldsymbol{c}^{y+}_x = \max(\boldsymbol{c}^y_x, 0), \quad \boldsymbol{c}^{y-}_x = \max(-\boldsymbol{c}^y_x, 0),$$

so that the constraints may be written as $\forall y \in Y \ \boldsymbol{c}^{y+} \cdot \boldsymbol{i} = \boldsymbol{c}^{y-} \cdot \boldsymbol{i}$. Also define

$$\forall y \in Y \quad C^y = \big(\max_{x \in X} \boldsymbol{c}^{y+}_x\big) + \big(\max_{x \in X} \boldsymbol{c}^{y-}_x\big).$$

The following iterative algorithm seeks values of λ_y satisfying the constraints.

For each $y \in Y$, initialise λ_y to its final value last time this algorithm was run;
repeat
  for each $x \in X$, do $\boldsymbol{i}_x := sig\big(\boldsymbol{m}_x / T_x + \sum_{y \in Y} \lambda_y \boldsymbol{c}^y_x\big)$ |
  for each $y \in Y$, do $\lambda_y := \lambda_y + \dfrac{1}{C^y} \ln\left(\dfrac{\boldsymbol{c}^{y-} \cdot \boldsymbol{i}}{\boldsymbol{c}^{y+} \cdot \boldsymbol{i}}\right)$
until equilibrium;
for each $x \in X$, if $\boldsymbol{i}_x$ is very close to 0 or 1, then round it to 0 or 1.

In this algorithm the semicolon means sequential composition, the '|' symbol means parallel composition, and the 'for each' loops are parallel loops. This means that all the assignment statements in the 'repeat' loop body may be executed concurrently in any fair order. Note that this loop is by far the most computationally expensive part of the entire recognition process, so the high degree of parallelism is relevant from the point of view of time complexity.

I have no proof of convergence for this, but a heuristic argument [50, §7.4] shows that it tends to bring the constraints closer to satisfaction. It can only halt when all the constraints are satisfied.

Thus the inclusion functions are chosen to maximise E. In the final stages of recognition, all the temperatures will converge to the minimum allowed temperature $T_{min} > 0$ and so each $\boldsymbol{i}_x$ is likely to approach 0 or 1, and is then rounded to 0 or 1, giving

$$E = \frac{IM(I,\mathcal{N}_1,u,v \circ p,i,j,B)}{T_{min}}.$$

Hence maximising E ultimately maximises $IM(I,\mathcal{N}_1,u,v \circ p,i,j,B)$. By theorem 33 this maximises $DM(I,\mathcal{N}_1,u,v \circ p)$ for the final pattern.

8. The structural operations

This section describes the *structural operations* by which the pattern is incrementally grown during recognition. There are four kinds:

- pruning operations,
- extension operations,
- merging two symbol tokens,
- partitioning a symbol token into two.

When such an operation is applied to the pattern $\mathcal{N}_1$, the parse $p\colon\mathcal{N}_1 \to \mathcal{N}_0$, the embedding token u on $\mathcal{N}_1$, and the inclusion functions (i,j) are updated accordingly.

8.1 Pruning operations

Definition. Given a network $\mathcal{N} = (\Sigma,N,H,E,K,W,P,A,F,S,C,G)$ and a pair (i,j) of inclusion functions on $\mathcal{N}$, a *pruning operation* is a transformation from $\mathcal{N}$ to a subnetwork $\mathcal{N}' = (\Sigma', N',H',E',K',W',P',A',F',S',C',G')$ such that

$$\forall\sigma\in\Sigma\backslash\Sigma'\ i(\sigma) = 0, \quad \forall n\in N\backslash N'\ i(n) = 0, \quad \forall e\in E\backslash E'\ i(e) = 0.$$

A pruning operation is *trivial* iff $\mathcal{N}' = \mathcal{N}$.

In practice we may confine ourselves to *elementary* pruning operations, which consist of

- *pruning a symbol* σ – removing σ and its subsymbols, and all their dependent nodes, hooks, edges and facets;
- *pruning a node* n – removing n and its subnodes, and all their dependent hooks, edges and facets;
- *pruning an edge* e – removing e and its subedges, and all their facets.

A pruning operation, since it only removes things with inclusion value 0, does not alter the value of the *IM* function [50, theorem 41].

Definition. A network $\mathcal{N} = (\Sigma,N,H,E,K,W,P,A,F,S,C,G)$ is *unprunable*, given a pair of inclusion functions (i,j) on $\mathcal{N}$, iff there is no proper subnetwork $\mathcal{N}' = (\Sigma',N',H',E',K',W',P',A',F',S',C',G')$ of $\mathcal{N}$ satisfying

$$\forall\sigma\in\Sigma\backslash\Sigma' \; i(\sigma) = 0, \quad \forall n\in N\backslash N' \; i(n) = 0, \quad \forall e\in E\backslash E' \; i(e) = 0.$$

(In other words, a network is unprunable iff no non-trivial pruning operation is possible on it, or, equivalently [50, theorem 40], iff no elementary pruning operation is possible on it.)

8.2 Extension operations

Definition. Given a grammar $\mathcal{N}_0$, a pattern $\mathcal{N}_1$, a homomorphism $p:\mathcal{N}_1 \to \mathcal{N}_0$, and an embedding token u for $\mathcal{N}_1$, an *extension* of $(\mathcal{N}_1,p,u)$ is a triple $(\mathcal{N}_1',p',u')$ where $\mathcal{N}_1$ is a subnetwork of $\mathcal{N}_1'$, $p':\mathcal{N}_1' \to \mathcal{N}_0$ is a homomorphism such that $p = p'|_{\mathcal{N}_1}$, and u' is an embedding token for $\mathcal{N}_1'$ such that $u = u'|_{\mathcal{N}_1}$.

Definition. $(\mathcal{N}_1',p',u')$ is a *minimal* extension of $(\mathcal{N}_1,p,u)$ satisfying a condition P iff

(i) $(\mathcal{N}_1',p',u')$ is an extension of $(\mathcal{N}_1,p,u)$ satisfying P;
(ii) for any extension $(\mathcal{N}_1'',p'',u'')$ of $(\mathcal{N}_1,p,u)$ satisfying P, $|\mathcal{N}_1'| \le |\mathcal{N}_1''|$,

where the cardinality $|\mathcal{N}|$ of a network $\mathcal{N}$ is defined by $|(\Sigma,N,H,E,K,W,P,A,F,S,C,G)| = |\Sigma| + |N| + |H| + |E| + |K|$.

Definition. An extension is *trivial* iff $\mathcal{N}_1' = \mathcal{N}_1$.

Using these notions we can define the extension operations used in the algorithm. The notation is as usual: we have a grammar $\mathcal{N}_0 = (\Sigma_0,N_0,H_0,E_0,K_0,W_0,P_0,A_0,F_0,S_0,C_0,G_0)$ and an embedding type $v = (sub,con,rel,symm,tem,in)$ on $\mathcal{N}_0$, a pattern $\mathcal{N}_1 = (\Sigma_1,N_1,H_1,E_1,K_1,W_1,P_1,A_1,F_1,S_1,C_1,G_1)$ with a parse $p:\mathcal{N}_1 \to \mathcal{N}_0$, an embedding token u, a pair of inclusion functions (i,j), and a bareness function B for $\mathcal{N}_1$; we shall produce an extended network $\mathcal{N}_1' = (\Sigma_1',N_1',H_1',E_1',K_1',W_1',P_1',A_1',F_1',S_1',C_1',G_1')$ with a new parse $p':\mathcal{N}_1' \to \mathcal{N}_0$ and embedding token u'. The threshold θ is the positive constant used in the *DM* function (see §6.2); the thresholds $\theta_0,\theta_1,\theta_2,\theta_3,\theta_4$ are used only in the extension operations and may be made dependent on parameters such as temperature.

In the extension operations the following additional conditions, referred to collectively as *the extension conditions*, will be imposed.

- $\forall\sigma\in\Sigma_1 \; (P_1'^{-1}(\{\sigma\}) \not\subseteq N_1 \;\Rightarrow\; j(\sigma) > \theta_0 \wedge B(\sigma) < \theta)$,
- $\forall h\in H_1 \; (F_1'^{-1}(\{h\}) \cup S_1'^{-1}(\{h\}) \not\subseteq E_1 \;\Rightarrow\; j(h) > \theta_0)$,
- $\forall e\in E_1' \; (A_1'(F_1'(e)) \notin N_1 \vee A_1'(S_1'(e)) \notin N_1 \;\Rightarrow\; W_1'(A_1'(F_1'(e))) = W_1'(A_1'(S_1'(e))))$,
- $\forall n \in N_1'\backslash N_1 \;\; E_{con(p'(n))}(u'(W_1'(n))^{-1} \cdot u'(n)) < \theta_1$,
- the symmetry condition for $\mathcal{N}_1', u', v \circ p'$.

The *extension operations* are as follows (see figure 8 below). For each one, the extension conditions are checked at the end and the operation is cancelled if they do not hold.

(a) (Joining two symbols.) Given symbols $\sigma_1,\sigma_2\in\Sigma_1$, an edge $e_0\in E_0$, and affine transformations $s_1\in symm(p(\sigma_1))$ and $s_2\in symm(p(\sigma_2))$ such that

- $P_0(A_0(F_0(e_0))) = p(\sigma_1)$ and $P_0(A_0(S_0(e_0))) = p(\sigma_2)$,
- $E_{rel(e_0)}(s_2^{-1} \cdot u(\sigma_2)^{-1} \cdot u(\sigma_1) \cdot s_1) < \theta_2$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ (if one exists) such that $\mathcal{N}_1'$ contains nodes n_1, n_2 and an edge e for which

- $p'(e) = e_0$, $A_1'(F_1'(e)) = n_1$, $A_1'(S_1'(e)) = n_2$, $P_1'(n_1) = \sigma_1$ and $P_1'(n_2) = \sigma_2$,
- $u'(n_1) = u(\sigma_1) \cdot s_1$ and $u'(n_2) = u(\sigma_2) \cdot s_2$.

(b) (Joining three symbols.) Given symbols $\sigma_1, \sigma_2, \sigma_3 \in \Sigma_1$, edges $e_{01}, e_{02} \in E_0$, $s_1 \in symm(p(\sigma_1))$, $s_2 \in symm(p(\sigma_2))$ and $s_3 \in symm(p(\sigma_3))$ such that

- $A_0(S_0(e_{01})) = A_0(F_0(e_{02}))$, $P_0(A_0(F_0(e_{01}))) = p(\sigma_1)$, $P_0(A_0(S_0(e_{01}))) = p(\sigma_2)$ and $P_0(A_0(S_0(e_{02}))) = p(\sigma_3)$,
- $E_{rel(e_{01})}(s_2^{-1} \cdot u(\sigma_2)^{-1} \cdot u(\sigma_1) \cdot s_1) + E_{rel(e_{02})}(s_3^{-1} \cdot u(\sigma_3)^{-1} \cdot u(\sigma_2) \cdot s_2) < \theta_3$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ (if one exists) such that $\mathcal{N}_1'$ contains nodes n_1, n_2, n_3 and edges e_1, e_2 for which

- $p'(e_1) = e_{01}$, $p'(e_2) = e_{02}$, $A_1'(F_1'(e_1)) = n_1$, $A_1'(S_1'(e_1)) = n_2 = A_1'(F_1'(e_2))$, $A_1'(S_1'(e_2)) = n_3$, $P_1'(n_1) = \sigma_1$, $P_1'(n_2) = \sigma_2$ and $P_1'(n_3) = \sigma_3$,
- $u'(n_1) = u(\sigma_1) \cdot s_1$, $u'(n_2) = u(\sigma_2) \cdot s_2$ and $u'(n_3) = u(\sigma_3) \cdot s_3$.

(There are also variations of operation (b), in which the directions of e_{01} and e_1 are reversed, or the directions of e_{02} and e_2 are reversed.)

(c) (Extending from a hook.) Given a hook $h \in H_1$, construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that

- $p'(F_1'^{-1}(\{h\})) = F_0^{-1}(p(\{h\}))$ and $p'(S_1'^{-1}(\{h\})) = S_0^{-1}(p(\{h\}))$,
- $\forall e \in E_1' \backslash E_1 \;\; E_{rel(p'(e))}(u'(A_1'(S_1'(e)))^{-1} \cdot u'(A_1'(F_1'(e)))) = 0$.

(d) (Extending from a facet.) Given a facet $k \in K_1$ such that

- $W_1(A_1(F_1(C_1(k)))) = W_1(A_1(S_1(C_1(k))))$
- $j(k) > \theta_0$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that

- $\overline{p'} \circ G_1'^{-1} \circ id_{\{k\}} = G_0^{-1} \circ \overline{p} \circ id_{\{k\}}$.

(e) (Extending from a part to a whole.) Given a symbol $\sigma \in \Sigma_1$, construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that

- $p'(P_1'^{-1}(\{\sigma\})) = P_0{}^{-1}(p(\{\sigma\}))$,
- $\forall n \in N_1' \backslash N_1 \; E_{con(p'(n))}(u'(W_1'(n))^{-1} \cdot u'(n)) = 0$.

(f) (Filling in a missing part between two parts.) Given nodes $n_1, n_3 \in N_1$, hooks $h_1, h_3 \in H_1$, edges $e_{01}, e_{02} \in E_0$ and an affine transformation f such that

- $W_1(n_1) = W_1(n_3)$, $A_1(h_1) = n_1$, $A_1(h_3) = n_3$, $A_0(S_0(e_{01})) = A_0(F_0(e_{02}))$, $F_0(e_{01}) = p(h_1)$ and $S_0(e_{02}) = p(h_3)$,
- $E_{rel(e_{01})}(f^{-1} \cdot u(n_1)) + E_{rel(e_{02})}(u(n_3)^{-1} \cdot f) < \theta_4$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ (if one exists) such that $\mathcal{N}_1'$ contains edges e_1, e_2 and a node n_2 for which

- $p'(e_1) = e_{01}$, $p'(e_2) = e_{02}$, $F'_1(e_1) = h_1$, $A'_1(S'_1(e_1)) = n_2 = A'_1(F'_1(e_2))$ and $S'_1(e_2) = h_3$,
- if $n_2 \notin N_1$ then $u'(n_2) = f$.

(There are also variations of operation (f), in which the directions of e_{01} and e_1 are reversed, or the directions of e_{02} and e_2 are reversed.)

(g) (Filling in a symbol between part and whole.) Given symbols $\sigma_1, \sigma_3 \in \Sigma_1$, symbols $\sigma_{01}, \sigma_{02}, \sigma_{03} \in \Sigma_0$, nodes $n_{01}, n_{02} \in N_0$, $s \in symm(\sigma_{03})$ and an affine transformation f such that

- $p(\sigma_1) = \sigma_{01}$, $p(\sigma_3) = \sigma_{03}$, $W_0(n_{01}) = \sigma_{01}$, $P_0(n_{01}) = \sigma_{02} = W_0(n_{02})$ and $P_0(n_{02}) = \sigma_{03}$,
- $E_{con(n_{01})}(u(\sigma_1)^{-1} \cdot f) + E_{con(n_{02})}(f^{-1} \cdot u(\sigma_3) \cdot s) < \theta_4$,

construct a minimal extension $(\mathcal{N}'_1, p', u')$ of $(\mathcal{N}_1, p, u)$ such that $\mathcal{N}'_1$ contains a symbol σ_2 and nodes n_1, n_2 for which

- $p'(n_1) = n_{01}$, $p'(n_2) = n_{02}$, $W'_1(n_1) = \sigma_1$, $P'_1(n_1) = \sigma_2 = W_1(n_2)$ and $P'_1(n_2) = \sigma_3$,
- if $n_1 \notin N_1$ then $u'(n_1) = u'(\sigma_2) = f$,
- if $n_2 \notin N_1$ then $u'(n_2) = u(\sigma_3) \cdot s$.

These extension operations are applied concurrently, in a fair order, controlled by probabilities; the probability is low in cases where new symbols would be created (particularly operations (a) and (e)), to avoid the creation of too many new symbols.

These operations are depicted in figure 8. As in previous figures, rectangles represent symbols, circles represent nodes, small filled discs represent hooks, lines with arrowheads halfway along represent edges, and small crosses represent facets (shown only for operation (e)); the W and P functions, which map each node to the whole and part symbols, are depicted by green and blue arrows. For each operation solid lines are used for the symbols, nodes, etc., assumed to be present in the pattern before the extension operation; dashed lines are used for the symbols, nodes, etc., added by the extension operation (if they are not already present in $\mathcal{N}_1$). Thus, for example, operation (g) adds one symbol and two nodes (and their associated hooks), unless a suitable symbol or suitable nodes already exist in $\mathcal{N}_1$. Note, however, that whenever an edge is added the appropriate number of superedges must also be added, in order that p' satisfy the conditions for a homomorphism; and the same applies to symbols, nodes, etc.; these are not shown in the figure.

Definition. $(\mathcal{N}_1, p, u)$ is *inextendable*, given i, j, v, B, iff none of the extension operations can be applied to it, other than ones giving a trivial extension.

8.3 Merging two symbol tokens

We may merge two symbol tokens of the same type that have similar embedding transformations (up to a symmetry transformation). Two symbol tokens $\sigma_1, \sigma_2 \in \Sigma_1$ of type $\sigma_0 \in \Sigma_0$ are considered to have similar embeddings up to a symmetry transformation iff there exists $s \in symm(\sigma_0)$ such that $E_{in(\sigma_0)}(s^{-1} \cdot u(\sigma_2)^{-1} \cdot u(\sigma_1))$ is below a threshold. If this condition holds then the symmetry s is applied to σ_2 (this is a local symmetry operation; the other symbols are unchanged); σ_1 and σ_2 are replaced by a single symbol; and the nodes and edges of σ_1 and σ_2 are pooled. (See [50, §8.6] for the formal definition.)

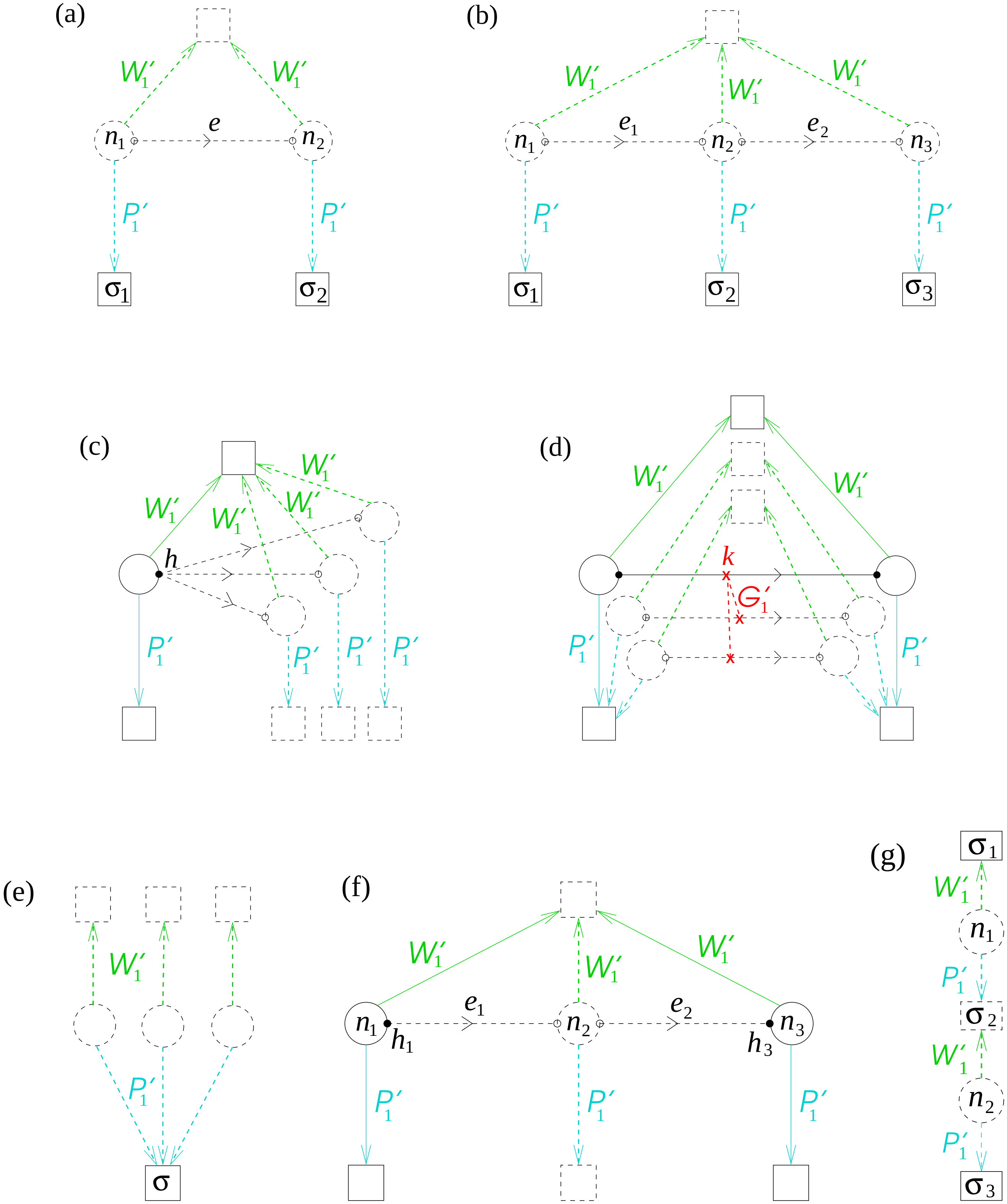


Figure 8. The extension operations. The figure shows the relevant parts of the pattern $\mathcal{N}_1'$ after each extension operation. Solid lines depict what must be present before the operation; dashed lines depict what is added if not already present.

8.4 Partitioning a symbol token into two

For any $\sigma_0 \in \Sigma_1$, if the nodes and symbol in $W_1{}^{-1}(\{\sigma_0\})$ can be partitioned into two disjoint non-empty subsets T_1, T_2, such that there is no edge between any element of T_1 and any element of T_2, then σ_0 may be replaced by two symbols, σ_1, σ_2, with σ_1 getting the nodes of T_1 and σ_2 getting the nodes of T_2. The subsymbols of σ_0 are glued to σ_1 or σ_2 as appropriate (or duplicated if necessary). The nodes, hooks, edges and facets above σ_0 must be duplicated. This operation is called *partitioning* σ_0 [50, §8.7]. It is roughly the inverse of the merging operation.

Definition. In a network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, a symbol $\sigma_0 \in \Sigma$ is *partitionable* iff there exist sets T_1, T_2 such that $T_1 \cup T_2 = W^{-1}(\{\sigma_0\})$ and $T_1 \cap T_2 = \emptyset$ and $T_1, T_2 \neq \emptyset$ and $F^{-1}(A^{-1}(T_1)) \cap S^{-1}(A^{-1}(T_2)) = \emptyset$ and $S^{-1}(A^{-1}(T_1)) \cap F^{-1}(A^{-1}(T_2)) = \emptyset$. The network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is *partitionable* iff at least one symbol in Σ is partitionable; otherwise it is *unpartitionable*.

If no more partitions are possible then we are one step forward in attaining a definite pattern, as the following theorem shows.

Theorem 46 (from [50]). If $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is an unpartitionable network satisfying the condition $W \circ A \circ F = W \circ A \circ S$ then $E \underset{A \circ S}{\overset{A \circ F}{\rightrightarrows}} N \cup \Sigma \xrightarrow{W} \Sigma$ is a coequaliser diagram in the category of sets.

9. The whole recognition process

There are a few details to add, and then I shall summarise the whole recognition algorithm. As usual, the grammar is $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$, the pattern is $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$, and the parse is $p{:}\mathcal{N}_1 \to \mathcal{N}_0$.

9.1 Line operations

To get the process started there are some operations that introduce symbol tokens of type *line* into the pattern (and possibly *bar* symbol tokens too – see §10). These are as follows.

(a) Create a line. A random search is used to find an initial embedding $u(\sigma)$ with a high value of $\rho_{I,T,sat}(u(\sigma))$

(b) Randomise a line's embedding, when it is bare and its inclusion value falls too low.

(c) Remove a bare line, if its temperature falls below a threshold. This operation has a small probability. The purpose of this is to tidy up the pattern by removing excess lines.

(d) Glue two lines together, end to end: the two lines are replaced by one; their bars are connected together, end to end.

9.2 How temperature is determined

Every symbol token, node token, hook token, edge token and facet token has a temperature. These change continually by the following three processes. (The notation $X \Rightarrow Y$ means the assignment statement $Y := \max(X, Y)$, and $X \Leftrightarrow Y$ means $X \Rightarrow Y$ and $Y \Rightarrow X$. Typical values for the spreading parameters are $\beta = 0.3$ and $\gamma = 0.2$.)

1. Every symbol, node, hook, edge or facet token created by a line operation or an extension operation is given a high initial temperature.

2. Temperature is spread through the pattern by applying the following operations periodically.

- For every hook $h \in H_1$, do $T_{A_1(h)} \Rrightarrow T_h$; $\beta T_h \Rrightarrow T_{A_1(h)}$.
- For every edge $e \in E_1$, do $T_{F_1(e)} \Leftrightarrow T_e$; $T_{S_1(e)} \Leftrightarrow T_e$.
- For every node $n \in N_1$, do
  $T_{P_1(n)} \Rrightarrow T_n$; $\gamma T_n \Rrightarrow T_{W_1(n)}$;
  if $n \in \mathrm{dom}(G_1)$ then $\gamma T_{W_1(n)} \Rrightarrow T_n$ else $\gamma T_n \Rrightarrow T_{P_1(n)}$.
- For every pair $k, k^* \in K_1$ such that $G_1(k, k^*)$, do $T_k \Leftrightarrow T_{k^*}$.
- For each $k \in K_1$, do $T_{C_1(k)} \Leftrightarrow T_k$.

3. Periodically each temperature T declines by the formula

$$T := a + \eta T$$

   where the constant η is slightly below 1 and the constant a is small and positive.

Consequently active regions of the pattern will have high temperature; regions that have settled down will have temperature converging to $T_{min} = a/(1-\eta)$.

9.3 How the bareness function B is determined

The purpose of the bareness function is to prevent the algorithm from getting stuck by doing the same operations repeatedly. The bareness values $B(\sigma), B(h), B(k)$ increase monotonically to the maximum θ, inhibiting repeated extension operations at the same place.

For each $h \in H_1$, $B(h)$ is initially 0 and increases by a fixed amount every time an extension operation is applied that adds edges to ${F_1}^{-1}(\{h\}) \cup {S_1}^{-1}(\{h\})$. This makes the algorithm more and more unwilling to remove all the edges and leave the hook bare.

For each $k \in K_1$, $B(k)$ is initially 0 and increases by a fixed amount every time an extension operation is applied that adds facets to $\{\, k^* \in K_1 \mid G_1(k, k^*)\,\}$. This prevents the algorithm from repeatedly adding and removing sub-facets to k forever.

For each $\sigma \in \Sigma_1$, if ${P_0}^{-1}(p(\{\sigma\})) = \emptyset$ then $B(\sigma)$ is initially set to θ, and never changes thereafter. If ${P_0}^{-1}(p(\{\sigma\})) \neq \emptyset$ then $B(\sigma)$ is initially set to a positive value $\theta_0 < \theta$; $B(\sigma)$ is increased by a fixed amount every time extension operation (a), (b), (e) or (g) is applied that adds nodes to ${P_1}^{-1}(\{\sigma\})$. The increment is chosen so that $\theta - \theta_0$ is a multiple of the increment. Ultimately every symbol token σ has $B(\sigma) = \theta$ [50, theorem 47]. This prevents these structural operations from being repeated indefinitely.

9.4 Summary of the entire recognition process

The input is an image I, a semi-definite network $\mathcal{N}_0$ (the grammar), and an embedding type v for $\mathcal{N}_0$ (defining all the geometric relationships in the grammar).

The recognition process is a sequence of steps, called *cycles*. In each cycle,

- i and j are recalculated (§7.4);
- u is adjusted by one step (subject to the symmetry condition) to increase $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$ (§7.3);
- all temperatures spread and decline a little (§9.2);
- structural operations are applied to $\mathcal{N}_1$ if the conditions are satisfied (elementary pruning operations, extension operations, merging two symbol tokens, and partitioning a symbol token into two); every new symbol, node, hook, edge or facet is given a high temperature (§9.2), and some bareness values are increased after an extension operation (§9.3);

- line operations are applied to $\mathcal{N}_1$ (§9.1).

The algorithm halts when

- no further structural operations are possible (except for trivial extensions);
- the temperatures have declined very close to the minimum T_{min}.

9.5 The outcome of the recognition algorithm

When recognition has finished the following have been constructed:

- a pattern $\mathcal{N}_1$,
- a parsing homomorphism $p{:}\mathcal{N}_1 \to \mathcal{N}_0$,
- a pair of inclusion functions (i,j) on $\mathcal{N}_1$,
- an embedding token u for $\mathcal{N}_1$,
- a bareness function B for $\mathcal{N}_1$.

The condition

$$\forall\sigma{\in}\Sigma_1\ B(\sigma) \leq \theta,\ \text{with equality if } {P_0}^{-1}(p(\{\sigma\})) = \emptyset$$

will hold, because it holds all the time. The condition

$$\forall x{\in}\Sigma_1 \cup N_1 \cup E_1\ i(x) \in \{0,1\}$$

is likely to hold, because all temperatures have reduced to a very low value T_{min}. Also, the condition

$$W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1$$

is likely to hold. This is because the final term in the definition of *IM* penalises incoherent edges (recall the discussion of this in §7.2). The penalty is large and its effect is amplified when temperature is very low; an incoherent edge will either pull the two wholes together until they are merged or be pruned.

If we assume that all three of these conditions hold we can apply the following theorem.

Theorem 48 (from [50]). If

(a) $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is a network satisfying $W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1$,

(b) $p{:}\mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, where $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ is a semi-definite network,

(c) (i,j) is a pair of inclusion functions on $\mathcal{N}_1$ satisfying $\forall x{\in}\Sigma_1 \cup N_1 \cup E_1\ i(x) \in \{0,1\}$,

(d) u is an embedding token for $\mathcal{N}_1$ and v is an embedding type for $\mathcal{N}_0$,

(e) B is a bareness function for $\mathcal{N}_1$ satisfying $\forall\sigma{\in}\Sigma_1\ B(\sigma) \leq \theta,$ with equality if ${P_0}^{-1}(p(\{\sigma\})) = \emptyset$,

(f) $\mathcal{N}_1$ is unprunable, given (i,j),

(g) $(\mathcal{N}_1, p, u)$ is inextendable, given i, j, v, B,

(h) $\mathcal{N}_1$ is unpartitionable,

then

(1) $\mathcal{N}_1$ is definite,

(2) $\forall x{\in}\Sigma_1 \cup N_1 \cup E_1\ i(x) = 1$,

(3) $IM(I, \mathcal{N}, u, v, i, j, B) = DM(I, \mathcal{N}, u, v)$.

Proof. This uses theorems from [50].

Theorem 46, using hypotheses (a,h), tells us that

(i) $E_1 \overset{A_1 \circ F_1}{\underset{A_1 \circ S_1}{\rightrightarrows}} N_1 \cup \Sigma_1 \overset{W_1}{\to} \Sigma_1$ is a coequaliser diagram in the category of sets.

Theorem 42, using hypotheses (c,f), gives

(j) $\forall e \in E_1 \; i(e) = 1$.

So theorem 45, using hypotheses (a,c,g) and (j), gives

(k) $\forall h \in H_1 \; j(h) = 0$ and $\forall k \in K_1 \setminus \mathrm{dom}(G_1) \; j(k) = 0$.

Then theorem 44, using hypotheses (c,f) and (i,k), gives conclusions (1,2).

Theorem 47, using hypotheses (a,e,g) and (1,2), then gives

(l) $\forall \sigma \in \Sigma_1 \setminus P_1(N_1) \; B(\sigma) = \theta$.

Conclusion (3) then follows by theorem 33, using (l) and (1,2). ∎

Thus the outcome is a definite pattern. The algorithm has sought to maximise $IM(I, \mathcal{N}, u, v, i, j, B)$, and hence to maximise $DM(I, \mathcal{N}, u, v)$ at the end. The symmetry condition is enforced throughout. The recognition problem is solved.

9.6 What can go wrong

The core of the theory is provably correct, but the more peripheral parts of the algorithm are supported only by heuristic arguments (which I have indicated throughout by use of the word 'likely').

(i) The algorithm for determining the inclusion functions (§7.4) is not guaranteed to halt. It finds only a local maximum of E, not a global maximum.

(ii) Even for low temperature, maximising E is not precisely the same as maximising $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$.

(iii) The monotonic raising of the bareness function may cut off possible structural extensions prematurely.

(iv) Symbol tokens may be merged or not merged, or partitioned or not partitioned, wrongly. (Errors in merging can be corrected by partitioning and vice versa.)

(v) The whole recognition algorithm is not guaranteed to halt.

(vi) It is not guaranteed that very low temperature will force all inclusion values to 0 or 1.

(vii) Incoherent edges may survive to the end.

(viii) Some portions of the image may have been overlooked and not covered by the pattern; perhaps they are noise, but perhaps not.

In practice it is only (iii), (iv), (v) and (viii) that matter.

10. Example grammars

I shall give three examples of grammars, designed to illustrate various aspects of iteration, recursion, and the use of subsymbols to enforce syntactic constraints. First, it is useful to distinguish between a 'line' (a one-dimensional line segment) and a 'bar' (a thin rectangle, with a positive width). I shall use both lines and bars together in my grammars. A line plays a role as part of a larger symbol, such as a hexagonal grid; a bar gives a finer-grained representation of the line. We introduce two symbol types, *line* and *bar*, where *line* has *bar* as its sole part. The geometric relationship between the line and the bar may stretch width-wise, to adapt the width of the bar to the image; this variation does not interfere with the variation in the geometric relationship between the line and the rest of the pattern. In the simplest implementation of this idea, each line token has one bar token as its sole part. However, instead I shall allow a line token to be made up of one or more bar tokens, joined end-to-end (see the grammar in figure 10): this is useful for representing bent or broken lines.

10.1 Hexagonal grids (hex)

The first example illustrates two-dimensional iteration (which is something that conventional graph grammars can represent only using context-sensitive production rules, see §2.3). The grammar for the hexagonal grid symbol type *hex*, with its subsymbols *hexagon*, *junctionA* and *junctionB*, was introduced in §3.3, in simplified form. I omitted to say there that there are also edges between the *hexagon*, *junctionA* and *junctionB* subsymbols and each of their nodes, which ensure that each node occurs once per symbol token. More importantly, I made no provision for what happens at the boundary of the grid. To cope with this we insert invisible *dummy* symbol tokens at the boundary of the grid, where the next *line* would be if the grid continued (figure 9).

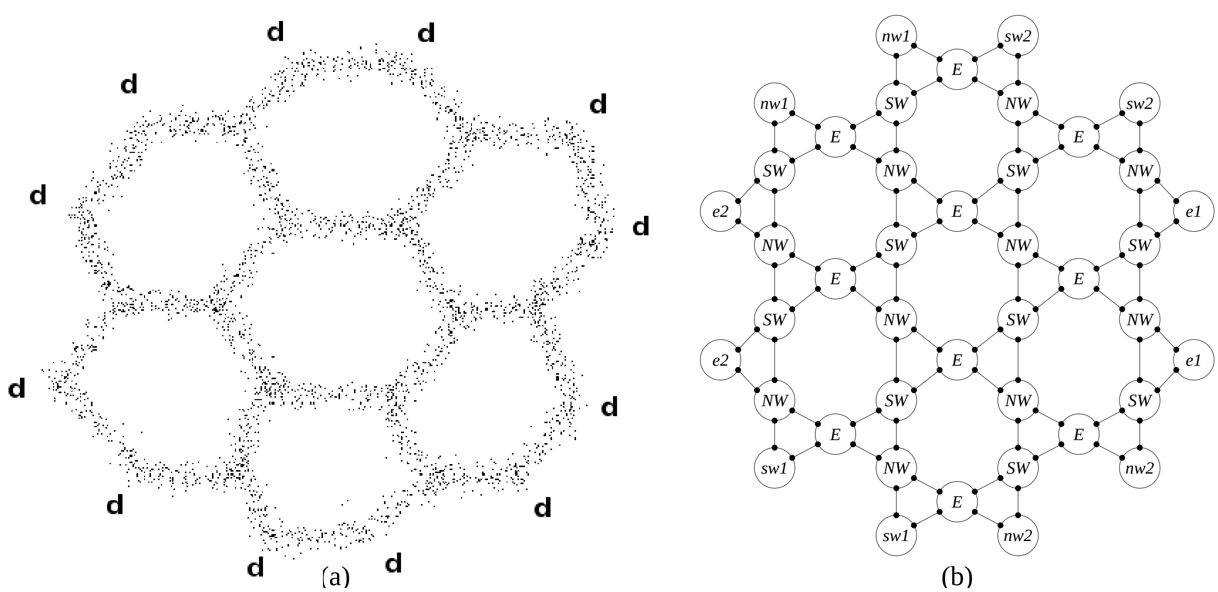


Figure 9. (a) A hexagonal grid image with *dummy* symbol tokens (marked by 'd') inserted at the boundary; (b) The corresponding pattern (only nodes, hooks and edges shown).

The *hex* grammar is then modified accordingly (figure 10). I have labelled the node types E (for east-oriented line), NW (northwest-oriented line), SW (southwest-oriented line), $e1$ (dummy on the east boundary), $e2$ (dummy on the west boundary), etc.; figure 9(b) shows the pattern corresponding to figure 9(a) (showing only the nodes, hooks and edges of the *hex*).

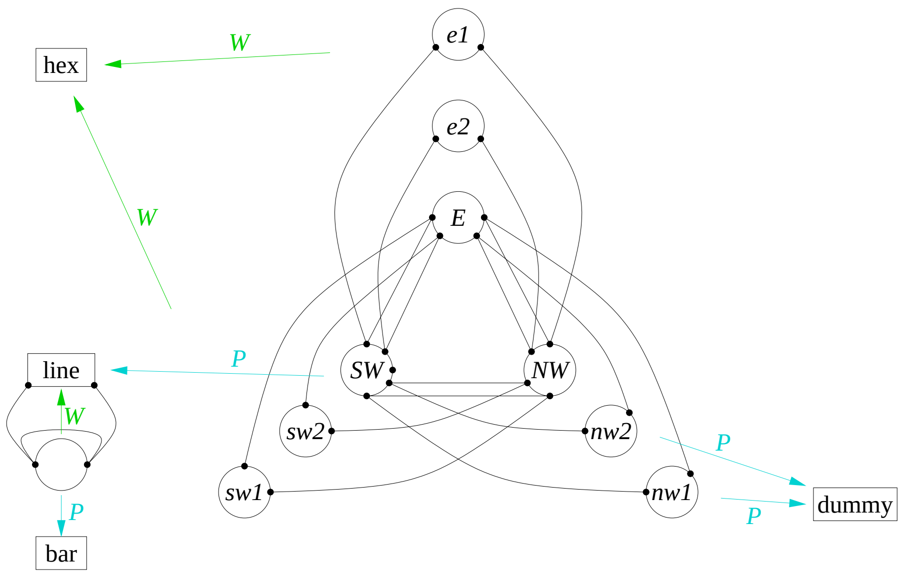


Figure 10. The *hex* grammar with the *dummy* and *bar* symbol types added.

The *junctionA* and *junctionB* subsymbols are modified similarly to allow for the possibility that one of the three lines is a dummy. Several variants of the *hexagon* subsymbol are also needed, to provide for the cases of a hexagon occurring on the boundary or at a corner of the grid.

10.2 Crosses

The next example involves crosses – see figure 11 for an example image. It consists of a central square, with four *trunks* radiating from it, each with the same number of smaller lines called *twigs*. The grammar is shown in figure 12. On the left is the grammar for the symbol type *cross*; the nodes N, E, S, W represent the lines of the square, the nodes NT, ET, ST, WT represent the four trunks, and the nodes Nt, Et, St, Wt represent the twigs. To enforce the constraint that every trunk has the same number of twigs we use a subsymbol 8*twigs*, which includes two adjacent twigs of each trunk. Figure 13 shows the pattern corresponding to the image in figure 11. The presence of 8*twigs* tokens in the pattern is governed by the facets. The facets α–δ in a *cross* token must be glued to corresponding sub-facets in an 8*twigs* token: this is what forces there to exist the required number of 8*twigs* subsymbol tokens, and this forces the number of twigs on each trunk to be equal.

(Again I am omitting edges from the symbols to the nodes, to avoid cluttering the diagrams. There are edges from *cross* to $N, E, S, W, NT, ET, ST, WT$ and from 8*twigs* to each of its nodes. This forces there to be one node token of each type per symbol token.)

10.3 Hnests

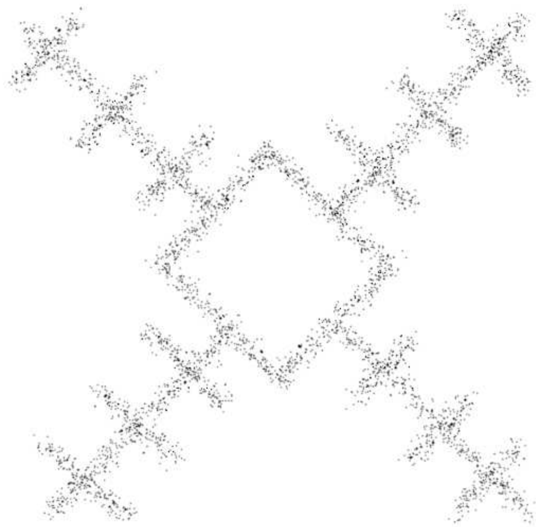


Figure 11. A image showing an example of a cross.

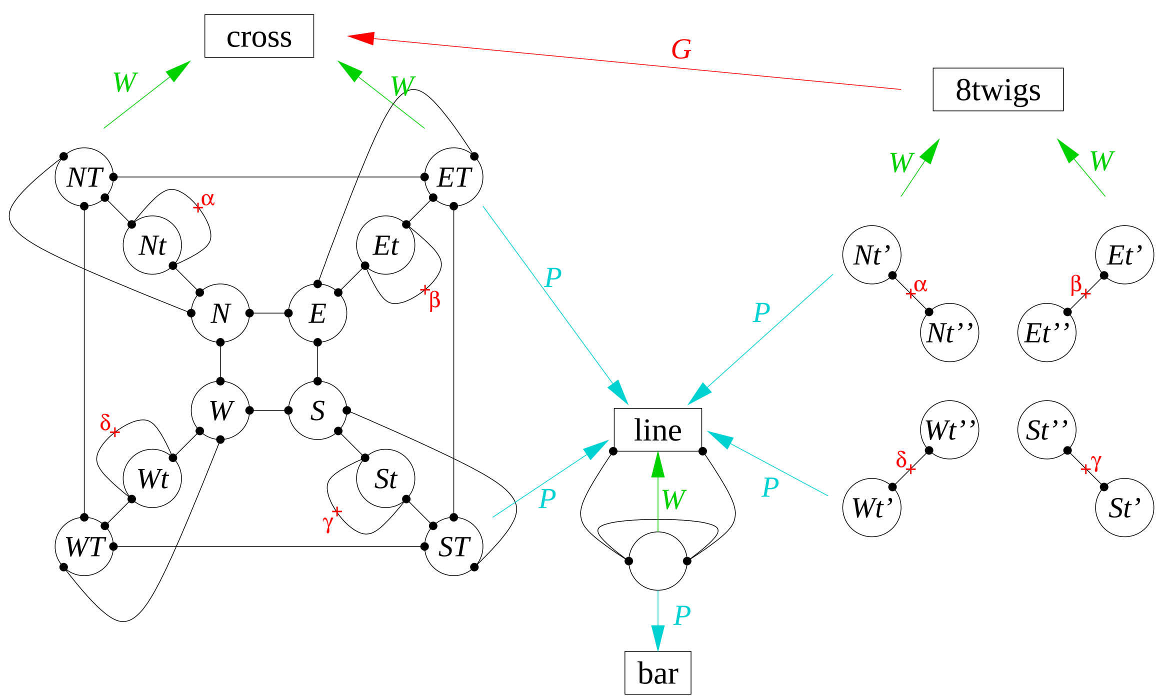


Figure 12. The *cross* grammar. Only a few W and P arrows are shown. 8*twigs* is glued to *cross*; nodes Nt' and Nt'' are glued to node Nt, etc.. The facets are shown as red crosses: α is glued to α, β to β, γ to γ, and δ to δ.

The third example involves nested recursion. Figure 14 shows an example of an *Hnest*, which consists of three lines forming an 'H' and two smaller *Hnests* nested inside it. Figure 15 shows

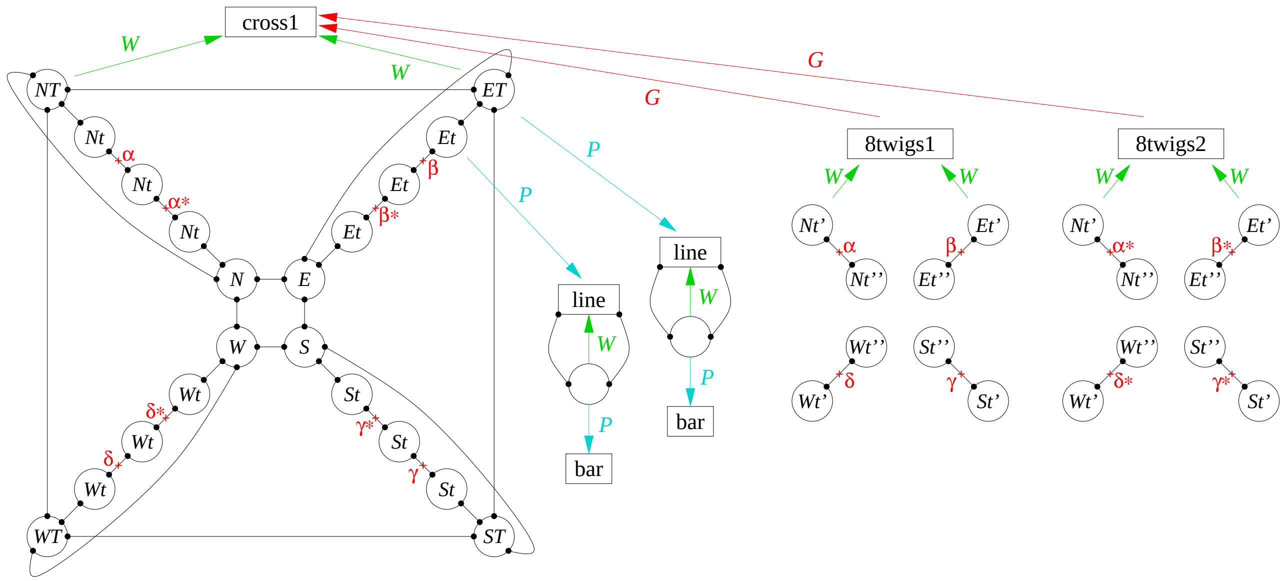


Figure 13. The pattern corresponding to the *cross* image in figure 11. Only two of the *line* tokens are shown. Nodes Nt', Nt'' are glued to node Nt, and so on. Facets $\alpha, \beta, \gamma, \delta, \alpha*, \beta*, \gamma*, \delta*$ are glued to facets $\alpha, \beta, \gamma, \delta, \alpha*, \beta*, \gamma*, \delta*$, respectively.

the grammar. As usual, each node represents a role that a symbol may play as a part of another. The nodes L, cb, R represent a line occurring as left bar, crossbar, and right bar of the 'H', respectively. The nodes T, B represent an *Hnest* occurring as the top or bottom nested *Hnest* of an *Hnest* (note that the *Hnest* symbol type is a part of itself). Every *Hnest* token either has two or no *Hnest* parts. There are no subsymbols.

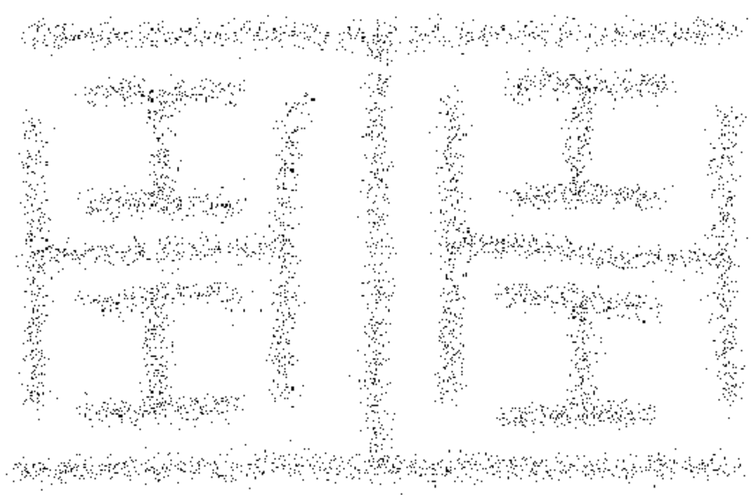


Figure 14. An *Hnest* image.

Figure 16 shows an example of an *Hnest* pattern.

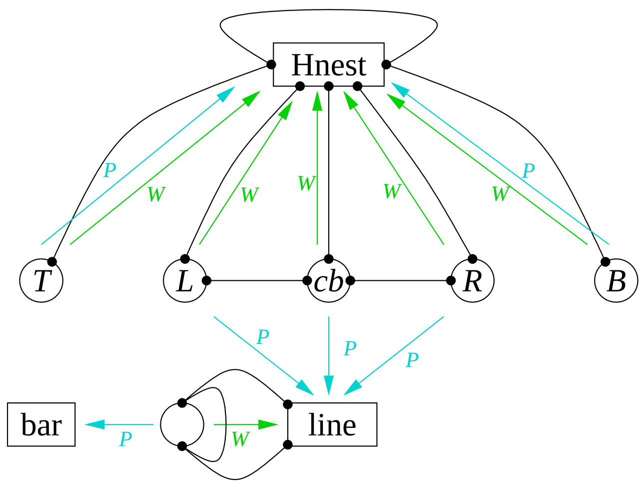


Figure 15. The *Hnest* grammar.

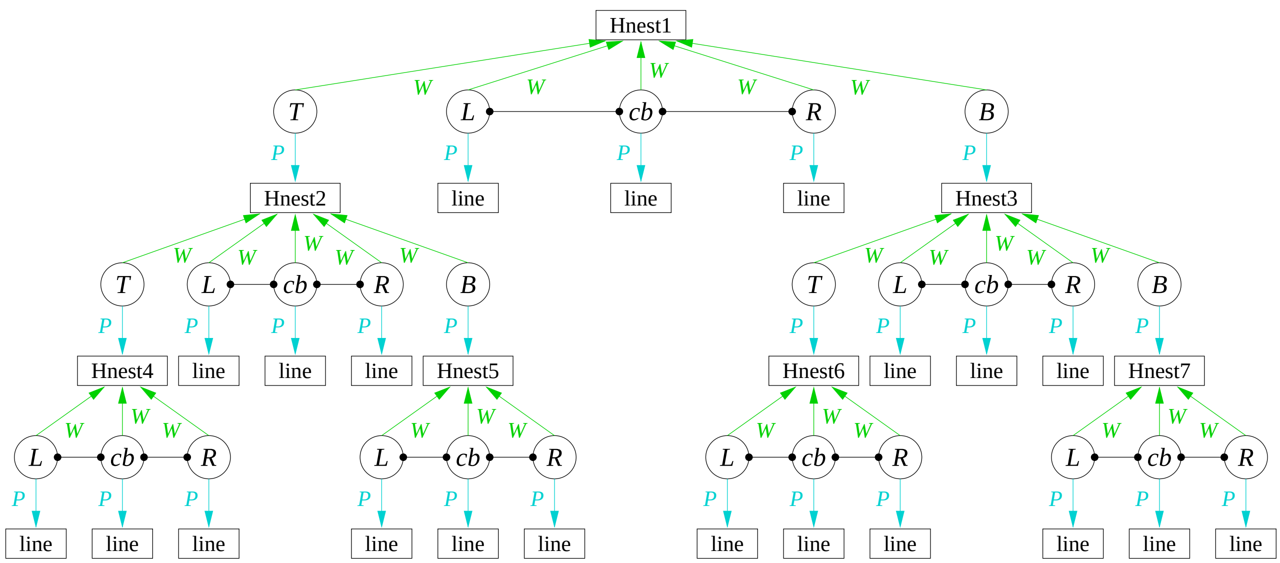


Figure 16. An *Hnest* pattern. Edges between *Hnest* tokens and their nodes are omitted. *Bar* tokens are omitted.

11. Example runs

I shall provide some examples of the algorithm recognising images according to the three grammars described in the previous section (combined together into a single grammar). These example runs are not intended as experiments and I make no quantitative claims; they merely provide proof of concept. The test images were designed to illustrate the recursive capabilities and aspects of error-tolerance listed as desirable in §1.

Test images were produced as follows. First I drew twenty hexes (with 2–4 hexagons on each side), twenty crosses (with 3–6 twigs on each trunk), and twenty Hnests (with 1–4 levels of nesting) by hand; these are irregularly drawn, with bent lines and variable angles and length ratios. The lines are automatically blurred. A composite image is produced by randomly selecting three of these sixty simple images and superimposing them, with random translations (in the range -300 to 300 pixels in each dimension), random rotations (in the range 0 to 2π) and random dilations (in the range 0.9 to 1.3). Repeating this produces an indefinitely large supply of composite images. Corrupted versions are produced by erasing one or two random discs of radius 50 pixels in the composite image.

The reasons for using synthetic images rather than natural ones are that they allow me to emphasise the features I am most concerned with (recursion and iteration, overlapping symbol tokens, geometric variability, indistinct and bent lines, and random erasures), while avoiding the need for special tricks, as often needed with natural images; they avoid issues I am not interested in (perspective, binocular vision, motion, occlusion, colour, texture, shadow and illumination); they allow me to calibrate the difficulty of the image set to match the capabilities of the algorithm; and they allow me to generate an unlimited number of patterns with consistent statistical properties and to test the algorithm under controlled conditions.

The above parameters were chosen to give images sufficiently complex and cluttered to illustrate the desired qualitative features. Varying the number of simple images makes no difference to the statistical properties of the results, so I have not gone beyond sixty; the variety in the composite images comes from the random overlapping of the simple images. Varying the number of composite images also makes no difference. There is sufficient geometric variability in the images to ensure that top-down and bottom-up inference must be used in combination: e.g., a junction between two lines cannot be recognised as belonging to a hex, a cross or an Hnest merely from the angle or ratio of line lengths but must be interpreted in the light of grammatical context.

The difficulty of a composite image depends on how much its three component images overlap, and this may be roughly measured by an *overlap index*, calculated as follows. Fit a circumscribing disc to each of the three component images. The overlap index is defined as the sum of the areas of overlap between every pair of discs, divided by the sum of the areas of the three discs. It ranges between 0 and 1. Another measure of difficulty is the number of symbol tokens in the desired pattern.

The recognition algorithm is run on 300 of these composite images and the results are summarised in tables 1–3, broken down by number of symbols and then by overlap index. The first six columns give the numbers of composite images for which the constructed pattern is topologically correct, subdivided by the number of wrong lines: a line is 'wrong' iff it is misplaced or is part of a misplaced hex, cross or Hnest. The next three columns give the numbers of composite images for which the constructed pattern is not topologically correct, subdivided by the number of component images (hexes, crosses or Hnests) that are not topologically correct (plus the number of spurious hexes, crosses or Hnests, if any). The final three columns give the numbers of composite images for which the algorithm gets stuck, i.e., does not halt, again subdivided by the number of components that are not topologically correct.

Tables 2 and 3 show that the algorithm has some ability to restore erased lines or junctions. However, it tends to restore them in the most grammatically probable place, rather than where they actually were before they were erased. This accounts for the increased number of wrong lines. There is also an increase in the number of cases where the algorithm gets stuck.

The figures show some examples. Figure 17 shows an image consisting of a hex, an Hnest and a cross, with two erased discs (erased discs are outlined in brown in the figures). Note that the Hnest is somewhat sheared. The image is successfully recognised, but with three misplaced lines of an H in one erased disc (this counts as '3 wrong lines' in table 3). The actual positions of the misplaced lines are marked in red and the correct positions in blue.

symbols	wrong lines 0	1	2	3	4	5+	topological error 1	2	3	stuck 1	2	3	total
50–149	22	2	0	0	0	0	3	0	0	0	0	0	27
150–249	74	11	4	1	0	1	10	0	0	2	2	0	105
250–349	27	3	3	2	0	0	3	0	0	1	0	1	40
350–449	23	2	2	2	0	0	1	0	0	3	1	0	34
450–549	16	0	0	2	1	0	1	0	0	6	1	1	28
550–649	20	4	4	2	2	0	7	0	0	2	2	0	43
650–749	3	1	0	1	0	1	1	0	0	1	3	0	11
750–849	0	1	0	0	0	0	0	0	0	0	2	6	9
850–949	0	0	0	0	0	0	0	0	0	0	1	0	1
950–1049	1	0	0	0	1	0	0	0	0	0	0	0	2
total	186	24	13	10	4	2	26	0	0	15	12	8	300

overlap index	wrong lines 0	1	2	3	4	5+	topological error 1	2	3	stuck 1	2	3	total
0–0.2	13	1	0	1	1	0	1	0	0	0	0	0	17
0.2–0.4	90	4	1	3	2	0	4	0	0	3	3	2	112
0.4–0.6	72	11	9	5	1	1	16	0	0	8	8	3	134
0.6–0.8	11	8	3	1	0	1	4	0	0	4	1	3	36
0.8–1	0	0	0	0	0	0	1	0	0	0	0	0	1
total	186	24	13	10	4	2	26	0	0	15	12	8	300

Table 1. Results for images with no erased discs.

symbols	wrong lines 0	1	2	3	4	5+	topological error 1	2	3	stuck 1	2	3	total
50–149	21	2	1	0	0	0	3	0	0	0	0	0	27
150–249	64	15	2	4	1	2	9	0	0	7	1	0	105
250–349	25	7	0	2	0	0	2	0	0	3	0	1	40
350–449	18	5	2	1	0	0	2	0	0	4	2	0	34
450–549	12	3	0	0	1	1	1	0	0	7	2	1	28
550–649	16	4	1	2	2	2	4	1	0	2	9	0	43
650–749	1	3	1	0	0	0	2	0	0	2	2	0	11
750–849	0	0	0	0	1	0	1	0	0	2	0	5	9
850–949	0	0	0	0	0	0	0	0	0	0	0	1	1
950–1049	0	0	0	0	1	1	0	0	0	0	0	0	2
total	157	39	7	9	6	6	24	1	0	27	16	8	300

overlap index	wrong lines 0	1	2	3	4	5+	topological error 1	2	3	stuck 1	2	3	total
0–0.2	14	0	0	1	1	1	0	0	0	0	0	0	17
0.2–0.4	69	14	2	3	2	1	7	0	0	7	6	1	112
0.4–0.6	59	19	5	5	1	3	14	1	0	16	7	4	134
0.6–0.8	15	5	0	0	2	1	3	0	0	4	3	3	36
0.8–1	0	1	0	0	0	0	0	0	0	0	0	0	1
total	157	39	7	9	6	6	24	1	0	27	16	8	300

Table 2. Results for images with one erased disc.

symbols	wrong lines 0	1	2	3	4	5+	topological error 1	2	3	stuck 1	2	3	total
50–149	20	0	0	3	1	0	2	0	0	1	0	0	27
150–249	59	17	7	1	1	1	10	0	0	4	5	0	105
250–349	18	10	1	0	1	0	3	0	0	5	2	0	40
350–449	18	5	1	1	1	1	1	0	0	5	1	0	34
450–549	10	3	0	3	0	0	2	0	0	4	5	1	28
550–649	13	5	4	1	1	1	3	0	0	5	10	0	43
650–749	3	1	1	0	0	0	2	0	0	0	4	0	11
750–849	1	0	0	1	0	0	0	0	0	1	2	4	9
850–949	0	0	0	0	0	0	0	0	0	0	0	1	1
950–1049	0	0	0	0	0	1	0	0	0	0	0	1	2
total	142	41	14	10	5	4	23	0	0	25	29	7	300

overlap index	wrong lines 0	1	2	3	4	5+	topological error 1	2	3	stuck 1	2	3	total
0–0.2	12	2	0	1	1	0	1	0	0	0	0	0	17
0.2–0.4	69	14	5	2	0	1	6	0	0	6	7	2	112
0.4–0.6	55	18	8	5	3	2	12	0	0	14	14	3	134
0.6–0.8	6	7	1	2	1	1	3	0	0	5	8	2	36
0.8–1	0	0	0	0	0	0	1	0	0	0	0	0	1
total	142	41	14	10	5	4	23	0	0	25	29	7	300

Table 3. Results for images with two erased discs.

Four lines correctly guessed in the other erased disc are shown in green: one is part of an H and three are part of a hex.

Figure 18 shows an image consisting of three hexes (I have added an outline to two of the hexes in the figure as an aid to the reader). This is successfully recognised. In a version of this image with one erased disc the hex outlined in red is only partially recognised (this is counted as 'stuck – 1 wrong symbol' in table 2). In a version with two erased discs the hex outlined in blue is recognised but the other two are not separated (this is counted as 'stuck – 2 wrong symbols' in table 3).

Figure 19 shows an image consisting of one hex and two crosses, with one erased disc. By accident the two crosses are oriented identically and positioned so that many of their lines overlap, yet the image is successfully recognised. A version of the image with two erased discs is not successfully recognised: the twigs of the crosses are not sorted out correctly (this counts as 'stuck – 2 wrong symbols' in table 3).

Figure 20 shows an image consisting of three Hnests, with two erased discs. This is recognised with two misplaced lines in a single H (this counts as '2 wrong lines' in table 3). The actual positions of the misplaced lines are marked in red and the correct positions in blue. The other line in the H (correctly recognised) is marked in green.

Figure 21 shows an image consisting of one hex and two Hnests, with one erased disc (the image is truncated in the figure). The hex is incompletely recognised: the lines marked in blue are missed. This is the commonest type of error for a hex. (This counts as 'topological error – 1 wrong symbol' in table 2.)

Figure 22 shows an image consisting of one hex and two Hnests, with two erased discs.

The algorithm recognises only part of the smaller Hnest: the H marked in red is recognised (as an Hnest in its own right), but the parts marked in blue are overlooked. This is a common type of error for an Hnest. (This counts as 'topological error – 1 wrong symbol' in table 3.)

Figure 23 shows how a bent or broken line may be represented by two or three bars joined end to end; four such lines are shown, the bars being marked by red rectangles. Long bent lines would be hard to recognise if this were not allowed. The brown circle outlines an erased disc. The whole image is recognised successfully.

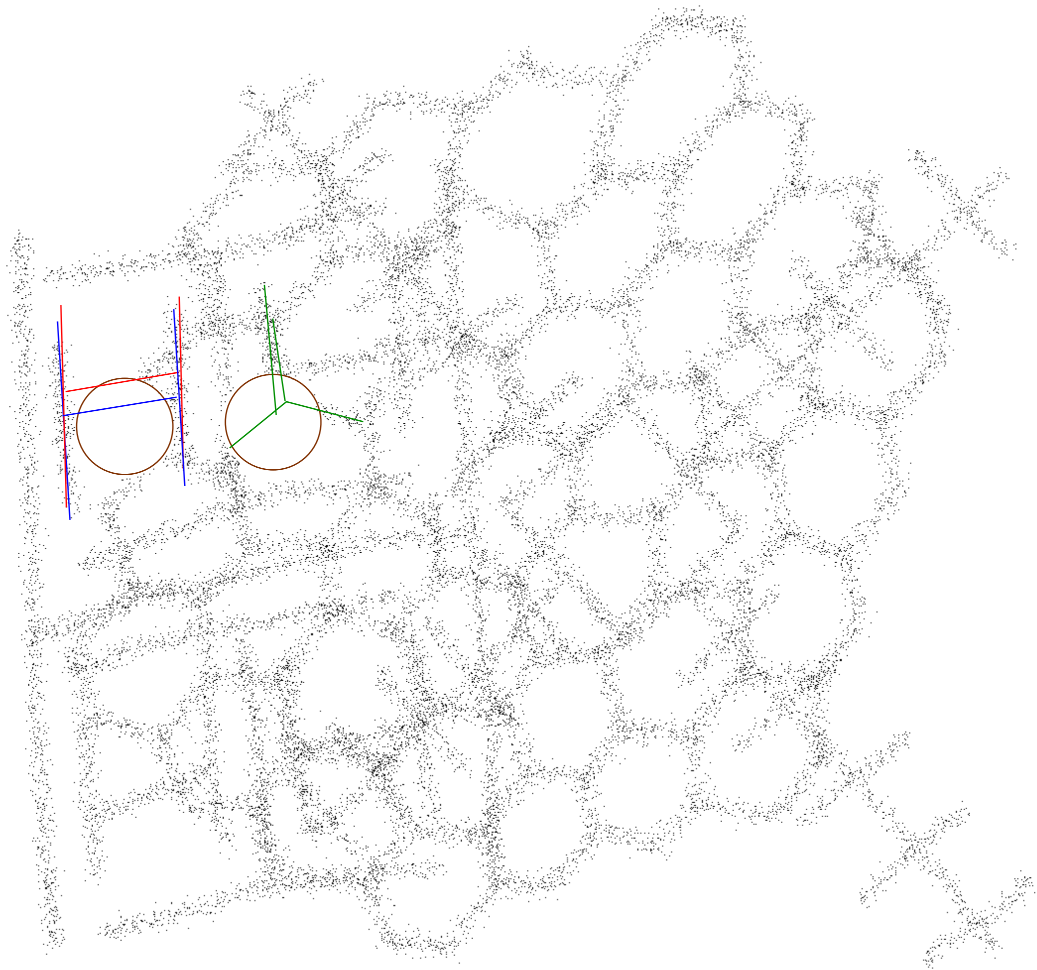


Figure 17. A hex, an Hnest and a cross. 557 symbols, overlap index = 0.666.

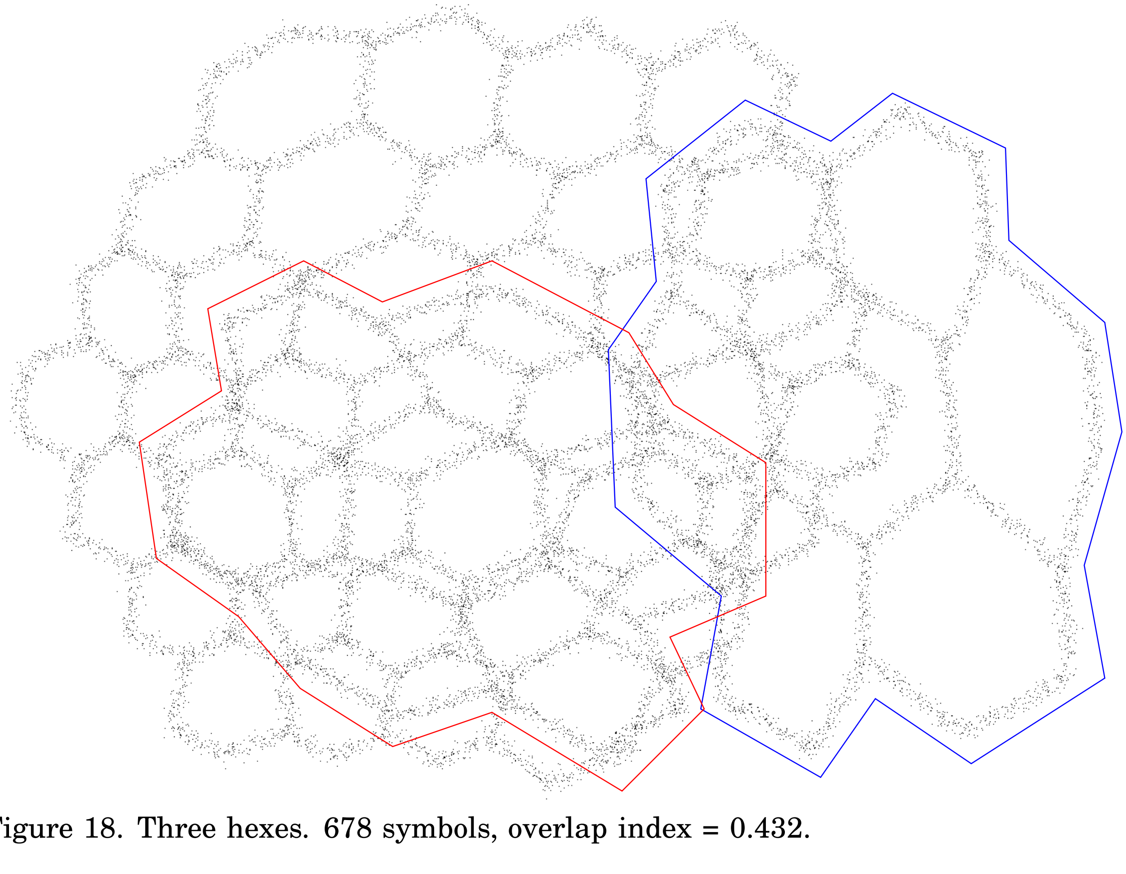


Figure 18. Three hexes. 678 symbols, overlap index = 0.432.

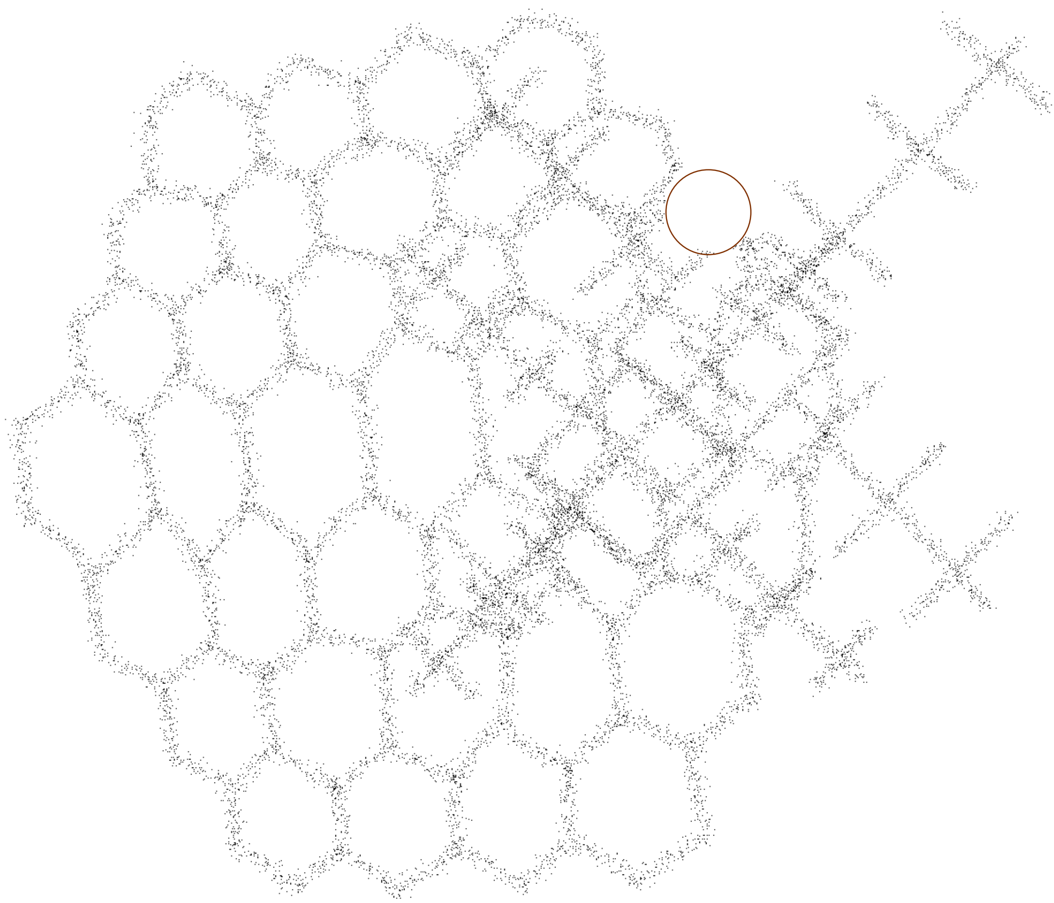


Figure 19. One hex and two crosses. 532 symbols, overlap index = 0.579.

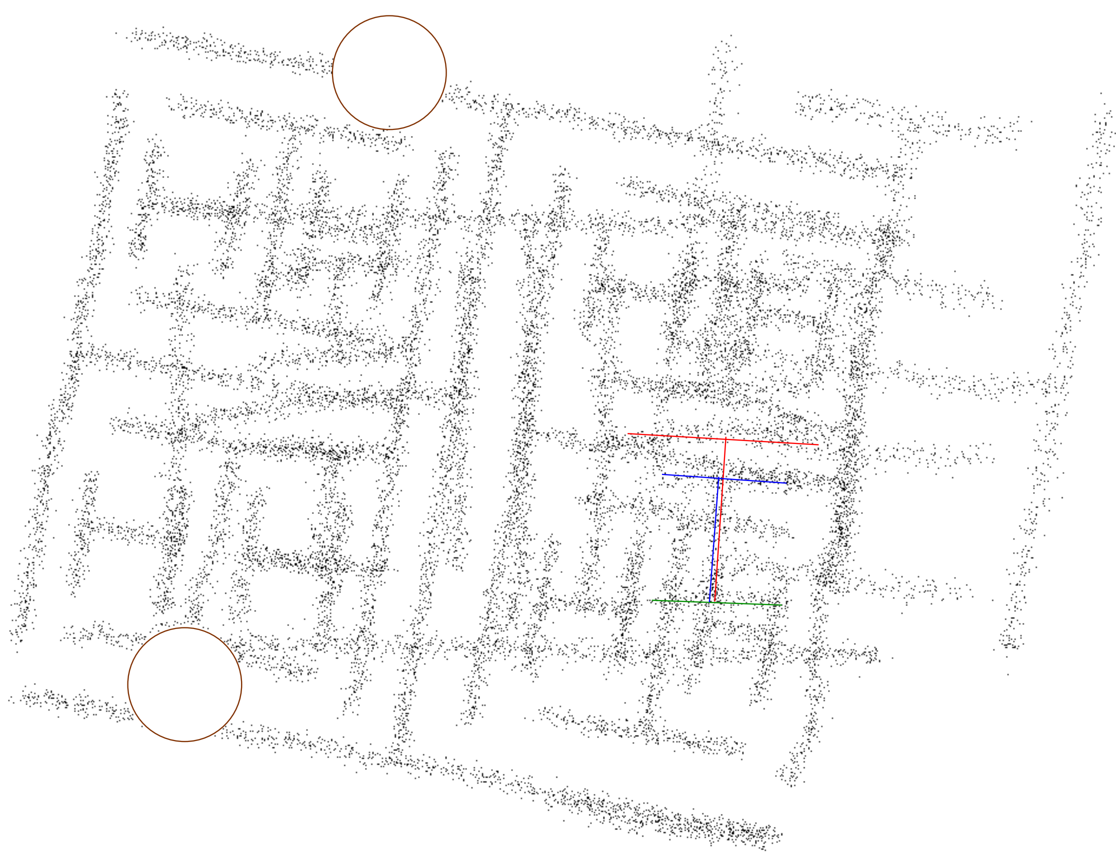


Figure 20. Three Hnests. 175 symbols, overlap index = 0.563.

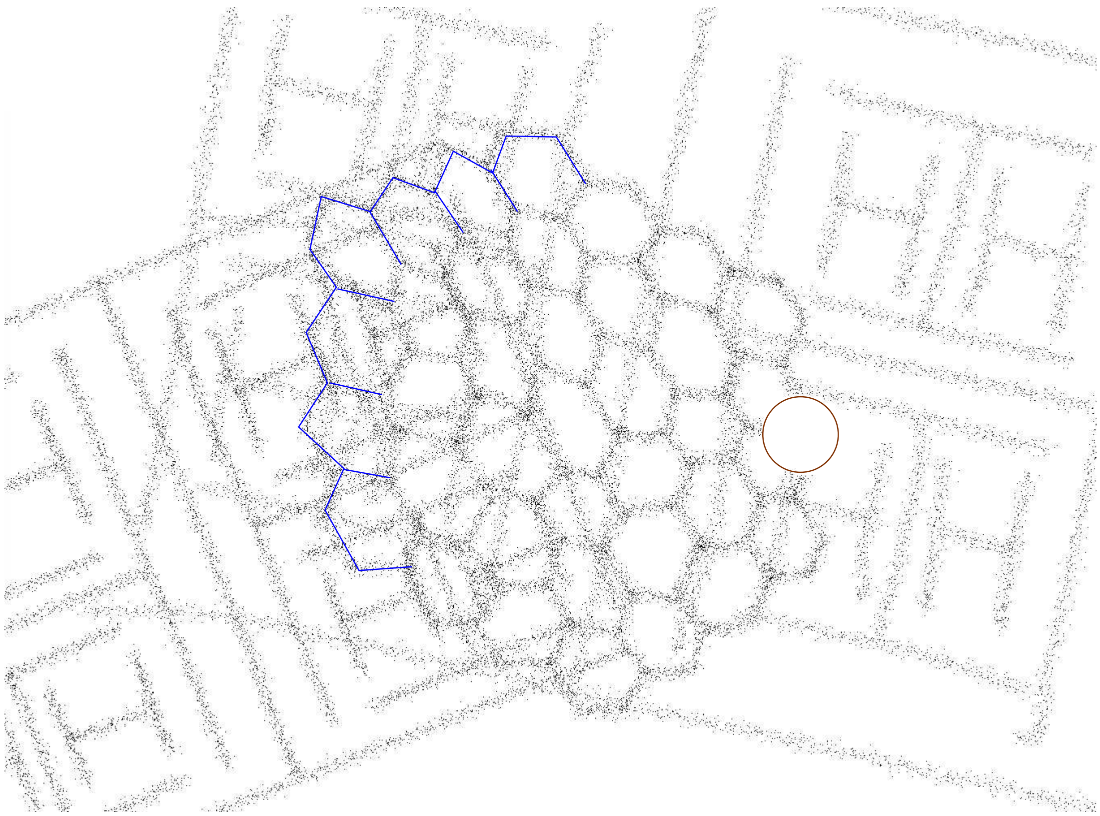


Figure 21. One hex and two Hnests. 662 symbols, overlap index = 0.387.

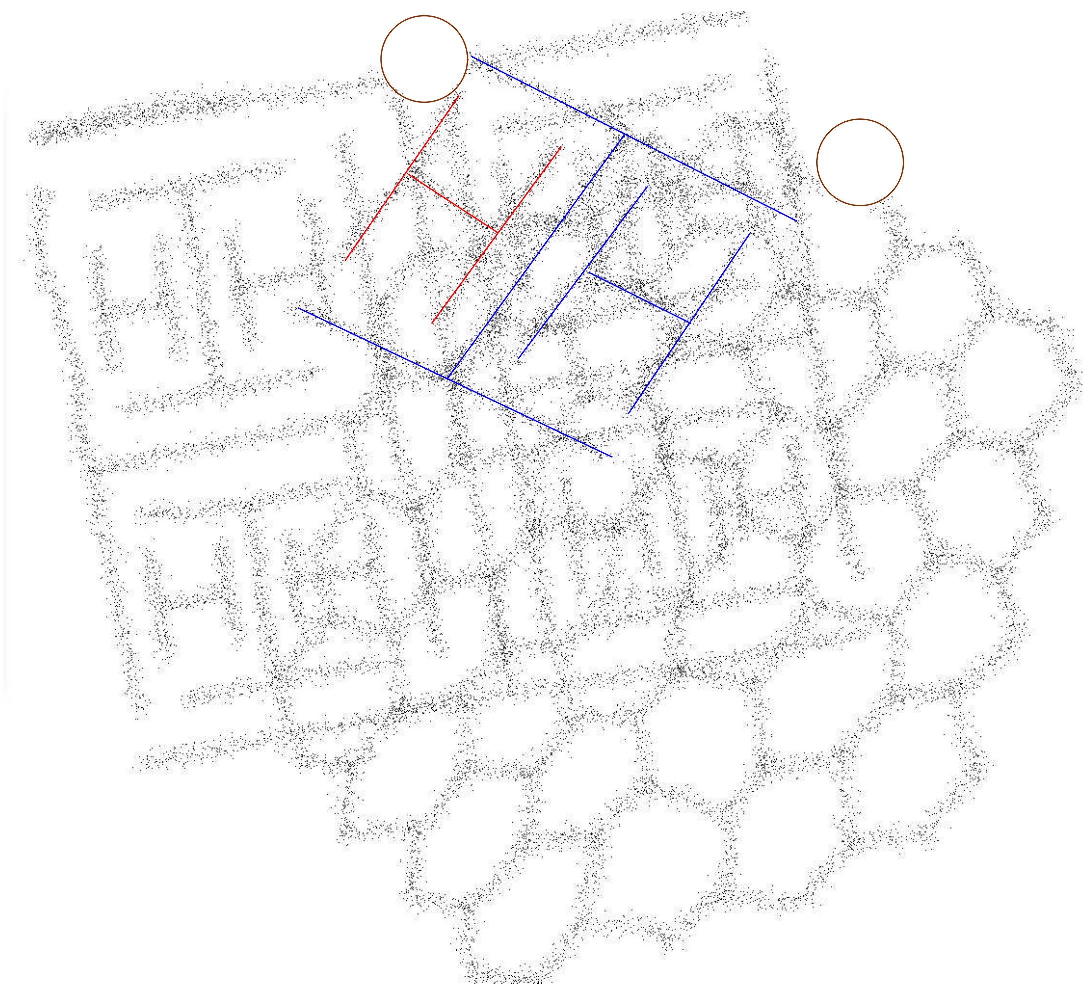


Figure 22. One hex and two Hnests. 573 symbols, overlap index = 0.502.

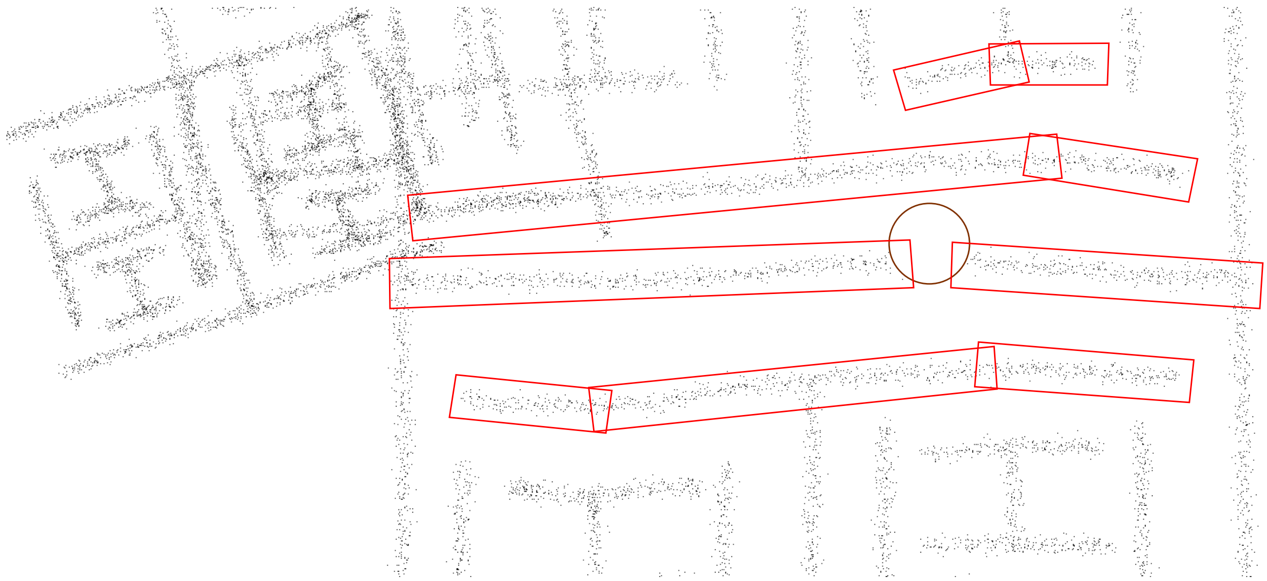


Figure 23. A detail of an image, showing four bent or broken lines.

12. Conclusions

The theoretical contributions of this paper are as follows.

- A new, mathematically well-founded formalism for representing the syntactic structure of graph-like patterns declaratively, fundamentally different from conventional graph grammars based on sets of production rules, in which parsing is seen as the construction of a homomorphism between two networks (a *pattern* and a *grammar*). This offers a new perspective on the problem of representing syntactic structure in non-string patterns, and it helps to overcome the limitations of production rules discussed in §2; it is also amenable to formal mathematical treatment.
- A technique for representing variable affine relationships using *fleximaps* (which could be extended to projective transformations if required); this is a more uniform, general, and mathematically well-founded concept than others in the literature (§5).
- A systematic treatment of affine invariance and symmetry (§§5.3–5.4). (Symmetry is a neglected topic in the literature.)
- A new recognition algorithm for these grammars, which is provably correct subject to certain qualifications (§9.5).

There is no hope of proving favourable time-complexity bounds for the recognition algorithm. As pointed out in §2, the graph parsing problem is NP-complete, even in the absence of errors, except for highly restricted classes of grammars. This certainly applies to my formulation of the problem too. However, the space complexity is linear in the maximum size of the pattern network during recognition (which may be larger than the final pattern).

The recognition algorithm has a number of desirable features:

- all aspects of pattern recognition are integrated in a single process and act synergistically, rather than being applied as a sequence of phases;
- bottom-up and top-down inference are integrated; alternative interpretations are developed in parallel;

- the whole of the image is processed simultaneously; there is no 'traversal' of the image;
- because the syntactic structure of a 'sentence' is represented declaratively by a network, rather than by a sequence of production rule applications, the whole structure is available to work on throughout the recognition process.

As a result the recognition algorithm has a combination of strong recursive representational capabilities and general error-tolerance, not achieved before in the literature. In particular it can

- represent iterative, hierarchical and nested recursive structure, including two dimensions of iteration,
- recognise patterns consisting of up to 1000 symbol tokens arranged in two-dimensional space,
- distinguish overlapping, indistinct or distorted symbol tokens (e.g., bent lines or lines with pieces missing), sometimes successfully disentangling highly cluttered images,
- restore missing lines or junctions in the most grammatically plausible position.

These features go some way towards substantiating my thesis (stated in §1) that symbol processing is not inherently brittle but is made so by the imposition of an unwarranted sequential order on it. When this sequentiality is discarded then robust symbol processing becomes possible.

As further work I intend to develop an algorithm that learns the fleximaps and grammar from example images, without supervision. I believe this can be done by extending the methods of [48].

References

[1] Aalbersberg, I.J., Engelfriet, J. & Rozenberg, G. (1987) Restricting the complexity of regular DNLC languages. In [38], pp. 147–166. https://doi/org/10.1007/3-540-18771-5_51.

[2] Aalbersberg, I.J., Rozenberg, G. & Ehrenfeucht, A. (1986) On the membership problem for regular DNLC grammars. *Discrete Applied Mathematics*, **13**, no. 1, 79–85. https://doi/org/10.1016/0166-218x(86)90070-3.

[3] Adachi, Y. (2004) Visual language design system based on context-sensitive NCE graph grammar. In *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*. Vol. 1. IEEE, pp. 502–507. https://doi/org/10.1109/icsmc.2004.1398348.

[4] Adachi, Y., Kobayashi, S., Tsuchida, K. & Yaku, T. (1999) An NCE context-sensitive graph grammar for visual design languages. In *Proceedings 1999 IEEE Symposium on Visual Languages*. IEEE, pp. 228–235. https://doi/org/10.1109/vl.1999.795908.

[5] Adachi, Y. & Nakajima, Y. (2000) A context-sensitive NCE graph grammar and its parsability. In *Proceedings 2000 IEEE International Symposium on Visual Languages*. IEEE, pp. 111–118. https://doi/org/10.1109/vl.2000.874374.

[6] Bauderon, M. (1996) A category-theoretic approach to vertex replacement: the generation of infinite graphs. In Cuny, J., Ehrig, H., Engels, G & Rozenberg, G. (eds)

Graph Grammars and Their Application to Computer Science, 5th International Workshop, 1995. Lecture Notes in Computer Science, vol. 1073. Springer, pp. 27–37. https://doi/org/10.1007/3-540-61228-9_77.

[7] Bauderon, M., Chen, R. & Ly, O. (2008) Pullback grammars are context-free. In H. Ehrig, R. Heckel, G. Rozenberg & G. Taentzer (eds) *Graph Transformations, ICGT 2008*. Lecture Notes in Computer Science, vol. 5214. Berlin: Springer, pp. 366–378. https://doi/org/10.1007/978-3-540-87405-8_25.

[8] Bauderon, M., Chen, R. & Ly, O. (2009) Context-free categorical grammars. In S. Bozapalidis & G. Rahonis (eds) *Algebraic Informatics, 3rd International Conference on Algebraic Informatics, CAI 2009*. Lecture Notes in Computer Science, vol. 5725. Berlin: Springer, pp. 160–171. https://doi/org/10.1007/978-3-642-03564-7_10.

[9] Bauderon, M. & Jacquet, H. (2001) Node rewriting in graphs and hypergraphs: a categorical framework. *Theoretical Computer Science*, **266**, 463–487. https://doi/org/10.1016/S0304-3975(00)00200-0.

[10] Ben-Arie, J. & Wang, Z. (1998) Pictorial recognition of objects employing affine invariance in the frequency domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, no. 6, 604–618. https://doi/org/10.1109/34.683774.

[11] Beveridge, J.R. & Riseman, E.M. (1997) How easy is matching 2D line models using local search? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, no. 6, 564–579. https://doi/org/10.1109/34.601245.

[12] Björklund, H., Björklund, J. & Ericson, P. (2024) Tree-based generation of restricted graph languages. *International Journal of Foundations of Computer Science*, **351**, nos 1–2, 215–243. https://doi/org/10.1142/S0129054123480106.

[13] Björklund, H., Drewes, F., Ericson, P. & Starke, F. (2021) Uniform parsing for hyperedge replacement grammars. *Journal of Computer and System Sciences*, **118**, 1–27. https://doi/org/10.1016/j.jcss.2020.10.002.

[14] Björklund, J., Drewes, F. & Jonsson, A. (2023) Generation and polynomial parsing of graph languages with non-structural reentrancies. *Computational Linguistics*, **49**, no. 4, 841–882. https://doi/org/10.1162/coli_a_00488.

[15] Blosetin, D. & Schürr, A. (1999) Computing with graphs and graph transformation. *Software – Practice and Experience*, **29**, no. 3, 197–217. https://doi/org/10.1002/(SICI)1097-024X(199903)29:3<197::AID-SPE228>3.0.CO;2-F.

[16] Brandenburg, F.J. (1988) On polynomial time graph grammars. In R. Cori & M. Wirsing (eds) *Proceedings of 5th Conference on Theoretical Aspects of Computer Science*. Lecture Notes in Computer Science, vol. 294. Berlin: Springer, pp. 227–236. https://doi/org/10.1007/BFb0035847.

[17] Brandenburg, F.J. & Skodinis, K. (2005) Finite graph automata for linear and boundary graph languages. *Theoretical Computer Science*, **332**, nos 1–3, 199–232. https://doi/org/10.1016/j.tcs.2004.09.040.

[18] Bunke, H. & Haller, B. (1989) A parser for context free plex grammars. In M. Nagl (ed.) *15th International Workshop on Graph-Theoretic Concepts in Computer*

Science. Lecture Notes in Computer Science, vol. 411. Berlin: Springer, pp. 136–150. https://doi/org/10.1007/3-540-52292-1_10.

[19] Bunke, H. & Haller, B. (1992) Syntactic analysis of context-free plex languages for pattern recognition. In H.S. Baird, H. Bunke & K. Yamamoto (eds) *Structured Document Image Analysis.* Berlin: Springer, pp. 500–519. https://doi/org/10.1007/978-3-642-77281-8_24.

[20] Calvillo, J., Brouwer, H. & Crocker, M.W. (2021) Semantic systematicity in connectionist language production. *Information*, **12**, no. 8. https://doi/org/10.3390/info12080329.

[21] Casey, R.G. & Lecolinet, E. (1996) A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, no. 7, 690–706.

[22] Celik, M. & Yanikoglu, B. (2011) Probabilistic mathematical formula recognition using a 2D context-free graph grammar. In *2011 International Conference on Document Analysis and Recognition.* IEEE Computer Society, pp. 161–166. https://doi/org/10.1109/icdar.2011.41.

[23] Chok, S. & Marriott, K. (1995) Automatic construction of user interfaces from constraint multiset grammars. In *Proceedings of the 11th IEEE Symposium on Visual Languages, VL '95, Darmstadt.* IEEE, pp. 242–249. https://doi/org/10.1109/vl.1995.520815.

[24] Chomsky, N. (1968) *Language and Mind.* New York: Harcourt Brace & World.

[25] Costagliola, G., De Lucia, A., Orefice, S. & Tortora, G. (1998) Positional grammars: a formalism for LR-like parsing of visual languages. In K. Marriott & B. Meyer (eds) *Visual Language Theory.* Berlin: Springer, pp. 171–191.

[26] Courcelle, B. (1987) An axiomatic definition of context-free rewriting and its application to NLC graph grammars. *Theoretical Computer Science*, **55**, nos 2–3, 141–181. https://doi/org/10.1016/0304-3975(87)90102-2.

[27] Courcelle, B. (1991) The monadic second-order logic of graphs V: on closing the gap between definability and recognizability. *Theoretical Computer Science*, **80**, no. 2, 153–202. https://doi/org/10.1016/0304-3975(91)90387-h.

[28] Courcelle, B. (1997) The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg (ed.) *Handbook of Graph Grammars and Computing by Graph Transformation, vol 1, Foundations.* Singapore: World Scientific, pp. 313–400. https://doi/org/10.1142/9789812384720_0005.

[29] Das, A. & Vemuri, R. (2009) Fuzzy logic based guidance to graph grammar framework for automated analog circuit design. In *22nd International Conference on VLSI Design.* IEEE,, pp. 445–450. https://doi/org/10.1109/vlsi.design.2009.79.

[30] Dong, J., Chen, Q., Huang, Z., Yang, J. & Yan, S. (2016) Parsing based on parselets: a unified deformable mixture model for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**, no. 1, 88–101. https://doi/org/10.1109/tpami.2015.2420563.

[31] Drewes, F., Hoffmann, B. & Minas, M. (2019) Extending predictive shift-reduce parsing to contextual hyperedge replacement grammars. In E. Guerra & F. Orejas (eds) *Graph Transformation, ICGT 2019.* Lecture Notes in Computer Science, vol. 11629. Springer, pp. 55–72. https://doi/org/10.1007/978-3-030-23611-3_4.

[32] Drewes, F., Hoffmann, B. & Minas, M. (2021) Rule-based top-down parsing for acyclic contextual hyperedge replacement grammars. In F. Gadducci & T. Kehrer (eds) *Graph Transformation, ICGT 2021.* Lecture Notes in Computer Science, vol. 12741. Springer, pp. 164–184. https://doi/org/10.1007/978-3-030-78946-6_9.

[33] Drewes, F., Kreowski, H.-J. & Habel, A. (1997) Hyperedge replacement graph grammars. In G. Rozenberg (ed.) *Handbook of Graph Grammars and Computing by Graph Transformation, vol 1, Foundations.* Singapore: World Scientific, pp. 95–162. https://doi/org/10.1142/9789812384720_0002.

[34] Drewes, F. & Stade, Y. (2024) On the power of local graph expansion grammars with and without additional restrictions. *Theoretical Computer Science*, **1015**, 114763. https://doi/org/10.1016/j.tcs.2024.114763.

[35] Ehrig, H. (1979) Introduction to the algebraic theory of graph grammars (a survey). In V. Claus, H. Ehrig & G. Rozenberg (eds) *Graph-Grammars and their Application to Computer Science and Biology.* Lecture Notes in Computer Science, vol. 73. Berlin: Springer, pp. 1–69. https://doi/org/10.1007/BFb0025714.

[36] Ehrig, H., Korff, M. & Löwe, M. (1991) Tutorial introduction to the algebraic approach of graph grammars based on double and single pushouts. In [37], pp. 24–37. https://doi/org/10.1007/BFb0017375.

[37] Ehrig, H., Kreowski, H.J. & Rozenberg, G. (eds) (1991) *Graph Grammars and Their Application to Computer Science, 4th International Workshop, Bremen, March 5-9, 1990.* Lecture Notes in Computer Science, vol. 532. Berlin: Springer. https://doi/org/10.1007/BFb0017372.

[38] Ehrig, H., Nagl, M., Rozenberg, G. & Rosenfeld, A. (eds) (1987) *Graph-Grammars and Their Application to Computer Science, 3rd International Workshop, Warrenton, Virginia, USA, December 2-6, 1986.* Lecture Notes in Computer Science, vol. 291. Berlin: Springer. https://doi/org/10.1007/3-540-18771-5.

[39] Engelfriet, J. (1991) A characterization of context-free NCE graph languages by monadic second-order logic on trees. In Ehrig, Kreowski *et al.* (1991), pp. 311–327. https://doi/org/10.1007/BFb0017397.

[40] Engelfriet, J. & Rozenberg, G. (1991) Graph grammars based on node rewriting: an introduction to NLC graph grammars. In [37], pp. 12–23. https://doi/org/10.1007/BFb0017374.

[41] Engelfriet, J. & Rozenberg, G. (1997) Node replacement graph grammars. In G. Rozenberg (ed.) *Handbook of Graph Grammars and Computing by Graph Transformation, vol 1, Foundations.* Singapore: World Scientific, pp. 1–94. https://doi/org/10.1142/9789812384720_0001.

[42] Fahmy, H. & Blostein, D. (1998) A graph-rewriting paradigm for discrete relaxation: application to sheet-music recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, **12**, no. 6, 763–799. https://doi/org/10.1142/S0218001498000439.

[43] Ferrucci, F., Tortora, G., Tucci, M. & Vitiello, G. (1994) A predictive parser for visual languages specified by relation grammars. In *Proceedings of the 10th IEEE Symposium*

on Visual Languages. Los Alamitos, CA: IEEE Computer Society Press, pp. 245–252. https://doi/org/10.1109/vl.1994.363611.

[44] Ferrucci, F., Tortora, G., Tucci, M. & Vitiello, G. (1998) Relation grammars: a formalism for syntactic and semantic analysis of visual languages. In K. Marriott & B. Meyer (eds) *Visual Language Theory*. Berlin: Springer, pp. 219–243.

[45] Fischler, M.A. & Elschlager, R.A. (1973) The representation and matching of pictorial structures. *IEEE Transactions on Computers*, **C-22**, no. 1, 67–92. https://doi/org/10.1109/t-c.1973.223602.

[46] Flasiński, M., Jurek, J. & Peszek, T. (2024) Multi-derivational parsing of vague languages – the new paradigm of syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **46**, no. 8, 5677–5691. https://doi/org/10.1109/TPAMI.2024.3367245.

[47] Flasiński, M. & Myśliński, S. (2010) On the use of graph parsing for recognition of isolated hand postures of Polish Sign Language. *Pattern Recognition*, **43**, 2249–2264. https://doi/org/10.1016/j.patcog.2010.01.004.

[48] Fletcher, P. (2001) Connectionist learning of regular graph grammars. *Connection Science*, **13**, no. 2, 127–188. https://doi/org/10.1080/09540090110072327.

[49] Fletcher, P. (2004) *Mathematical Theory of Recursive Symbol Systems*. Technical report GEN-0401, Keele University, Computer Science Department. https://eprints.keele.ac.uk/id/eprint/63/.

[50] Fletcher, P. (2023) *Mathematical Theory of Networks for Syntactic Pattern Recognition*. Supplement to this paper.

[51] Fodor, J.A. & Pylyshyn, Z.W. (1988) Connectionism and cognitive architecture: a critical analysis. *Cognition*, **28**, nos 1 & 2, 3–71. https://doi/org/10.1016/0010-0277(88)90031-5.

[52] Fürst, L., Mernik, M. & Mahnic, V. (2011) Improving the graph grammar parser of Rekers and Schürr. *IET Software*, **5**, no. 2, 246–261. https://doi/org/10.1049/iet-sen.2010.0081.

[53] Garfield, J.L. (1997) Mentalese not spoken here: computation, cognition and causation. *Philosophical Psychology*, **10**, no. 4, 413–435. https://doi/org/10.1080/09515089708573231.

[54] Gilroy, S., Lopez, A. & Maneth, S. (2017) Parsing graphs with regular graph grammars. In *Proceedings of 6th Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, pp. 199–208. https://doi/org/10.18653/v1/s17-1024.

[55] Golin, E. (1991) Parsing visual languages with picture layout grammars. *Journal of Visual Languages and Computing*, **2**, 371–394. https://doi/org/10.1016/s1045-926x(05)80005-9.

[56] Habel, A. (1992) *Hyperedge Replacement: Grammars and Languages*. Lecture Notes in Computer Science, vol. 643. Berlin: Springer. https://doi/org/10.1007/BFb0013875.

[57] Han, F. & Zhu, S.-C. (2009) Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**, no. 1, 59–73. https://doi/org/10.1109/tpami.2008.65.

[58] Hu, W. & Zhu, S.C. (2015) Learning 3D object templates by quantizing geometry and appearance spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, no. 6, 1190–1205. https://doi/org/10.1109/tpami.2014.2362141.

[59] Julca-Aguilar, F., Mouchère, H., Viard-Gaudin, C. & Hirata, N.S.T. (2020) A general framework for the recognition of online handwritten graphics. *International Journal on Document Analysis and Recognition*, **23**, no. 2, 143–160. https://doi/org/10.1007/s10032-019-00349-6.

[60] Kaul, M. (1983) Parsing of graphs in linear time. In H. Ehrig, M. Nagl & G. Rozenberg (eds) *2nd International Workshop on Graph Grammars and their Application to Computer Science 1982.* Lecture Notes in Computer Science, vol. 153. Berlin: Springer, pp. 206–218. https://doi/org/10.1007/bfb0000108.

[61] Kleyko, D., Davies, M., Frady, E.P., Kanerva, P., Kent, S.J., Olshausen, B.A., Osipov, E., Rabaey, J.M., Rachkovskij, D.A., Rahimi, A. & Sommer, F.T. (2022) Vector symbolic architectures as a computing framework for emerging hardware. *Proceedings of the IEEE*, **110**, no. 10, 1538–1571. https://doi/org/10.1109/JPROC.2022.3209104.

[62] Kong, J, Zhang, K & Zeng, X. (2006) Spatial graph gramamrs for graphical user interfaces. *ACM Transactions on Computer-Human Interaction*, **13**, no. 2, 268–307. https://doi/org/10.1145/1165734.1165739.

[63] Kong, J. & Zhang, K. (2004) Parsing spatial graph grammars. In P. Bottoni, C. Hundhausen, S. Levialdi & G. Tortora (eds) *Proceedings of the 2004 IEEE Symposium on Visual Langages and Human Centric Computing (VLHCC'04).* IEEE Computer Society, pp. 99–101. https://doi/org/10.1109/vlhcc.2004.40.

[64] Kosmala, A., Lavirotte, S., Pottier, L. & Rigoll, G. (1999) On-Line Handwritten Formula Recognition using Hidden Markov Models and Context Dependent Graph Grammars. In *Proceedings of 5th International Conference on Document Analysis and Recognition.* IEEE Computer Society Press, pp. 107–110. https://doi/org/10.1109/icdar.1999.791736.

[65] Lange, K.J. & Welzl, E. (1987) String grammars with disconnecting, or a basic root of the difficulty in graph grammar parsing. *Discrete Applied Mathematics*, **16**, no. 1, 17–30. https://doi/org/10.1016/0166-218x(87)90051-5.

[66] Lavirotte, S. & Pottier, L. (1997) Optical formula recognition. In *Proc. of 4th Int. Conf. on Document Analysis and Recognition.* Vol. 1. IEEE Computer Society, pp. 357–361. https://doi/org/10.1109/ICDAR.1997.619871.

[67] Li, C., Huang, L., Chen, L. & Yu, C. (2013) BGG: A graph grammar approach for software architecture verification and reconfiguration. In *Proceedings, 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.* IEEE, pp. 291–298. https://doi/org/10.1109/imis.2013.56.

[68] Liang, X., Liu, S., Yang, J., Liu, L., Dong, J., Lin, L. & Yan, S. (2015) Deep human parsing with active template regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, no. 12, 2402–2414.

[69] Lichtblau, U. (1991) Recognizing rooted context-free flowgraph languages in polynomial time. In Ehrig, Kreowski & Rozenberg (1991), pp. 538–548. https://doi/org/10.1007/BFb0017411.

[70] Lin, L., Gong, H., Li, L. & Wang, L. (2009) Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters*, **30**, 180–186. https://doi/org/10.1016/j.patrec.2008.02.023.

[71] Lin, L., Wu, T., Porway, J. & Xu, Z. (2009) A stochastic graph grammar for compositional object representation and recognition. *Pattern Recognition*, **42**, no. 7, 1297–1307. https://doi/org/10.1016/j.patcog.2008.10.033.

[72] Liu, Y. & Yang, F. (2021) EGG+: A graph grammar formalism with uncertain structure processing mechanism. *Journal of Logic and Computation*, **31**, no. 7, 1800–1819. https://doi/org/10.1093/logcom/exab055.

[73] Liu, Y. & Yang, F. (2021) An enhancement to graph grammar for the specification of edge semantics. In *2021 IEEE International Conference on Progress in Informatics and Computing.* IEEE, pp. 1–5. https://doi/org/10.1109/PIC53636.2021.9687048.

[74] López, D. & Piñaga, I. (2000) Syntactic pattern recognition by error correcting analysis on tree automata. In F.J. Ferri, J.M. Iñesta, A. Amin & P. Pudil (eds) *Advances in Pattern Recognition, SSPR / SPR 2000.* Lecture Notes in Computer Science, vol. 1876. Berlin: Springer, pp. 133–142. https://doi/org/10.1007/3-540-44522-6_14.

[75] Main, M.G. & Rozenberg, G. (1987) Fundamentals of edge-label controlled graph grammars. In [38], pp. 411–426. https://doi/org/10.1007/3-540-18771-5_67.

[76] Mas, J., Lladós, J., Sanchez, G. & Jorge, J.A.P. (2010) A syntactic approach based on distortion-tolerant Adjacency Grammars and a spatial-directed parser to interpret sketched diagrams. *Pattern Recognition*, **43**, no. 12, 4148–4164. https://doi/org/10.1016/j.patcog.2010.07.003.

[77] Messmer, B.T. & Bunke, H. (1998) A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, no. 5, 493–504.

[78] Modolo, D. & Ferrari, V. (2018) Learning semantic part-based models from Google images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**, no. 6, 1502–1509.

[79] Nagl, M. (1987) Set theoretic approaches to graph grammars. In [38], pp. 41–54. https://doi/org/10.1007/3-540-18771-5_43.

[80] Parizeau, M. & Plamondon, R. (1995) A fuzzy-syntactic approach to allograph modeling for cursive script recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, no. 7, 702–712. https://doi/org/10.1109/34.391412.

[81] Park, S., Nie, B.X. & Zhu, S.C. (2018) Attribute and-or grammar for joint parsing of human pose, parts and attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**, no. 7, 1555–1569. https://doi/org/10.1109/tpami.2017.2731842.

[82] Ranjan, R., Patel, V.M. & Chellappa, R. (2019) HyperFace: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **41**, no. 1, 121–135.

[83] Raphael, C. & Jin, R. (2014) Optical Music Recognition on the International Music Score Library Project. In B. Coüasnon & E.K. Ringger (eds) *Document Recognition and Retrieval XXI*. San Francisco: Soc Imaging Sci & Technol, SPIE. https://doi/org/10.1117/12.2040247.

[84] Rekers, J. & Schürr, A. (1997) Defining and parsing visual languages with layered graph grammars. *Journal of Visual Languages and Computing*, **8**, no. 1, 27–55. https://doi/org/10.1006/jvlc.1996.0027.

[85] Rocha, J. & Pavlidis, T. (1995) Character recognition without segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, no. 9, 903–909.

[86] Rozenberg, G. & Welzl, E. (1986) Boundary NLC graph grammars – basic definitions, normal forms, and complexity. *Information and Control*, **69**, nos 1–3, 136–167. https://doi/org/10.1016/s0019-9958(86)80045-6.

[87] Sabinasz, D & Schoner, G. (2022) A neural dynamic model perceptually grounds nested noun phrases. *Proceedings of the 44th Annual Meeting of the Cognitive Science Society*, **44**, 847–853. https://doi/org/10.1111/tops.12630.

[88] Sánchez, G., Lladós, J. & Tombre, K. (2002) An error-correction graph grammar to recognize texture symbols. In D. Blostein & Y.B. Kwon (eds) *Graphics Recognition Algorithms and Applications, GREC 2001*. Lecture Notes in Computer Science, vol. 2390. Berlin: Springer, pp. 28–138. https://doi/org/10.1007/3-540-45868-9_10.

[89] Schuster, R. (1987) *Graphgrammatiken und Grapheinbettungen: Algorithmen und Komplexität*. PhD thesis, Universität Passau.

[90] Shi, Z., Zeng, X., Huang, S., Qi, Z., Li, H., Hu, B., Zhang, S., Liu, Y. & Wang, C. (2015) A method to simplify description and implementation of graph grammars. In *6th International Conference on Computing, Communication and Networking Technologies, Denton, USA*. IEEE, pp. 1–6. https://doi/org/10.1109/icccnt.2015.7395225.

[91] Shi, Z., Zeng, X., Zou, Y., Huang, S., Li, H., Hu, B. & Yao, Y. (2018) A temporal graph grammar formalism. *Journal of Visual Languages and Computing*, **47**, 62–76. https://doi/org/10.1016/j.jvlc.2018.06.003.

[92] Si, Z. & Zhu S.-C. (2013) Learning AND-OR templates for object recognition and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, no. 9, 2189–2205. https://doi/org/10.1109/tpami.2013.35.

[93] Simon, L., Teboul, O., Koutsourakis, P. & Paragios, N. (2011) Random exploration of the procedural space for single-view 3D modeling of buildings. *International Journal of Computer Vision*, **93**, no. 2, 253–271.

[94] Siskind, J.M., Sherman, J.J., Pollak, I., Harper, M.P. & Bouman, C.A. (2007) Spatial random tree grammars for modeling hierarchical structure in images with regions of arbitrary shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, no. 9, 1504–1519. https://doi/org/10.1109/TPAMI.2007.1169.

[95] Skomorowski, M. (2007) Syntactic recognition of distorted patterns by means of random graph parsing. *Pattern Recognition Letters*, **28**, 572–581. https://doi/org/10.1016/j.patrec.2006.10.003.

[96] Smolensky, P. (1988) On the proper treatment of connectionism. *Behavioral and Brain Sciences*, **11**, no. 1, 1–23. https://doi/org/10.1017/S0140525X00052432.

[97] Tombre, K. (1996) Structural and syntactic methods in line drawing analysis: to which extent do they work? In P. Perner, P. Wang & A. Rozenfeld (eds) *Advances in Syntactic and Structural Pattern Recognition, SSPR '96.* Lecture Notes in Computer Science, vol. 1121. Berlin: Springer, pp. 310–321.

[98] Tu, Z., Chen, X., Yuille, A.L. & Zhu, S.-C. (2005) Image parsing: unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, **63**, no. 2, 113–140. https://doi/org/10.1007/s11263-005-6642-x.

[99] Valveny, E. & Marti, E. (2003) A model for image generation and symbol recognition through the deformation of lineal shapes. *Pattern Recognition Letters*, **24**, no. 15, 2857–2867. https://doi/org/10.1016/s0167-8655(03)00144-2.

[100] Wang, L., Wang, Y. & Gao, W. (2011) Mining layered grammar rules for action recognition. *International Journal of Computer Vision*, **93**, no. 2, 162–182. https://doi/org/10.1007/s11263-010-0393-z.

[101] Wittenburg, K.B. (1992) Earley-style parsing for relational grammars. In *Proceedings of the IEEE Workshop on Visual Languages.* Los Alamitos, CA: IEEE Computer Society Press, pp. 192–199. https://doi/org/10.1109/wvl.1992.275765.

[102] Wittenburg, K.B. & Weitzman, L.M. (1998) Relational grammars: theory and practice in a visual language interface for process modeling. In K. Marriott & B. Meyer (eds) *Visual Language Theory.* Berlin: Springer, pp. 193–217.

[103] Ye, Y.J. & Sun, W.W. (2020) Exact yet efficient graph parsing, bi-directional locality and the constructivist hypothesis. In D. Jurafsky, J. Chai, N. Schluter & J. Tetreault (eds) *58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, pp. 4100–4110. https://doi/org/10.18653/v1/2020.acl-main.377.

[104] Yuan, Y., Liu, X., Dikubab, W., Liu, H., Ji, Z., Wu, Z. & Bai, X. (2022) Syntax-Aware Network for Handwritten Mathematical Expression Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, pp. 4543–4552. https://doi/org/10.1109/CVPR52688.2022.00451.

[105] Yuille, A. L., Hallinan, P. W. & Cohen, D. S. (1992) Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, **8**, no. 2, 99–111. https://doi/org/10.1007/bf00127169.

[106] Zhang, D.Q., Zhang, K. & Cao, J. (2001) A context-sensitive graph grammar formalism for the specification of visual languages. *The Computer Journal*, **44**, no. 3, 187–200. https://doi/org/10.1093/comjnl/44.3.186.

[107] Zhu, L., Chen, Y., Li, C. & Yuille, A. (2011) Max margin learning of hierarchical configural deformable templates (HCDTs) for efficient object parsing and pose estimation. *International Journal of Computer Vision*, **93**, 1–21. https://doi/org/10.1007/s11263-010-0375-1.

[108] Zhu, L., Chen, Y. & Yuille, A. (2009) Unsupervised learning of probabilistic grammar-Markov models for object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**, no. 1, 114–128. https://doi/org/10.1109/tpami.2008.67.

[109] Zou, Y. (2024) A comparison of reserved graph grammar and edge-based graph grammar. *Computer Journal*, **68**, no. 2, 145–162. https://doi/org/10.1093/comjnl/bxae100.

[110] Zou, Y., Lü, J. & Tao, X. (2019) Research on context of implicit context-sensitive graph grammars. *Journal of Computer Languages*, **51**, 241–260. https://doi/org/10.1016/j.cola.2019.01.002.

[111] Zou, Y., Zeng, X. & Zhu, Y. (2022) A general parsing algorithm with context matching for context-sensitive graph grammars. *Multimedia Tools and Applications*, **81**, 273–297. https://doi/org/10.1007/s11042-021-11076-8.

SUPPLEMENT

Mathematical Theory of Networks for Syntactic Pattern Recognition

Peter Fletcher
School of Computer Science and Mathematics,
Keele University,
Keele,
Staffordshire, ST5 2BG, U.K.
Email: p.fletcher@keele.ac.uk

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

There are no conflicts of interest.

Abstract

This is a mathematical supplement to the paper 'A New Graph Grammar Formalism for Robust Syntactic Pattern Recognition', available at https://doi.org/10.48550/arXiv.2504.15975. It provides the full mathematical theory and complete proofs.

Table of Contents

1. Introduction 4

2. Functions and Relations 6
   - 2.1 Introduction 6
   - 2.2 Basic definitions and theorems 6
   - 2.3 Acyclic and connected relations 11
   - 2.4 Minimal relations relative to a function 18

3. Networks and Homomorphisms 20
   - 3.1 Introduction 20
   - 3.2 Definition of network and homomorphism 20
   - 3.3 Minimal relations relative to a network 23
   - 3.4 Semi-definite and definite networks 25
   - 3.5 Properties preserved by homomorphisms 28
   - 3.6 Theorems on definiteness 31
   - 3.7 Subnetworks 40
   - 3.8 The recognition problem and process (initial statement) 42

4. Embeddings and the Definite Match Function 43
   - 4.1 Introduction 43
   - 4.2 Embedding tokens and types 43
   - 4.3 Symmetries 46
   - 4.4 Templates and saturation 49
   - 4.5 The definite match function 50
   - 4.6 Invariance theorems 51
   - 4.7 The recognition problem – definitive statement 56

5. Inclusion Functions 57
   - 5.1 Introduction 57
   - 5.2 Definition of inclusion functions 57
   - 5.3 The recognition process 59

6. The Indefinite Match Function 60
   - 6.1 Introduction 60
   - 6.2 Template matching 60
   - 6.3 The indefinite match function 61
   - 6.4 Invariance theorems 62
   - 6.5 Adjustment of the embedding token 63
   - 6.6 The recognition process 63

7. How the Inclusion Functions are Determined 64
   - 7.1 Introduction 64
   - 7.2 The inclusion vector, $\boldsymbol{i}$ 64
   - 7.3 How $\boldsymbol{i}$ is determined 66
   - 7.4 Algorithm for finding $\boldsymbol{i}$ 66
   - 7.5 The recognition process 69

8. The Structural Operations 70
8.1 Introduction 70
8.2 Pruning operations 70
8.3 Unprunability and its consequences 76
8.4 Extension operations 81
8.5 Inextendability and its consequences 85
8.6 Merging two symbol tokens 86
8.7 Partitioning a symbol token into two 87
8.8 The outcome of the recognition process 89

9. The Whole Recognition Process 90
9.1 Introduction 90
9.2 Line operations 90
9.3 How temperature is determined 91
9.4 How the bareness function B is determined 92
9.5 Summary of the entire recognition process 93
9.6 The outcome of recognition 93

Summary of Notation and Terminology 96

1. Introduction

The purpose of this project is to develop a new graph grammar formalism for error-tolerant syntactic pattern recognition. This paper provides the mathematical theory in full. It should be read in conjunction with the paper *A New Graph Grammar Formalism for Robust Syntactic Pattern Recognition*, available at https://doi.org/10.48550/arXiv.2504.15975, which provides a broader account of the context, purpose and application of this project, with example grammars and runs.

This paper develops further the mathematical framework in my 2004 technical report, *Mathematical Theory of Recursive Symbol Systems* (Technical Report GEN-0401, Keele University, Computer Science Department, available at https://eprints.keele.ac.uk/id/eprint/63/), which I shall cite here simply as '(2004)'. Sections 1–5 of (2004) are still valid, but section 6 (Networks and Homomorphisms) and section 7 (The Recognition Problem) are inadequate and this report supersedes them. In particular, I have extended the concept of network, mainly by the addition of *subsymbols*, and developed the mathematical theory to define the pattern recognition process fully.

The contents of this report are as follows.

Section 2 develops the general theory of functions and relations that will be needed in the subsequent sections.

Section 3 gives new definitions of *network*, *homomorphism* and *definite* and *semi-definite* networks, superseding those in (2004, §6).

Section 4 extends the theory of embeddings, introduced in (2004, §6.4). The *Match* function, which was introduced in (2004, §7.3), is treated more satisfactorily here as two functions, a *definite match function DM* and an *indefinite match function IM* (defined in §6). The subject of symmetries is treated more thoroughly and several theorems are proved on the affine invariance of the grammatical framework. The theory of templates (introduced in (2004, §4)) is extended here by introducing the concept of *saturation*, which expresses the notion that when several templates overlap in the image plane they interfere with one another. The recognition problem can now be formally stated.

Section 5 develops the theory of inclusion functions, which were introduced in (2004, §7.2).

Section 6 defines the *indefinite match function IM*, which applies to networks that are not necessarily definite, with inclusion functions. The invariance theorems of §4 are generalised to *IM*. The calculation of the adjustment of the embeddings by gradient ascent in (2004) is adapted to *IM*.

Section 7 states the mechanism by which the inclusion functions are determined during recognition.

Section 8 defines the *structural operations* by which the pattern grows during recognition.

Section 9 states the entire recognition algorithm.

The whole theory is built up by a process of stepwise refinement. The first four sections establish the underlying framework and specify the recognition problem. Sections 5–9 develop the algorithm for solving it incrementally, starting with an abstract account of the recognition process and refining it step by step with each chapter. Stepwise refinement is a valuable method for managing the complexity of the mathematical theory and the algorithm: it allows for a separation of concerns, particularly between the structural and geometric aspects of the theory, and enables the theory to be built up one layer at a time, with the whole recognition process in view at each stage, at ever-increasing levels of detail.

At each stage of the theory I shall make explicit the correctness conditions the recognition algorithm must satisfy to solve the recognition problem, culminating in theorem 48. The earlier, more fundamental stages (§§2–6) are clean, rigorous and provably correct; the subsequent stages are progressively messier, with arbitrary parameters and algorithms not guaranteed to converge to the optimum solution. I distinguish clearly between those properties of the recognition process that are guaranteed to hold, those that are likely to hold, and those that one merely hopes will hold.

2. Functions and Relations

2.1 Introduction

This section introduces some concepts involving functions and relations that will be needed in the theory of networks to follow, especially the concept of an *acyclic* relation, a *connected* relation relative to a function, and a *minimal* relation relative to a function. Relations will be used in the following section to express the way in which subsymbols are glued to supersymbols.

2.2 Basic definitions and theorems

A *function* $f{:}X \to Y$ maps every element of the set X to a unique element of the set Y. X is called the *domain* of f, dom(f).

Definition. For any function $f{:}X \to Y$ and any sets A, B,

$$f(A) = \{\, f(x) \mid x \in X \cap A \,\}, \qquad f^{-1}(B) = \{\, x \in X \mid f(x) \in B \,\}.$$

The *range* of f, ran(f), is defined as $f(X)$.

Definition. For any function $f{:}X \to Y$ and any set $A \subseteq X$, the *restriction* of f to A, $f|_A{:}A \to Y$, is defined by

$$\forall x{\in}A\ f|_A(x) = f(x).$$

Definition. For any functions $f{:}A \to B$ and $g{:}C \to D$, the *composition* of f and g, $f \circ g{:}C \cap g^{-1}(A) \to B$, is defined by

$$\forall x{\in}C \cap g^{-1}(A) \quad (f \circ g)(x) = f(g(x)).$$

The term *relation* will be understood to mean a binary relation applicable to all objects: for any relation R and every pair of objects a, b, either $R(a,b)$ holds or it does not.

Definition. The *domain* dom(R) and *range* ran(R) of a relation R are defined by

$$\mathrm{dom}(R) = \{\, a \mid \exists b\ R(b,a) \,\}, \qquad \mathrm{ran}(R) = \{\, b \mid \exists a\ R(b,a) \,\}.$$

Definition. A relation R is said to be *on* a set A iff $\mathrm{dom}(R) \subseteq A$ and $\mathrm{ran}(R) \subseteq A$.

Definition. For any set A, the *identity* relation id_A is defined by

$$\forall a, b \quad (id_A(a,b) \iff a = b \in A).$$

Definition. The empty relation $\perp$ is defined by $\forall a, b\ \neg\perp(a,b)$.

Definition. For any relations R and S,

$$\begin{aligned}
R = S &\iff \forall a, b\ (R(a,b) \iff S(a,b)) \\
R \subseteq S &\iff \forall a, b\ (R(a,b) \Rightarrow S(a,b)) \\
R \subset S &\iff R \subseteq S \wedge R \neq S.
\end{aligned}$$

Definition. For any relations R and S, the relations $R \cap S$, $R \cup S$ and $R \backslash S$ are defined by

$$\forall a,b \quad \begin{cases} (R \cap S)(a,b) \Leftrightarrow R(a,b) \wedge S(a,b), \\ (R \cup S)(a,b) \Leftrightarrow R(a,b) \vee S(a,b), \\ (R \backslash S)(a,b) \Leftrightarrow R(a,b) \wedge \neg S(a,b). \end{cases}$$

Definition. For any relations R and S, the *composed* relation $R \circ S$ is defined by

$$\forall a,c \quad ((R \circ S)(a,c) \;\Leftrightarrow\; \exists b\ (R(a,b) \wedge S(b,c))).$$

(In all expressions, the composition operator $\circ$ has higher syntactic precedence than $\cap$, $\cup$ and $\backslash$.)

Definition. For any relation R, the *inverse* relation R^{-1} is defined by

$$\forall a,b \quad (R^{-1}(b,a) \;\Leftrightarrow\; R(a,b)).$$

Definition. The *graph* of a function f is the relation $\overline{f}$ defined by

$$\forall a,b \quad (\overline{f}(b,a) \;\Leftrightarrow\; a \in \mathrm{dom}(f) \wedge b = f(a)).$$

Theorem 1. (Obvious properties of relations.) For any relations R,S,T, any functions f,g, and any sets A,X,Y,

(i) $\overline{f \circ g} = \overline{f} \circ \overline{g}$, $\mathrm{dom}(\overline{f}) = \mathrm{dom}(f)$, and $\mathrm{ran}(\overline{f}) = \mathrm{ran}(f)$

(ii) if $f{:}X \to Y$ then $\overline{f} \circ id_X = \overline{f} = id_Y \circ \overline{f}$,

(iii) if $f,g{:}X \to Y$ and $\overline{f} \subseteq \overline{g}$ then $f = g$,

(iv) $\overline{f|_A} = \overline{f} \circ id_A$,

(v) $id_A \circ \overline{f} = \overline{f} \circ id_{f^{-1}(A)}$,

(vi) $id_A \circ R \subseteq R$ and $R \circ id_A \subseteq R$, with equality if R is on A,

(vii) $id_X \circ id_Y = id_{X \cap Y} = id_X \cap id_Y$ and $id_{X \cup Y} = id_X \cup id_Y$,

(viii) $\perp \circ R = \perp = R \circ \perp$,

(ix) $T \subseteq R \cap S$ iff $T \subseteq R$ and $T \subseteq S$ (in particular, $R \cap S \subseteq R$ and $R \cap S \subseteq S$),

(x) $R \cup S \subseteq T$ iff $R \subseteq T$ and $S \subseteq T$ (in particular, $R \subseteq R \cup S$ and $S \subseteq R \cup S$),

(xi) $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$, $(R \cap S)^{-1} = R^{-1} \cap S^{-1}$, $(R \cup S)^{-1} = R^{-1} \cup S^{-1}$,

(xii) if $R \subseteq S$ then $R^{-1} \subseteq S^{-1}$, $R \circ T \subseteq S \circ T$ and $T \circ R \subseteq T \circ S$,

(xiii) if $R \subseteq S$ then $\mathrm{dom}(R) \subseteq \mathrm{dom}(S)$ and $\mathrm{ran}(R) \subseteq \mathrm{ran}(S)$,

(xiv) if $R \circ S = \perp$ then $\mathrm{dom}(R) \cap \mathrm{ran}(S) = \emptyset$,

(xv) $\mathrm{ran}(R) = \mathrm{dom}(R^{-1})$,

(xvi) $id_{\mathrm{dom}(R)} \subseteq R^{-1} \circ R$ and $id_{\mathrm{ran}(R)} \subseteq R \circ R^{-1}$,

(xvii) $\mathrm{dom}(R \circ id_A) = \mathrm{dom}(R) \cap A$ and $\mathrm{ran}(id_A \circ R) = A \cap \mathrm{ran}(R)$.

Theorem 2. If $f{:}X \to Y$, and R and S are relations, then

(i) $\overline{f} \circ \overline{f}^{-1} \subseteq id_Y$,

(ii) $id_X \subseteq \overline{f}^{-1} \circ \overline{f}$,

(iii) if $R \subseteq S \circ \overline{f}$ then $R \circ \overline{f}^{-1} \subseteq S \circ id_Y$,

(iv) if $\overline{f} \circ R \subseteq S$ then $id_X \circ R \subseteq \overline{f}^{-1} \circ S$.

Proof. (i) For any y, y' such that $(\overline{f} \circ \overline{f}^{-1})(y, y')$, we have $y, y' \in Y$ and $\exists x \in X\ (y = f(x) \wedge f(x) = y')$, which implies that $y = y'$, and hence $id_Y(y, y')$.

(ii) For any $x \in X$, there exists y (namely $f(x)$) such that $f(x) = y$ and $y = f(x)$, and hence $(\overline{f}^{-1} \circ \overline{f})(x, x)$.

(iii) If $R \subseteq S \circ \overline{f}$ then $R \circ \overline{f}^{-1} \subseteq S \circ \overline{f} \circ \overline{f}^{-1} \subseteq S \circ id_Y$, by part (i).

(iv) If $\overline{f} \circ R \subseteq S$ then $id_X \circ R \subseteq \overline{f}^{-1} \circ \overline{f} \circ R \subseteq \overline{f}^{-1} \circ S$, by part (ii). ▮

Theorem 3. For any relations R, S, T and any function f,
(i) $(R \cup S) \circ T = R \circ T \ \cup\ S \circ T$,
(ii) $T \circ (R \cup S) = T \circ R \ \cup\ T \circ S$,
(iii) $(R \cap S) \circ T \subseteq R \circ T \ \cap\ S \circ T$,
(iv) $T \circ (R \cap S) \subseteq T \circ R \ \cap\ T \circ S$,
(v) $(R \cap S) \circ \overline{f} = R \circ \overline{f} \ \cap\ S \circ \overline{f}$,
(vi) $\overline{f}^{-1} \circ (R \cap S) = \overline{f}^{-1} \circ R \ \cap\ \overline{f}^{-1} \circ S$.

Proof. (i) For any a, c,

$$\begin{aligned}
((R \cup S) \circ T)(a, c) &\text{ iff } \exists b\ ((R \cup S)(a, b) \wedge T(b, c)) \\
&\text{ iff } \exists b\ ((R(a, b) \vee S(a, b)) \wedge T(b, c)) \\
&\text{ iff } \exists b\ (R(a, b) \wedge T(b, c)) \vee \exists b\ (S(a, b) \wedge T(b, c)) \\
&\text{ iff } (R \circ T \ \cup\ S \circ T)(a, c).
\end{aligned}$$

(ii) For any a, c,

$$\begin{aligned}
(T \circ (R \cup S))(a, c) &\text{ iff } \exists b\ (T(a, b) \wedge (R \cup S)(b, c)) \\
&\text{ iff } \exists b\ (T(a, b) \wedge (R(b, c) \vee S(b, c))) \\
&\text{ iff } \exists b\ (T(a, b) \wedge R(b, c)) \vee \exists b\ (T(a, b) \wedge S(b, c)) \\
&\text{ iff } (T \circ R \ \cup\ T \circ S)(a, c).
\end{aligned}$$

(iii) From $R \cap S \subseteq R$ and $R \cap S \subseteq S$ we infer $(R \cap S) \circ T \subseteq R \circ T$ and $(R \cap S) \circ T \subseteq S \circ T$, and hence $(R \cap S) \circ T \subseteq R \circ T \ \cap\ S \circ T$.

(iv) From $R \cap S \subseteq R$ and $R \cap S \subseteq S$ we infer $T \circ (R \cap S) \subseteq T \circ R$ and $T \circ (R \cap S) \subseteq T \circ S$, and hence $T \circ (R \cap S) \subseteq T \circ R \ \cap\ T \circ S$.

(v) For any a, c,

$$\begin{aligned}
((R \cap S) \circ \overline{f})(a, c) &\text{ iff } \exists b\ ((R \cap S)(a, b) \wedge c \in \mathrm{dom}(f) \wedge b = f(c)) \\
&\text{ iff } c \in \mathrm{dom}(f) \wedge (R \cap S)(a, f(c)) \\
&\text{ iff } c \in \mathrm{dom}(f) \wedge R(a, f(c)) \wedge S(a, f(c)) \\
&\text{ iff } \exists b\ (R(a, b) \wedge c \in \mathrm{dom}(f) \wedge b = f(c)) \ \wedge\ \exists b'\ (S(a, b') \wedge c \in \mathrm{dom}(f) \wedge b' = f(c)) \\
&\text{ iff } (R \circ \overline{f})(a, c) \ \wedge\ (S \circ \overline{f})(a, c) \\
&\text{ iff } (R \circ \overline{f} \ \cap\ S \circ \overline{f})(a, c).
\end{aligned}$$

(vi) Part (v) gives $(R^{-1} \cap S^{-1}) \circ \overline{f} = R^{-1} \circ \overline{f} \ \cap\ S^{-1} \circ \overline{f}$, and taking the inverse gives $\overline{f}^{-1} \circ (R \cap S) = \overline{f}^{-1} \circ R \ \cap\ \overline{f}^{-1} \circ S$. ▮

Theorem 4. For any relations R, S, T, and any sets A, B,
(i) $R \circ S \ \cap\ T \subseteq (R \ \cap\ T \circ S^{-1}) \circ S$,

(ii) $R \circ S \ \cap \ T \subseteq R \circ (S \ \cap \ R^{-1} \circ T)$,
(iii) $R \circ id_A \ \cap \ S = (R \cap S) \circ id_A$,
(iv) $id_A \circ R \ \cap \ S = id_A \circ (R \cap S)$,
(v) if $id_A \circ R \subseteq S \circ id_B$ then $id_A \circ R \subseteq R \circ id_B$,
(vi) if $R \circ id_A \subseteq id_B \circ S$ then $R \circ id_A \subseteq id_B \circ R$.

Proof. (i) For any x, z,

$$\begin{aligned}(R \circ S \ \cap \ T)(x,z) &\text{ iff } \exists y \ (R(x,y) \wedge S(y,z) \wedge T(x,z)) \\ &\text{ implies } \exists y, z' \ (R(x,y) \wedge T(x,z') \wedge S(y,z') \wedge S(y,z)) \quad \text{by taking } z' = z \\ &\text{ iff } \exists y \ (R(x,y) \wedge (T \circ S^{-1})(x,y) \wedge S(y,z)) \\ &\text{ iff } ((R \ \cap \ T \circ S^{-1}) \circ S)(x,z).\end{aligned}$$

(ii) Applying part (i), with S^{-1} substituted for R, R^{-1} substituted for S, and T^{-1} substituted for T,

$$S^{-1} \circ R^{-1} \ \cap \ T^{-1} \subseteq (S^{-1} \ \cap \ T^{-1} \circ R) \circ R^{-1}$$

and then inverting this gives

$$R \circ S \ \cap \ T \subseteq R \circ (S \ \cap \ R^{-1} \circ T).$$

(iii) Using part (i), $R \circ id_A \ \cap \ S \subseteq (R \cap S \circ id_A) \circ id_A \subseteq (R \cap S) \circ id_A$. Conversely, using theorem 3(iii), $(R \cap S) \circ id_A \subseteq R \circ id_A \ \cap \ S \circ id_A \subseteq R \circ id_A \ \cap \ S$.

(iv) Applying part (iii) with R^{-1} substituted for R and S^{-1} substituted for S gives $R^{-1} \circ id_A \ \cap \ S^{-1} = (R^{-1} \cap S^{-1}) \circ id_A$, and inverting this gives $id_A \circ R \ \cap \ S = id_A \circ (R \cap S)$.

(v) If $id_A \circ R \subseteq S \circ id_B$ then, combining this with $id_A \circ R \subseteq R$ gives

$$\begin{aligned}id_A \circ R &\subseteq S \circ id_B \ \cap \ R \\ &= (S \cap R) \circ id_B \qquad \text{by part (iii)} \\ &\subseteq R \circ id_B \qquad \text{since } S \cap R \subseteq R.\end{aligned}$$

(vi) If $R \circ id_A \subseteq id_B \circ S$ then, combining this with $R \circ id_A \subseteq R$ gives

$$\begin{aligned}R \circ id_A &\subseteq id_B \circ S \ \cap \ R \\ &= id_B \circ (S \cap R) \qquad \text{by part (iv)} \\ &\subseteq id_B \circ R \qquad \text{since } S \cap R \subseteq R.\end{aligned}$$

∎

Theorem 5. For any relation R, $R = \overline{f}$ for some function $f{:}X \to Y$ iff $R \circ R^{-1} \subseteq id_Y$ and $id_X \subseteq R^{-1} \circ R$.

Proof. ($\Rightarrow$) If $R = \overline{f}$ then $R \circ R^{-1} \subseteq id_Y$ and $id_X \subseteq R^{-1} \circ R$ by theorem 2(i),(ii).

($\Leftarrow$) Suppose $R \circ R^{-1} \subseteq id_Y$ and $id_X \subseteq R^{-1} \circ R$. For any $x \in X$, $id_X(x,x)$ holds, so $(R^{-1} \circ R)(x,x)$ holds, so there exists y such that $R(y,x)$.

For any $x \in X$ and any y, if $R(y,x)$ then $(R \circ R^{-1})(y,y)$, so $id_Y(y,y)$, so $y \in Y$.

For any $x \in X$ and $y, y' \in Y$, if $R(y,x)$ and $R(y',x)$ then $(R \circ R^{-1})(y,y')$, so $id_Y(y,y')$, so $y = y'$.

These facts show that $R = \overline{f}$ for a function $f{:}X \to Y$. ∎

Theorem 6. A rectangle of functions $\begin{tikzcd} D & C \arrow[l, "g"'] \\ B \arrow[u, "f"] & A \arrow[l, "p"] \arrow[u, "q"'] \end{tikzcd}$ is a pullback in the category of sets iff

$$\overline{f}^{-1} \circ \overline{g} = \overline{p} \circ \overline{q}^{-1} \quad \text{and} \quad \overline{p}^{-1} \circ \overline{p} \ \cap\ \overline{q}^{-1} \circ \overline{q} = id_A.$$

Proof. I shall use the standard characterisation of a pullback in the category of sets, namely $f \circ p = g \circ q$ and $\forall b{\in}B\ \forall c{\in}C\ (f(b) = g(c) \Rightarrow \exists! a{\in}A\ (p(a) = b \wedge q(a) = c))$.

($\Rightarrow$) Suppose the rectangle is a pullback. I shall verify the two equations.

$\underline{\overline{f}^{-1} \circ \overline{g} \subseteq \overline{p} \circ \overline{q}^{-1}}$. For any b and c, if $(\overline{f}^{-1} \circ \overline{g})(b,c)$ then $b{\in}B$, $c{\in}C$ and $\exists d\ (f(b) = d \wedge d = g(c))$, so $f(b) = g(c)$, so $\exists a{\in}A\ (b = p(a) \wedge q(a) = c)$ by the pullback condition, so $(\overline{p} \circ \overline{q}^{-1})(b,c)$.

$\underline{\overline{p} \circ \overline{q}^{-1} \subseteq \overline{f}^{-1} \circ \overline{g}}$. For any b and c, if $(\overline{p} \circ \overline{q}^{-1})(b,c)$ then there exists $a{\in}A$ such that $b = p(a) \wedge q(a) = c$, so $f(b) = f(p(a)) = g(q(a)) = g(c)$, so $(\overline{f}^{-1} \circ \overline{g})(b,c)$.

$\underline{id_A \subseteq \overline{p}^{-1} \circ \overline{p} \ \cap\ \overline{q}^{-1} \circ \overline{q}}$. This follows from $id_A \subseteq \overline{p}^{-1} \circ \overline{p}$ and $id_A \subseteq \overline{q}^{-1} \circ \overline{q}$, which come from theorem 2(ii).

$\underline{\overline{p}^{-1} \circ \overline{p} \ \cap\ \overline{q}^{-1} \circ \overline{q} \subseteq id_A}$. For any a, a', if $(\overline{p}^{-1} \circ \overline{p} \ \cap\ \overline{q}^{-1} \circ \overline{q})(a, a')$ then $a, a' \in A$ and $\exists b{\in}B\ (p(a) = b \wedge b = p(a'))$ and $\exists c{\in}C\ (q(a) = c \wedge c = q(a'))$, hence $f(b) = f(p(a)) = g(q(a)) = g(c)$, so by the pullback condition there exists only one $x{\in}A$ such that $p(x) = b$ and $q(x) = c$, so $a = a'$, as required.

($\Leftarrow$) Suppose $\overline{f}^{-1} \circ \overline{g} = \overline{p} \circ \overline{q}^{-1}$ and $\overline{p}^{-1} \circ \overline{p} \ \cap\ \overline{q}^{-1} \circ \overline{q} = id_A$. I shall verify the conditions for a pullback.

$\underline{f \circ p = g \circ q}$. For any $a{\in}A$, $(\overline{p} \circ \overline{q}^{-1})(p(a), q(a))$ holds, so by the equation $\overline{f}^{-1} \circ \overline{g} = \overline{p} \circ \overline{q}^{-1}$ we have $(\overline{f}^{-1} \circ \overline{g})(p(a), q(a))$, which means that $f(p(a)) = g(q(a))$, as required.

$\underline{\forall b{\in}B\ \forall c{\in}C\ (f(b) = g(c) \Rightarrow \exists a{\in}A\ (p(a) = b \wedge q(a) = c))}$. Consider any $b{\in}B$ and $c{\in}C$ such that $f(b) = g(c)$. Then $(\overline{f}^{-1} \circ \overline{g})(b,c)$ holds, so by the equation $\overline{f}^{-1} \circ \overline{g} = \overline{p} \circ \overline{q}^{-1}$ we have $(\overline{p} \circ \overline{q}^{-1})(b,c)$, which means that there exists $a{\in}A$ such that $b = p(a)$ and $q(a) = c$ as required.

$\underline{\forall b{\in}B\ \forall c{\in}C\ (f(b) = g(c) \Rightarrow \exists^{\leq 1} a{\in}A\ (p(a) = b \wedge q(a) = c))}$. Consider any $b{\in}B$ and $c{\in}C$ such that $f(b) = g(c)$. Suppose there exist $a, a'{\in}A$ such that $p(a) = b = p(a')$ and $q(a) = c = q(a')$. Then $(\overline{p}^{-1} \circ \overline{p} \ \cap\ \overline{q}^{-1} \circ \overline{q})(a, a')$, so by $\overline{p}^{-1} \circ \overline{p} \ \cap\ \overline{q}^{-1} \circ \overline{q} = id_A$ we have $a = a'$, as required. ▮

Theorem 7. For any sets A, B, C, and any functions $p{:}A \to C$ and $q{:}B \to C$, $A \xrightarrow{p} C \xleftarrow{q} B$ is a sum diagram in the category of sets iff

$$\overline{p} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{q}^{-1} = id_C, \quad \overline{p}^{-1} \circ \overline{p} = id_A, \quad \overline{q}^{-1} \circ \overline{q} = id_B, \quad \overline{p}^{-1} \circ \overline{q} = \perp, \quad \overline{q}^{-1} \circ \overline{p} = \perp.$$

Proof. ($\Rightarrow$) Assume $A \xrightarrow{p} C \xleftarrow{q} B$ is a sum diagram.

$\underline{\overline{p} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{q}^{-1} = id_C}$. Theorem 2(i) gives $\overline{p} \circ \overline{p}^{-1} \subseteq id_C$ and $\overline{q} \circ \overline{q}^{-1} \subseteq id_C$; so $\overline{p} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{q}^{-1} \subseteq id_C$. To show that this inclusion is an equality, choose a set X with more than one element and any functions $f{:}A \to X$ and $g{:}B \to X$. Since $A \xrightarrow{p} C \xleftarrow{q} B$ is a sum diagram, there exists a unique function $i{:}C \to X$ such that $i \circ p = f$ and $i \circ q = g$. If $\overline{p} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{q}^{-1} \subset id_C$ then choose $c{\in}C$ such that $(\overline{p} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{q}^{-1})(c,c)$ does not hold. Define $j{:}C \to X$ as equal to i except at c. Then, for all $a{\in}A$, $(\overline{p} \circ \overline{p}^{-1})(p(a), p(a))$ holds, so $p(a) \neq c$, so $j(p(a)) = i(p(a)) = f(a)$. Thus $j \circ p = f$. A similar argument shows that $j \circ q = g$. This contradicts the uniqueness of i. The contradiction establishes that $\overline{p} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{q}^{-1} = id_C$.

$\underline{\overline{p}^{-1} \circ \overline{p} = id_A}$. Theorem 2(ii) gives $id_A \subseteq \overline{p}^{-1} \circ \overline{p}$. For the converse, choose a set $X = A$, a function $f{:}A \to X$ equal to the identity function on A, and an arbitrary function $g{:}B \to X$

(ignoring the case $A = \emptyset$, for which $\overline{p}^{-1} \circ \overline{p} = id_A$ is trivially true). Then by the sum property there exists a unique function $i{:}C \to X$ such that $i \circ p = f$ and $i \circ q = g$; thus $i{:}C \to A$ and $\overline{\imath} \circ \overline{p} = id_A$. Then $id_A \circ \overline{p}^{-1} \subseteq \overline{\imath} \circ id_C$ by theorem 2(iii), so $\overline{p}^{-1} \subseteq \overline{\imath}$, so $\overline{p}^{-1} \circ \overline{p} \subseteq \overline{\imath} \circ \overline{p} = id_A$, as required.

$\underline{\overline{q}^{-1} \circ \overline{q} = id_B}$. A similar argument applies.

$\underline{\overline{p}^{-1} \circ \overline{q} = \perp}$. Choose a set X with more than one element, and choose $f{:}A \to X$ and $g{:}B \to X$ with disjoint images, so that $\overline{f}^{-1} \circ \overline{g} = \perp$. By the sum property, there exists a unique function $i{:}C \to X$ such that $i \circ p = f$ and $i \circ q = g$. Then

$$\perp = \overline{f}^{-1} \circ \overline{g} = (\overline{\imath} \circ \overline{p})^{-1} \circ \overline{\imath} \circ \overline{q} = \overline{p}^{-1} \circ \overline{\imath}^{-1} \circ \overline{\imath} \circ \overline{q} \supseteq \overline{p}^{-1} \circ id_C \circ \overline{q} = \overline{p}^{-1} \circ \overline{q}$$

so $\perp = \overline{p}^{-1} \circ \overline{q}$, as required.

$\underline{\overline{q}^{-1} \circ \overline{p} = \perp}$. This follows by inverting $\overline{p}^{-1} \circ \overline{q} = \perp$.

($\Leftarrow$) Assume the five equations; I shall show $A \xrightarrow{p} C \xleftarrow{q} B$ is a sum diagram. Given any set X and functions $f{:}A \to X$ and $g{:}B \to X$, define a relation $R = \overline{f} \circ \overline{p}^{-1} \cup \overline{g} \circ \overline{q}^{-1}$. First note that

$$\begin{aligned} R \circ \overline{p} &= \overline{f} \circ \overline{p}^{-1} \circ \overline{p} \ \cup\ \overline{g} \circ \overline{q}^{-1} \circ \overline{p} = \overline{f} \circ id_A \ \cup\ \overline{g} \circ \perp = \overline{f}, \\ R \circ \overline{q} &= \overline{f} \circ \overline{p}^{-1} \circ \overline{q} \ \cup\ \overline{g} \circ \overline{q}^{-1} \circ \overline{q} = \overline{f} \circ \perp \ \cup\ \overline{g} \circ id_B = \overline{g}. \end{aligned}$$

Then

$$R \circ R^{-1} = R \circ (\overline{p} \circ \overline{f}^{-1} \ \cup\ \overline{q} \circ \overline{g}^{-1}) = R \circ \overline{p} \circ \overline{f}^{-1} \ \cup\ R \circ \overline{q} \circ \overline{g}^{-1} = \overline{f} \circ \overline{f}^{-1} \ \cup\ \overline{g} \circ \overline{g}^{-1} \subseteq id_X$$

by theorem 2(i), and

$$\begin{aligned} R^{-1} \circ R &= (\overline{p} \circ \overline{f}^{-1} \ \cup\ \overline{q} \circ \overline{g}^{-1}) \circ (\overline{f} \circ \overline{p}^{-1} \ \cup\ \overline{g} \circ \overline{q}^{-1}) \supseteq \overline{p} \circ \overline{f}^{-1} \circ \overline{f} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{g}^{-1} \circ \overline{g} \circ \overline{q}^{-1} \\ &\supseteq \overline{p} \circ id_A \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ id_B \circ \overline{q}^{-1} = \overline{p} \circ \overline{p}^{-1} \ \cup\ \overline{q} \circ \overline{q}^{-1} = id_C \end{aligned}$$

using theorem 2(ii). By theorem 5, $R = \overline{\imath}$, for some function $i{:}C \to X$. The equations $R \circ \overline{p} = \overline{f}$ and $R \circ \overline{q} = \overline{g}$ then imply $i \circ p = f$ and $i \circ q = g$.

To show the uniqueness of i, consider any function $j{:}C \to X$ satisfying $j \circ p = f$ and $j \circ q = g$. Then $\overline{\jmath} \circ \overline{p} = \overline{f}$ and $\overline{\jmath} \circ \overline{q} = \overline{g}$, so by theorem 2(iii) $\overline{f} \circ \overline{p}^{-1} \subseteq \overline{\jmath} \circ id_C = \overline{\jmath}$ and $\overline{g} \circ \overline{q}^{-1} \subseteq \overline{\jmath} \circ id_C = \overline{\jmath}$, so $\overline{\imath} = R = \overline{f} \circ \overline{p}^{-1} \ \cup\ \overline{g} \circ \overline{q}^{-1} \subseteq \overline{\jmath}$. Since i and j are both functions from C to X, this is sufficient to imply $i = j$, as required. ∎

2.3 Acyclic and connected relations

Definition. A relation R is *finite* iff there are finitely many pairs (a, b) such that $R(a, b)$ holds.

Definition. The 'not-equal-to' relation NE is defined by $NE(a, b) \Leftrightarrow a \neq b$.

Theorem 8. For any relation R,

(i) $\mathrm{dom}(R \ \cap\ NE \circ R) = \{\, a \mid \exists^{>1} b\ R(b, a) \,\}$,

(ii) $\mathrm{dom}(R \backslash NE \circ R) = \{\, a \mid \exists! b\ R(b, a) \,\}$,

(iii) $\mathrm{dom}(R \ \cap\ NE \circ R) \ \cap\ \mathrm{dom}(R \backslash NE \circ R) = \emptyset$,

(iv) $\mathrm{dom}(R \ \cap\ NE \circ R) \ \cup\ \mathrm{dom}(R \backslash NE \circ R) = \mathrm{dom}(R)$,

(v) $\mathrm{ran}(R \ \cap\ R \circ NE) = \{\, b \mid \exists^{>1} a\ R(b, a) \,\}$,

(vi) $\mathrm{ran}(R \backslash R \circ NE) = \{\, b \mid \exists! a\ R(b, a) \,\}$,

(vii) $\mathrm{ran}(R \;\cap\; R \circ NE) \;\cap\; \mathrm{ran}(R \backslash R \circ NE) = \emptyset$,
(viii) $\mathrm{ran}(R \;\cap\; R \circ NE) \;\cup\; \mathrm{ran}(R \backslash R \circ NE) = \mathrm{ran}(R)$.

Proof. (i) For any a,

$$\begin{aligned} a \in \mathrm{dom}(R \;\cap\; NE \circ R) \quad &\text{iff} \quad \exists b\; (R \;\cap\; NE \circ R)(b,a) \quad \text{iff} \quad \exists b\; R(b,a) \;\wedge\; \exists b'\; (b \neq b' \wedge R(b',a)) \\ &\text{iff} \quad \exists b, b'\; (b \neq b' \wedge R(b,a) \wedge R(b',a)) \quad \text{iff} \quad \exists^{>1} b\; R(b,a). \end{aligned}$$

(ii) For any a,

$$\begin{aligned} a \in \mathrm{dom}(R \backslash NE \circ R) \quad &\text{iff} \quad \exists b\; (R \backslash NE \circ R)(b,a) \quad \text{iff} \quad \exists b\; R(b,a) \;\wedge\; \neg \exists b'\; (b \neq b' \wedge R(b',a)) \\ &\text{iff} \quad \exists ! b\; R(b,a). \end{aligned}$$

(iii) and (iv) follow from (i) and (ii).

(v)–(viii) follow from (i)–(iv) respectively by substituting R^{-1} for R. ∎

Theorem 9. For any relations R, S, T,

$$\mathrm{dom}(R \cap T) \;\cap\; \mathrm{dom}(S \backslash NE \circ T) \subseteq \mathrm{dom}(R \cap S).$$

Proof. For any $a \in \mathrm{dom}(R \cap T) \;\cap\; \mathrm{dom}(S \backslash NE \circ T)$, there exists b such that $R(b,a)$ and $T(b,a)$ and there exists c such that $S(c,a)$ but not $(NE \circ T)(c,a)$. The last of these means that there is no d not equal to c such that $T(d,a)$; hence $b = c$. Thus we have $R(b,a)$ and $S(b,a)$, so $a \in \mathrm{dom}(R \cap S)$. ∎

Definition. A relation R is *acyclic* iff

$$\neg \exists R^*\; (\bot \neq R^* \subseteq R \;\wedge\; R^* \subseteq (R^* \circ NE \cap NE \circ R^*)).$$

(Informally, R is acyclic iff every non-empty subrelation R^* has an element of valency 1, i.e., an element related to just one element by R^* or to just one element by R^{*-1}. This is equivalent to the non-existence of a finite cyclic sequence $x_1, \ldots x_n, x_1$, with n even and $n > 2$, where the terms are all different and and related by $x_1 \xrightarrow{R} x_2 \xleftarrow{R} x_3 \xrightarrow{R} x_4 \xleftarrow{R} x_5 \xrightarrow{R} \cdots \xrightarrow{R} x_n \xleftarrow{R} x_1$. However, we shall not need this characterisation in the theory that follows.)

Theorem 10. If a relation R is acyclic and $R' \subseteq R$ then R' is also acyclic.

Proof. This is immediate from the definition. ∎

Theorem 11.

(i) If $f{:}X \to Y$, R is a relation on X, and S is a relation on Y, such that

$$S = \overline{f} \circ R \circ \overline{f}^{-1}, \qquad \overline{f}^{-1} \circ \overline{f} \;\cap R^{-1} \circ R \subseteq id_X, \qquad R \subseteq R \circ NE$$

then $S \subseteq S \circ NE$.

(ii) If $f{:}X \to Y$, R is a relation on X, and S is a relation on Y, such that

$$S = \overline{f} \circ R \circ \overline{f}^{-1}, \qquad \overline{f}^{-1} \circ \overline{f} \;\cap R \circ R^{-1} \subseteq id_X, \qquad R \subseteq NE \circ R$$

then $S \subseteq NE \circ S$.

Proof. (i) Define the universal relations U_X on X and U_Y on Y by

$$\forall a,b \quad (U_X(a,b) \quad \text{iff} \quad a \in X \wedge b \in X, \quad U_Y(a,b) \quad \text{iff} \quad a \in Y \wedge b \in Y).$$

Then $U_Y \subseteq id_Y \cup NE$, so

$$\begin{aligned} U_X &= \overline{f}^{-1} \circ U_Y \circ \overline{f} \subseteq \overline{f}^{-1} \circ (id_Y \cup NE) \circ \overline{f} = \overline{f}^{-1} \circ id_Y \circ \overline{f} \ \cup \ \overline{f}^{-1} \circ NE \circ \overline{f} \\ &= \overline{f}^{-1} \circ \overline{f} \ \cup \ \overline{f}^{-1} \circ NE \circ \overline{f} \end{aligned}$$

so

$$U_X \backslash \overline{f}^{-1} \circ \overline{f} \subseteq \overline{f}^{-1} \circ NE \circ \overline{f}. \tag{1}$$

Also, $R^{-1} \circ R \subseteq U_X$, since R is on X, and

$$\overline{f}^{-1} \circ \overline{f} \ \cap \ R^{-1} \circ R \ \cap \ NE \subseteq id_X \cap NE = \perp$$

so

$$\begin{aligned} R^{-1} \circ R \ \cap \ NE &\subseteq U_X \backslash \overline{f}^{-1} \circ \overline{f} \\ &\subseteq \overline{f}^{-1} \circ NE \circ \overline{f} \qquad \text{by (1)} \end{aligned} \tag{2}$$

so

$$\begin{aligned} S &= \overline{f} \circ R \circ \overline{f}^{-1} \\ &= \overline{f} \circ (R \cap R) \circ \overline{f}^{-1} \\ &\subseteq \overline{f} \circ (R \ \cap \ R \circ NE) \circ \overline{f}^{-1} && \text{since } R \subseteq R \circ NE \\ &\subseteq \overline{f} \circ R \circ (R^{-1} \circ R \ \cap \ NE) \circ \overline{f}^{-1} && \text{by theorem 4(ii)} \\ &\subseteq \overline{f} \circ R \circ \overline{f}^{-1} \circ NE \circ \overline{f} \circ \overline{f}^{-1} && \text{by (2)} \\ &= S \circ NE \circ \overline{f} \circ \overline{f}^{-1} && \text{since } S = \overline{f} \circ R \circ \overline{f}^{-1} \\ &\subseteq S \circ NE \circ id_Y && \text{by theorem 2(i)} \\ &\subseteq S \circ NE \end{aligned}$$

as required.

(ii) Inverting the hypotheses $S = \overline{f} \circ R \circ \overline{f}^{-1}$ and $R \subseteq NE \circ R$ gives $S^{-1} = \overline{f} \circ R^{-1} \circ \overline{f}^{-1}$ and $R^{-1} \subseteq R^{-1} \circ NE$. Applying part (i) of the theorem, with R^{-1} and S^{-1} substituted for R and S, gives $S^{-1} \subseteq S^{-1} \circ NE$, which can be inverted to $S \subseteq NE \circ S$. ▮

Theorem 12. If $f{:}X \to Y$, R is a relation on X, and S is a relation on Y, such that

$$\overline{f} \circ R \subseteq S \circ \overline{f}, \qquad \overline{f}^{-1} \circ \overline{f} \ \cap \ R^{-1} \circ R \subseteq id_X, \qquad \overline{f}^{-1} \circ \overline{f} \ \cap \ R \circ R^{-1} \subseteq id_X$$

then R is acyclic if S is.

Proof. Assuming S is acyclic, we shall infer that R is. Consider any relation R^* such that $\perp \neq R^* \subseteq R$; we must show $R^* \not\subseteq R^* \circ NE \cap NE \circ R^*$. Define a relation $S^* = \overline{f} \circ R^* \circ \overline{f}^{-1}$ on Y. Then $S^* \neq \perp$ (choose any x, x' such that $R^*(x,x')$; then $S^*(f(x), f(x'))$). Also, $\overline{f} \circ R^* \subseteq \overline{f} \circ R \subseteq S \circ \overline{f}$, so $S^* = \overline{f} \circ R^* \circ \overline{f}^{-1} \subseteq S \circ id_Y = S$ by theorem 2(iii). Since S is acyclic we conclude that $S^* \not\subseteq S^* \circ NE \ \cap \ NE \circ S^*$.

Now, if $R^* \subseteq R^* \circ NE \ \cap \ NE \circ R^*$ then $R^* \subseteq R^* \circ NE$ and $R^* \subseteq NE \circ R^*$. Also,

$$\overline{f}^{-1} \circ \overline{f} \ \cap \ R^{*-1} \circ R^* \subseteq \overline{f}^{-1} \circ \overline{f} \ \cap \ R^{-1} \circ R \subseteq id_X$$

and

$$\overline{f}^{-1} \circ \overline{f} \ \cap \ R^* \circ R^{*-1} \subseteq \overline{f}^{-1} \circ \overline{f} \ \cap \ R \circ R^{-1} \subseteq id_X$$

so, by theorem 11, $S^* \subseteq S^* \circ NE$ and $S^* \subseteq NE \circ S^*$. This gives $S^* \subseteq S^* \circ NE \ \cap \ NE \circ S^*$, which is not true.

This contradiction establishes that $R^* \not\subseteq R^* \circ NE \ \cap \ NE \circ R^*$, as required. ∎

Definition. If $f{:}X \to Y$ and R is a relation on X then R is *connected* relative to f iff, for any set Z and function $g{:}X \to Z$ such that $\overline{g} \circ R \subseteq \overline{g}$, there exists a function $i{:}f(X) \to Z$ such that $i \circ f = g$.

(Informally, R is connected relative to f iff, for any $x, x' \in X$ such that $f(x) = f(x')$, there exists a finite sequence $x = x_0, x_1, \ldots x_n = x'$ such that, for each $i \in \{1, \ldots n\}$, $R(x_{i-1}, x_i)$ or $R(x_i, x_{i-1})$ holds.)

Theorem 13. Given sets N, Σ, a function $P{:}N \to \Sigma$, and a finite relation G on N such that
(a) $G \circ G = \bot$,
(b) G is acyclic,
(c) $\overline{P} \circ G \subseteq \overline{P}$,
then
(i) $\forall X \subseteq \Sigma \ |P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| \geq 0$,
(ii) for any $X \subseteq \Sigma$, if $|P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| = 0$ then $P^{-1}(X) = \emptyset$,
(iii) if $\forall X \subseteq \Sigma \ |P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| \leq |X|$ then G is connected relative to P.

Proof. The proof is by induction on $|G|$. For the induction basis, suppose $|G| = 0$; then we can verify (i), (ii) and (iii) directly, as follows.

(i) For any $X \subseteq \Sigma$, $|P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| = |P^{-1}(X)| \geq 0$.

(ii) For any $X \subseteq \Sigma$ such that $|P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| = 0$, we have $|P^{-1}(X)| = 0$ and hence $P^{-1}(X) = \emptyset$, as required.

(iii) If $\forall X \subseteq \Sigma \ |P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| \leq |X|$ then

$$\forall \sigma \in \Sigma \quad |P^{-1}(\{\sigma\})| \leq 1$$

which means that P is injective. In that case, given any Z and $Q{:}N \to Z$, the unique function $i{:}P(N) \to Z$ such that $i \circ P = Q$ is defined by $\forall n \in N \ i(P(n)) = Q(n)$.

For the induction step, suppose that $|G| > 0$, and define

$$\begin{aligned}
G' &= G \ \cap \ G \circ NE \ \cap \ NE \circ G \\
S_1 &= \mathrm{ran}(G \backslash G \circ NE) \\
S_2 &= \mathrm{dom}((G \ \cap \ G \circ NE) \backslash NE \circ G) \\
S &= S_1 \cup S_2 \\
N' &= N \backslash S \\
P' &= P|_{N'}.
\end{aligned}$$

These have the following properties.

(α) $G' \subset G$.

Proof. This follows since G is acyclic and not equal to $\bot$.

(β) G' is a finite relation on N'.

Proof. G' is finite since $G' \subset G$ and G is finite. To show that G' is on N',

$$\begin{aligned}
\mathrm{dom}(G') \cap S_1 &\subseteq \mathrm{dom}(G) \cap \mathrm{ran}(G) = \emptyset && \text{since } G \circ G = \bot \\
\mathrm{dom}(G') \cap S_2 &\subseteq \mathrm{dom}(G \cap NE \circ G) \cap \mathrm{dom}(G \backslash NE \circ G) = \emptyset && \text{by theorem 8(iii)} \\
\mathrm{ran}(G') \cap S_1 &\subseteq \mathrm{ran}(G \cap G \circ NE) \cap \mathrm{ran}(G \backslash G \circ NE) = \emptyset && \text{by theorem 8(vii)} \\
\mathrm{ran}(G') \cap S_2 &\subseteq \mathrm{ran}(G) \cap \mathrm{dom}(G) = \emptyset && \text{since } G \circ G = \bot
\end{aligned}$$

thus $\mathrm{dom}(G') \subseteq N \backslash S = N'$ and $\mathrm{ran}(G') \subseteq N \backslash S = N'$, as required.

(γ) For every $x \in S$, there exists a unique y such that $(G \backslash G')(x,y)$ or $(G \backslash G')(y,x)$ (these two cases being mutually exclusive).

Proof. $(G \backslash G')(x,y)$ and $(G \backslash G')(y,x)$ are mutually exclusive by $(G \backslash G') \circ (G \backslash G') \subseteq G \circ G = \bot$. We have $G \backslash G \circ NE \subseteq G \backslash G'$, so

$$S_1 \subseteq \mathrm{ran}(G \backslash G'), \tag{1}$$

and

$$S_1 \cap \ \mathrm{ran}\big((G \backslash G') \cap (G \backslash G') \circ NE\big) \subseteq S_1 \cap \ \mathrm{ran}(G \cap G \circ NE) = \emptyset \tag{2}$$

by theorem 8(vii), so from (1), (2) and theorem 8(viii)

$$S_1 \subseteq \mathrm{ran}\big((G \backslash G') \backslash (G \backslash G') \circ NE\big).$$

Using theorem 8(vi), this can be written as

$$\forall x \in S_1 \, \exists! y \ (G \backslash G')(x,y). \tag{3}$$

Similarly, $(G \cap G \circ NE) \backslash NE \circ G \subseteq G \backslash G'$, so

$$S_2 \subseteq \mathrm{dom}(G \backslash G'), \tag{4}$$

and

$$S_2 \ \cap \ \mathrm{dom}\big((G \backslash G') \cap NE \circ (G \backslash G')\big) \subseteq \mathrm{dom}(G \backslash NE \circ G) \ \cap \ \mathrm{dom}(G \cap NE \circ G) = \emptyset \tag{5}$$

by theorem 8(iii), so from (4), (5) and theorem 8(iv)

$$S_2 \subseteq \mathrm{dom}\big((G \backslash G') \backslash NE \circ (G \backslash G')\big).$$

Using theorem 8(ii) this can be written as

$$\forall x \in S_2 \, \exists! y \ (G \backslash G')(y,x). \tag{6}$$

Consequently, from (3), (6), and $(G \backslash G') \circ (G \backslash G') = \bot$,

$$\forall x \in S \ \exists! y \ ((G \backslash G')(x,y) \vee (G \backslash G')(y,x))$$

as required.

(δ) $\forall x, y\ (G(x,y) \Rightarrow x \in N' \vee y \in N')$.
Proof. First,

$$\mathrm{ran}(G \cap G \circ NE)\ \cap\ S_1 = \mathrm{ran}(G \cap G \circ NE)\ \cap\ \mathrm{ran}(G \backslash G \circ NE) = \emptyset \tag{7}$$

by theorem 8(vii). Secondly,

$$\mathrm{ran}(G \cap G \circ NE) \cap S_2 = \mathrm{ran}(G \cap G \circ NE) \cap \mathrm{dom}((G \cap G \circ NE) \backslash NE \circ G) \subseteq \mathrm{ran}(G) \cap \mathrm{dom}(G) = \emptyset. \tag{8}$$

Combining (7) and (8) gives

$$\mathrm{ran}(G \cap G \circ NE)\ \cap\ S = \emptyset. \tag{9}$$

Next,

$$\mathrm{dom}(G \backslash G \circ NE)\ \cap\ S_1 = \mathrm{dom}(G \backslash G \circ NE)\ \cap\ \mathrm{ran}(G \backslash G \circ NE) \subseteq \mathrm{dom}(G)\ \cap\ \mathrm{ran}(G) = \emptyset. \tag{10}$$

Also, applying theorem 9, with $R = G \backslash G \circ NE$, $S = G \cap G \circ NE$, and $T = G$, gives

$$\mathrm{dom}(G \backslash G \circ NE)\ \cap\ \mathrm{dom}((G \cap G \circ NE) \backslash NE \circ G) \subseteq \mathrm{dom}((G \backslash G \circ NE)\ \cap\ (G \cap G \circ NE)).$$

The right-hand side is $\mathrm{dom}(\bot)$, so this simplifies to

$$\mathrm{dom}(G \backslash G \circ NE)\ \cap\ S_2 = \emptyset. \tag{11}$$

Combining (10) and (11) gives

$$\mathrm{dom}(G \backslash G \circ NE)\ \cap\ S = \emptyset. \tag{12}$$

Finally, given any x, y such that $G(x,y)$ holds, either $(G \circ NE)(x,y)$ holds, in which case $x \notin S$ by (9), or $(G \circ NE)(x,y)$ does not hold, in which case $y \notin S$ by (12). Thus $x \in N'$ or $y \in N'$.

(ε) For every x, y such that $(G \backslash G')(x,y)$ holds, either $x \in S$ or $y \in S$.
Proof. Using the definition $G' = G\ \cap\ G \circ NE\ \cap\ NE \circ G$,

$$G \backslash G' = (G \backslash G \circ NE)\ \cup\ ((G \cap G \circ NE) \backslash NE \circ G)$$

so if $(G \backslash G')(x,y)$ then $x \in \mathrm{ran}(G \backslash G \circ NE) = S_1$ or $y \in \mathrm{dom}((G \cap G \circ NE) \backslash NE \circ G) = S_2$; thus $x \in S$ or $y \in S$.

By (δ), we cannot have both $x \in S$ and $y \in S$.

(ζ) $\forall x \in S\ \exists y \in N'\ (G(x,y) \vee G(y,x))$.
Proof. Given $x \in S$, the existence of y satisfying $G(x,y) \vee G(y,x)$ is assured by (γ), and it must be in N' because of (δ).

(η) $\forall x, y \in N\ (G(x,y) \Rightarrow P(x) = P(y))$.
Proof. The hypothesis $\overline{P} \circ G \subseteq \overline{P}$ means

$$\forall \sigma, y\ ((\overline{P} \circ G)(\sigma, y) \Rightarrow \overline{P}(\sigma, y))$$

which may be expanded to

$$\forall\sigma,y\ (\exists x\ (\sigma = P(x) \wedge G(x,y)) \Rightarrow \sigma = P(y))$$

which is logically equivalent to

$$\forall x,y{\in}N\ (G(x,y) \Rightarrow P(x) = P(y))$$

since $\mathrm{dom}(P) = N$ and G is on N.

(θ) $\forall X \subseteq \Sigma\ |(G\backslash G') \circ id_{P^{-1}(X)}| = |S \cap P^{-1}(X)|$.
Proof. By (γ) and (ε), there is a bijection ω between pairs (x,y) such that $(G\backslash G')(x,y)$ holds and elements of S:

$$\omega(x,y) = x \text{ or } y, \text{ whichever one is in } S.$$

This gives $|G\backslash G'| = |S|$. Moreover, for any $X \subseteq \Sigma$, and any x,y satisfying $(G\backslash G')(x,y)$, $((G\backslash G') \circ id_{P^{-1}(X)})(x,y)$ iff $y \in P^{-1}(X)$ iff $x \in P^{-1}(X)$ (this is because $P(x) = P(y)$, from (η)), so this in turn is equivalent to $\omega(x,y) \in P^{-1}(X)$. Thus ω maps the pairs (x,y) satisfying $((G\backslash G') \circ id_{P^{-1}(X)})(x,y)$ onto $S \cap P^{-1}(X)$. The conclusion follows.

(ι) $G' \circ G' = \perp$.
Proof. This follows since $G' \subset G$ and $G \circ G = \perp$.

(κ) G' is acyclic.
Proof. This follows from $G' \subset G$ by theorem 10.

(λ) $\overline{P'} \circ G' \subseteq \overline{P'}$.
Proof. We have $G' = G' \circ id_{N'}$, $\overline{P'} = \overline{P} \circ id_{N'} \subseteq \overline{P}$, $G' \subset G$, and $\overline{P} \circ G \subseteq \overline{P}$, so

$$\overline{P'} \circ G' = \overline{P'} \circ G' \circ id_{N'} \subseteq \overline{P} \circ G \circ id_{N'} \subseteq \overline{P} \circ id_{N'} = \overline{P'}.$$

(μ) $|G'| < |G|$.
Proof. This follows from $G' \subset G$.

(ν) $\forall X \subseteq \Sigma\ |P'^{-1}(X)| - |G' \circ id_{P'^{-1}(X)}| = |P^{-1}(X)| - |G \circ id_{P^{-1}(X)}|$.
Proof. We have $P'^{-1}(X) = P^{-1}(X) \cap N' = P^{-1}(X)\backslash S$ and hence

$$|P'^{-1}(X)| = |P^{-1}(X)| - |S \cap P^{-1}(X)|.$$

Also, $G' \circ id_{P'^{-1}(X)} = G' \circ id_{N'} \circ id_{P^{-1}(X)} = G' \circ id_{P^{-1}(X)}$, since G' is on N', so

$$|G' \circ id_{P'^{-1}(X)}| = |G' \circ id_{P^{-1}(X)}| = |G \circ id_{P^{-1}(X)}| - |(G\backslash G') \circ id_{P^{-1}(X)}| = |G \circ id_{P^{-1}(X)}| - |S \cap P^{-1}(X)|$$

by (θ). Hence

$$|P'^{-1}(X)| - |G' \circ id_{P'^{-1}(X)}| = |P^{-1}(X)| - |G \circ id_{P^{-1}(X)}|.$$

(ξ) $P^{-1}(X) = \emptyset$ if $P'^{-1}(X) = \emptyset$.

Proof. I shall prove the contrapositive. If $P^{-1}(X) \neq \emptyset$ then there exists $x \in N$ such that $P(x) \in X$. If $x \in N'$ then $x \in P'^{-1}(X)$, whereas if $x \notin N'$ then $x \in S$, so by (ζ) there exists $y \in N'$ such that $G(x,y)$ or $G(y,x)$. By (η), $P(y) = P(x) \in X$, so $y \in P'^{-1}(X)$. In either case, $P'^{-1}(X) \neq \emptyset$.

(ο) $P'(N') = P(N)$.

Proof. Recall that $P' = P|_{N'}$, so $P'(N') = P(N')$. By (ζ), for every $x \in S$ there exists $y \in N'$ such that $G(x,y)$ or $G(y,x)$; by (η), this implies $P(x) = P(y)$. This shows that $P(S) \subseteq P(N')$. Since $N = S \cup N'$, this gives $P(N) = P(S) \cup P(N') = P(N')$. Hence $P'(N') = P(N)$.

(π) G is connected relative to P if G' is connected relative to P'.

Proof. Consider any set Z and any function $Q{:}N \to Z$ such that $\overline{Q} \circ G \subseteq \overline{Q}$. Define $Q' = Q|_{N'}{:}N' \to Z$, and note that

$$\overline{Q'} \circ G' = \overline{Q'} \circ G' \circ id_{N'} \subseteq \overline{Q} \circ G \circ id_{N'} \subseteq \overline{Q} \circ id_{N'} = \overline{Q'}.$$

Assuming G' is connected relative to P', there exists a function $i{:}P'(N') \to Z$ such that $i \circ P' = Q'$; this means that $i \circ P$ agrees with Q on N'. By (ο), we have $i{:}P(N) \to Z$. Moreover I claim that $i \circ P = Q$. For any $x \in S$, by (ζ) there exists $y \in N'$ such that $G(x,y)$ or $G(y,x)$. Then (η) gives $P(x) = P(y)$, and by similar reasoning the hypothesis $\overline{Q} \circ G \subseteq \overline{Q}$ gives $Q(x) = Q(y)$. Then

$$i(P(x)) = i(P(y)) = Q(y) = Q(x).$$

So $i \circ P = Q$ holds on S as well as on N', i.e., it holds on the whole of N. Thus we conclude that $i{:}P(N) \to Z$ and $i \circ P = Q$, as required.

Now we are ready to apply the induction step of the theorem. By (β), (ι)–(μ), we can apply the inductive hypothesis to N', P', G', and thereby verify (i), (ii) and (iii) for N, P, G:

(i) for any $X \subseteq \Sigma$, applying (ν), $|P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| = |P'^{-1}(X)| - |G' \circ id_{P'^{-1}(X)}| \geq 0$ by inductive hypothesis;

(ii) for any $X \subseteq \Sigma$, if $|P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| = 0$ then $|P'^{-1}(X)| - |G' \circ id_{P'^{-1}(X)}| = 0$ by (ν), so $P'^{-1}(X) = \emptyset$ by inductive hypothesis, so $P^{-1}(X) = \emptyset$ by (ξ);

(iii) if $\forall X \subseteq \Sigma\ |P^{-1}(X)| - |G \circ id_{P^{-1}(X)}| \leq |X|$ then $\forall X \subseteq \Sigma\ |P'^{-1}(X)| - |G' \circ id_{P'^{-1}(X)}| \leq |X|$ by (ν), so G' is connected relative to P' by inductive hypothesis, so G is connected relative to P by (π). ∎

2.4 Minimal relations relative to a function

Definition. If $f{:}X \to Y$ and R is a relation on X then R is *minimal* relative to f iff

$$\forall R^* \subseteq R \ \ (\overline{f} \circ R^* = \overline{f} \circ R \Rightarrow R^* = R).$$

(Informally, R is minimal relative to f iff, for any $x \in X$ and $y \in Y$, there exists at most one $x' \in X$ such that $f(x') = y$ and $R(x', x)$.)

Theorem 14. If $f{:}X \to Y$ and R is a relation on X then R is minimal relative to f iff

$$\overline{f}^{-1} \circ \overline{f} \ \cap \ R \circ R^{-1} \subseteq id_X.$$

Proof. ($\Rightarrow$) Suppose R is minimal relative to f. For any x, x' such that $(\overline{f}^{-1} \circ \overline{f} \cap R \circ R^{-1})(x, x')$, we have $x, x' \in X$, $f(x) = f(x')$, and there exists z such that $R(x,z)$ and $R(x',z)$. Now suppose $x \neq x'$. Define a relation R^* equal to R except that $R^*(x,z)$ does not hold. Thus $R^* \subset R$, but $\overline{f} \circ R^* = \overline{f} \circ R$, since $(\overline{f} \circ R^*)(f(x), z)$ still holds (via x') and $\overline{f} \circ R^*$ is identical to $\overline{f} \circ R$ in all other respects. The existence of R^* contradicts the minimality of R. This contradiction demonstrates that $x = x'$, as required.

($\Leftarrow$) Suppose $\overline{f}^{-1} \circ \overline{f} \cap R \circ R^{-1} \subseteq id_X$. Consider any $R^* \subseteq R$ such that $\overline{f} \circ R^* = \overline{f} \circ R$. Then

$$
\begin{aligned}
R &= id_X \circ R \ \cap \ R \\
&\subseteq \overline{f}^{-1} \circ \overline{f} \circ R \ \cap \ R && \text{by theorem 2(ii)} \\
&= \overline{f}^{-1} \circ \overline{f} \circ R^* \ \cap \ R && \text{since } \overline{f} \circ R^* = \overline{f} \circ R \\
&\subseteq (\overline{f}^{-1} \circ \overline{f} \ \cap \ R \circ R^{*-1}) \circ R^* && \text{by theorem 4(i)} \\
&\subseteq (\overline{f}^{-1} \circ \overline{f} \ \cap \ R \circ R^{-1}) \circ R^* && \text{since } R^* \subseteq R \\
&\subseteq id_X \circ R^* && \text{since } \overline{f}^{-1} \circ \overline{f} \cap R \circ R^{-1} \subseteq id_X \\
&= R^*
\end{aligned}
$$

so $R^* = R$, as required. ∎

3. Networks and Homomorphisms

3.1 Introduction

My aim is to state formally the *recognition problem* (the problem of interpreting an image using a grammar) and to define the *recognition process* that solves the problem. Because the theory is complex I shall develop it by stepwise refinement, beginning in this section with an outline statement of the recognition problem and refining it in the following section, then building up the recognition process incrementally.

A grammar will be represented by a formal structure called a *network*, containing symbol types. Parsing consists of constructing another network called a *pattern*, containing symbol tokens, and establishing a homomorphism from the pattern network to the grammar network. The grammar must satisfy certain conditions: it must be a *semi-definite* network. The pattern is not bound by these conditions during recognition, but by the end it must satisfy a stronger set of conditions: it must be a *definite* network. All these concepts are defined in this section.

3.2 Definition of network and homomorphism

Definition. A *network* is a 12-tuple $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, where Σ, N, H, E, K are disjoint finite sets; $W{:}N \cup \Sigma \to \Sigma$, $P{:}N \to \Sigma$, $A{:}H \to N \cup \Sigma$, $F,S{:}E \to H$ and $C{:}K \to E$ are functions such that $\forall \sigma \in \Sigma\ W(\sigma) = \sigma$; and G is a relation on $\Sigma \cup N \cup H \cup E \cup K$ such that

(1) $id_\Sigma \circ G = G \circ id_\Sigma$, $id_N \circ G = G \circ id_N$, $id_H \circ G = G \circ id_H$, $id_E \circ G = G \circ id_E$, $id_K \circ G = G \circ id_K$;

(2) $\overline{W} \circ G = G \circ \overline{W}$, $\overline{P} \circ G \subseteq \overline{P}$, $\overline{A} \circ G \subseteq G \circ \overline{A}$, $\overline{F} \circ G \subseteq G \circ \overline{F}$, $\overline{S} \circ G \subseteq G \circ \overline{S}$, $\overline{C} \circ G \subseteq G \circ \overline{C}$;

(3) G_H and G_H^{-1} are minimal relative to A; G_K and G_K^{-1} are minimal relative to C; G_E and G_E^{-1} are minimal relative to F and S (where $G_H = G \circ id_H$, $G_K = G \circ id_K$ and $G_E = G \circ id_E$);

(4) $G \circ G = \perp$.

The elements of Σ, N, H, E, K are called *symbols*, *nodes*, *hooks*, *edges* and *facets*, respectively. G is called the *gluing relation*; $G(x,y)$ means that y is glued to x, i.e., y is a subsymbol, subnode, subhook, subedge or subfacet of x, i.e., x is a supersymbol, supernode, superhook, superedge or superfacet of y. The functions W, P, A, F, S, C express the incidence relations: a node n connects a part $P(n)$ to a whole $W(n)$; a hook h is attached to a node (or possibly a symbol) $A(h)$; an edge e runs from its *first* hook $F(e)$ to its *second* hook $S(e)$; a facet k belongs to the edge $C(k)$.

The clauses of the definition may be paraphrased informally as follows.

(1) The gluing relation G may be considered as the disjoint union of a relation $G_\Sigma = G \circ id_\Sigma$ on symbols, a relation $G_N = G \circ id_N$ on nodes, a relation $G_H = G \circ id_H$ on hooks, a relation $G_E = G \circ id_E$ on edges, and a relation $G_K = G \circ id_K$ on facets. I shall always use similar notation below for the parts of a gluing relation of a network: e.g., if $(\Sigma_2, N_2, H_2, E_2, K_2, W_2, P_2, A_2, F_2, S_2, C_2, G_2)$ is a network then $G_{2\Sigma} = G_2 \circ id_{\Sigma_2}$ (the symbol part of G_2), $G_{2N} = G_2 \circ id_{N_2}$ (the node part of G_2), etc..

(2) G preserves incidence. e.g., if a hook h_1 is glued to a hook h_2 then the node $A(h_1)$ is glued to the node $A(h_2)$. (The first equation also says that every node of a subsymbol is a subnode.)

(3) A hook is glued to at most one hook of any given node; a facet is glued to at most one facet of any given edge; and an edge is glued to at most one edge of any given hook (for each edge direction).

(4) A subsymbol, subnode, subhook, subedge or subfacet cannot be also be a supersymbol, supernode, superhook, superedge or superfacet.

Definition. If $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ and $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ are networks, a *homomorphism* $p\colon \mathcal{N}_1 \to \mathcal{N}_0$ is a function from $\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1$ to $\Sigma_0 \cup N_0 \cup H_0 \cup E_0 \cup K_0$ such that

(1) $p(\Sigma_1) \subseteq \Sigma_0, \quad p(N_1) \subseteq N_0, \quad p(H_1) \subseteq H_0, \quad p(E_1) \subseteq E_0, \quad p(K_1) \subseteq K_0$;

(2) $W_0 \circ p = p \circ W_1, \quad P_0 \circ p = p \circ P_1, \quad F_0 \circ p = p \circ F_1, \quad S_0 \circ p = p \circ S_1$;

(3) $\begin{array}{ccc} N_0 \cup \Sigma_0 & \xleftarrow{p|_{N_1 \cup \Sigma_1}} & N_1 \cup \Sigma_1 \\ {\scriptstyle A_0}\uparrow & & \uparrow{\scriptstyle A_1} \\ H_0 & \xleftarrow{p|_{H_1}} & H_1 \end{array}$ and $\begin{array}{ccc} E_0 & \xleftarrow{p|_{E_1}} & E_1 \\ {\scriptstyle C_0}\uparrow & & \uparrow{\scriptstyle C_1} \\ K_0 & \xleftarrow{p|_{K_1}} & K_1 \end{array}$ are pullbacks in the category of sets;

(4) $\overline{p} \circ G_1 = G_0 \circ \overline{p}$;

(5) G_1 is minimal relative to p.

The clauses of this definition may be paraphrased informally as follows.

(1) p maps symbols to symbols, nodes to nodes, hooks to hooks, edges to edges, and facets to facets.

(2) p preserves the W, P, F, S incidence functions.

(3) p maps the hooks of any node n bijectively onto the hooks of $p(n)$, and maps the facets of any edge e bijectively onto the facets of $p(e)$.

(4) p preserves gluing (i.e., if $G_1(x,y)$ then $G_0(p(x),p(y))$), and if $p(x)$ is a subsymbol, subnode, subhook, subedge or subfacet then so is x.

(5) The gluings in $\mathcal{N}_1$ are induced by those in $\mathcal{N}_0$, i.e., two things are glued in $\mathcal{N}_1$ only if they are forced to be by condition (4).

Theorem 15. If $p\colon \mathcal{N}_1 \to \mathcal{N}_0$ and $q\colon \mathcal{N}_2 \to \mathcal{N}_1$ are homomorphisms then so is $p \circ q\colon \mathcal{N}_2 \to \mathcal{N}_0$.

Proof.

First,

$$p(q(\Sigma_2)) \subseteq p(\Sigma_1) \subseteq \Sigma_0, \quad p(q(N_2)) \subseteq p(N_1) \subseteq N_0, \quad p(q(H_2)) \subseteq p(H_1) \subseteq H_0,$$
$$p(q(E_2)) \subseteq p(E_1) \subseteq E_0, \quad p(q(K_2)) \subseteq p(K_1) \subseteq K_0,$$

as required.

Secondly,

$$W_0 \circ p \circ q = p \circ W_1 \circ q = p \circ q \circ W_2, \quad P_0 \circ p \circ q = p \circ P_1 \circ q = p \circ q \circ P_2,$$
$$F_0 \circ p \circ q = p \circ F_1 \circ q = p \circ q \circ F_2, \quad S_0 \circ p \circ q = p \circ S_1 \circ q = p \circ q \circ S_2,$$

as required.

Thirdly, $\begin{array}{ccc} N_0 \cup \Sigma_0 & \xleftarrow{p|_{N_1 \cup \Sigma_1}} & N_1 \cup \Sigma_1 \\ {\scriptstyle A_0}\uparrow & & \uparrow{\scriptstyle A_1} \\ H_0 & \xleftarrow{p|_{H_1}} & H_1 \end{array}$ and $\begin{array}{ccc} N_1 \cup \Sigma_1 & \xleftarrow{q|_{N_2 \cup \Sigma_2}} & N_2 \cup \Sigma_2 \\ {\scriptstyle A_1}\uparrow & & \uparrow{\scriptstyle A_2} \\ H_1 & \xleftarrow{q|_{H_2}} & H_2 \end{array}$ are pullbacks, and therefore so is $\begin{array}{ccc} N_0 \cup \Sigma_0 & \xleftarrow{(p \circ q)|_{N_2 \cup \Sigma_2}} & N_2 \cup \Sigma_2 \\ {\scriptstyle A_0}\uparrow & & \uparrow{\scriptstyle A_2} \\ H_0 & \xleftarrow{(p \circ q)|_{H_2}} & H_2 \end{array}$. This is proved as follows. We have

$$A_0 \circ (p \circ q)|_{H_2} = A_0 \circ p|_{H_1} \circ q|_{H_2} = p|_{N_1 \cup \Sigma_1} \circ A_1 \circ q|_{H_2} = p|_{N_1 \cup \Sigma_1} \circ q|_{N_2 \cup \Sigma_2} \circ A_2 = (p \circ q)|_{N_2 \cup \Sigma_2} \circ A_2.$$

Also, for any set X and functions $f\colon X \to H_0$ and $g\colon X \to N_2 \cup \Sigma_2$ such that $A_0 \circ f = (p \circ q)|_{N_2 \cup \Sigma_2} \circ g$, by the first pullback there exists a unique $i\colon X \to H_1$ such that $p|_{H_1} \circ i = f$ and $A_1 \circ i = q|_{N_2 \cup \Sigma_2} \circ g$.

Then, by the second pullback, there exists a unique $j{:}X \to H_2$ such that $q|_{H_2} \circ j = i$ and $A_2 \circ j = g$. Thus j satisfies the desired equations $(p \circ q)|_{H_2} \circ j = f$ and $A_2 \circ j = g$. To show that j is the only such function, consider any function $k{:}X \to H_2$ such that $(p \circ q)|_{H_2} \circ k = f$ and $A_2 \circ k = g$. Then $p|_{H_1} \circ q|_{H_2} \circ k = f$ and $A_1 \circ q|_{H_2} \circ k = q|_{N_2 \cup \Sigma_2} \circ A_2 \circ k = q|_{N_2 \cup \Sigma_2} \circ g$, so, by the uniqueness clause in the first pullback, $q|_{H_2} \circ k = i$. Then, by the uniqueness clause in the second pullback, $k = j$. This verifies the required pullback condition.

Similarly, $\begin{tikzcd} E_0 & E_1 \arrow[l, "p|_{E_1}"'] \\ K_0 \arrow[u, "C_0"] & K_1 \arrow[l, "p|_{K_1}"] \arrow[u, "C_1"'] \end{tikzcd}$ and $\begin{tikzcd} E_1 & E_2 \arrow[l, "q|_{E_2}"'] \\ K_1 \arrow[u, "C_1"] & K_2 \arrow[l, "q|_{K_2}"] \arrow[u, "C_2"'] \end{tikzcd}$ are pullbacks, and hence so is $\begin{tikzcd} E_0 & E_2 \arrow[l, "(p\circ q)|_{E_2}"'] \\ K_0 \arrow[u, "C_0"] & K_2 \arrow[l, "(p\circ q)|_{K_2}"] \arrow[u, "C_2"'] \end{tikzcd}$.

Fourthly, $\overline{p \circ q} \circ G_2 = \overline{p} \circ \overline{q} \circ G_2 = \overline{p} \circ G_1 \circ \overline{q} = G_0 \circ \overline{p} \circ \overline{q} = G_o \circ \overline{p \circ q}$.

Fifthly, G_1 is minimal relative to p and G_2 is minimal relative to q; we must show that G_2 is minimal relative to $p \circ q$. By theorem 14, $\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ G_1^{-1} \subseteq id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1}$ and $\overline{q}^{-1} \circ \overline{q} \ \cap \ G_2 \circ G_2^{-1} \subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2}$. Hence,

$$\begin{aligned}
&\overline{p \circ q}^{-1} \circ \overline{p \circ q} \ \cap \ G_2 \circ G_2^{-1} \\
&\qquad = \overline{q}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \circ \overline{q} \ \cap \ G_2 \circ G_2^{-1} \\
&\qquad \subseteq \overline{q}^{-1} \circ \left(\overline{p}^{-1} \circ \overline{p} \ \cap \ \overline{q} \circ G_2 \circ G_2^{-1} \circ \overline{q}^{-1}\right) \circ \overline{q} && \text{by theorem 4(i),(ii)} \\
&\qquad = \overline{q}^{-1} \circ \left(\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ \overline{q} \circ \overline{q}^{-1} \circ G_1^{-1}\right) \circ \overline{q} && \text{since } q \text{ is a homomorphism} \\
&\qquad \subseteq \overline{q}^{-1} \circ \left(\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ G_1^{-1}\right) \circ \overline{q} && \text{by theorem 2(i)} \\
&\qquad \subseteq \overline{q}^{-1} \circ id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1} \circ \overline{q} \\
&\qquad = \overline{q}^{-1} \circ \overline{q}.
\end{aligned}$$

Hence

$$\begin{aligned}
\overline{p \circ q}^{-1} \circ \overline{p \circ q} \ \cap \ G_2 \circ G_2^{-1} &\subseteq \overline{q}^{-1} \circ \overline{q} \ \cap \ G_2 \circ G_2^{-1} \\
&\subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2}.
\end{aligned}$$

By theorem 14 again, G_2 is minimal relative to $p \circ q$, as required. ∎

Theorem 16. If $f{:}\mathcal{N} \to \mathcal{N}'$ is a homomorphism and is also a bijective function then the inverse function $f^{-1}{:}\mathcal{N}' \to \mathcal{N}$ is a homomorphism (called the *inverse homomorphism* of f).

Proof. First, the condition $f^{-1}(\Sigma') \subseteq \Sigma$ follows from $f(N) \subseteq N'$, $f(H) \subseteq H'$, $f(E) \subseteq E'$ and $f(K) \subseteq K'$. The similar conditions $f^{-1}(N') \subseteq N$, $f^{-1}(H') \subseteq H$, $f^{-1}(E') \subseteq E$ and $f^{-1}(K') \subseteq K$ follow similarly.

Secondly, the condition $W \circ f^{-1} = f^{-1} \circ W'$ is verified by $W \circ f^{-1} = f^{-1} \circ f \circ W \circ f^{-1} = f^{-1} \circ W' \circ f \circ f^{-1} = f^{-1} \circ W'$. The similar conditions $P \circ f^{-1} = f^{-1} \circ P'$, $F \circ f^{-1} = f^{-1} \circ F'$ and $S \circ f^{-1} = f^{-1} \circ S'$ follow similarly.

Thirdly, the pullback $\begin{tikzcd} N\cup\Sigma & N'\cup\Sigma' \arrow[l, "f^{-1}|_{N'\cup\Sigma'}"'] \\ H \arrow[u, "A"] & H' \arrow[l, "f^{-1}|_{H'}"] \arrow[u, "A'"'] \end{tikzcd}$ is verified as follows. The equation $A \circ f^{-1}|_{H'} = f^{-1}|_{N' \cup \Sigma'} \circ A'$ is verified as in the previous paragraph. To verify the other half of the pullback property, given any set X and functions $p{:}X \to H$ and $q{:}X \to N' \cup \Sigma'$ such that $A \circ p = f^{-1}|_{N' \cup \Sigma'} \circ q$, define $i = f|_H \circ p{:}X \to H'$, giving $f^{-1}|_{H'} \circ i = f^{-1}|_{H'} \circ f|_H \circ p = p$ and $A' \circ i = A' \circ f|_H \circ p = f|_{N \cup \Sigma} \circ A \circ p = f|_{N \cup \Sigma} \circ f^{-1}|_{N' \cup \Sigma'} \circ q = q$, as required. To show i is unique, consider any $j{:}X \to H'$ such that $f^{-1}|_{H'} \circ j = p$ and $A' \circ j = q$; then $j = f|_H \circ p = i$, as required.

The pullback $\begin{tikzcd} E & E' \arrow[l, "f^{-1}|_{E'}"'] \\ K \arrow[u, "C"] & K' \arrow[l, "f^{-1}|_{K'}"] \arrow[u, "C'"'] \end{tikzcd}$ is verified similarly.

Fourthly, the condition $\overline{f^{-1}} \circ G' = G \circ \overline{f^{-1}}$ is verified by

$$\overline{f^{-1}} \circ G' = \overline{f^{-1}} \circ G' \circ \overline{f \circ f^{-1}} = \overline{f^{-1}} \circ G' \circ \overline{f} \circ \overline{f^{-1}} = \overline{f^{-1}} \circ \overline{f} \circ G \circ \overline{f^{-1}} = \overline{f^{-1} \circ f} \circ G \circ \overline{f^{-1}} = G \circ \overline{f^{-1}}.$$

Finally the condition that G' be minimal relative to f^{-1} is verified as follows. For any relation $R^* \subseteq G'$, if $\overline{f^{-1}} \circ R^* = \overline{f^{-1}} \circ G'$ then $\overline{f} \circ \overline{f^{-1}} \circ R^* = \overline{f} \circ \overline{f^{-1}} \circ G'$, so, since $\overline{f} \circ \overline{f^{-1}} = \overline{f \circ f^{-1}} = id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$, we have $R^* = G'$, as required. ∎

Definition. An *isomorphism* is a homomorphism $f: \mathcal{N} \to \mathcal{N}'$ with an inverse homomorphism $f^{-1}: \mathcal{N}' \to \mathcal{N}$. An *automorphism* of $\mathcal{N}$ is an isomorphism from $\mathcal{N}$ to $\mathcal{N}$.

3.3 Minimal relations relative to a network

Definition. If $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is a network then its gluing relation G is *minimal* relative to $\mathcal{N}$ iff

$$id_K \subseteq G_K \circ G_K^{-1} \cup G_K^{-1} \circ G_K$$

and, for any relation $G^* \subseteq G$ such that

- $\overline{W} \circ G^* \subseteq G^* \circ \overline{W}$, $\overline{P} \circ G^* \subseteq \overline{P}$, $\overline{A} \circ G^* \subseteq G^* \circ \overline{A}$, $\overline{F} \circ G^* \subseteq G^* \circ \overline{F}$, $\overline{S} \circ G^* \subseteq G^* \circ \overline{S}$, $\overline{C} \circ G^* \subseteq G^* \circ \overline{C}$,
- $id_K \subseteq G_K^* \circ {G_K^*}^{-1} \cup {G_K^*}^{-1} \circ G_K^*$ (where $G_K^* = G^* \circ id_K$),

we have $G^* = G$.

(Informally, the condition that G be minimal relative to $\mathcal{N}$ means that every facet is glued to another facet (i.e., it is either a subfacet or a superfacet) and G is as small as it can be, subject to this constraint.)

Theorem 17. For any network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$,

(i) if G is minimal relative to $\mathcal{N}$ then

$$\begin{aligned} G_\Sigma &= \overline{W} \circ G_N \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1}, \qquad & G_N &= id_N \circ \overline{A} \circ G_H \circ \overline{A}^{-1}, \\ G_H &= \overline{F} \circ G_E \circ \overline{F}^{-1} \ \cup \ \overline{S} \circ G_E \circ \overline{S}^{-1}, & G_E &= \overline{C} \circ G_K \circ \overline{C}^{-1}; \end{aligned} \tag{$*$}$$

(ii) if $G_K \circ G_K^{-1} \subseteq id_K \subseteq G_K \circ G_K^{-1} \cup G_K^{-1} \circ G_K$ then the converse of (i) holds, i.e., equations $(*)$ imply that G is minimal relative to $\mathcal{N}$.

Proof. (i) Suppose G is minimal relative to $\mathcal{N}$. Define a relation $G^* = G_K^* \cup G_E^* \cup G_H^* \cup G_N^* \cup G_\Sigma^*$, where

$$\begin{aligned} G_K^* &= G_K \\ G_E^* &= \overline{C} \circ G_K^* \circ \overline{C}^{-1} \\ G_H^* &= \overline{F} \circ G_E^* \circ \overline{F}^{-1} \cup \overline{S} \circ G_E^* \circ \overline{S}^{-1} \\ G_N^* &= id_N \circ \overline{A} \circ G_H^* \circ \overline{A}^{-1} \\ G_\Sigma^* &= \overline{W} \circ G_N^* \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G_H^* \circ \overline{A}^{-1}. \end{aligned}$$

First I shall show that $G^* \subseteq G$. We already have $G^*_K = G_K$, but also

$$\begin{aligned}
G^*_E &= \overline{C} \circ G^*_K \circ \overline{C}^{-1} = \overline{C} \circ G_K \circ \overline{C}^{-1} = \overline{C} \circ G \circ \overline{C}^{-1} \subseteq G \circ \overline{C} \circ \overline{C}^{-1} \subseteq G \circ id_E = G_E \\
G^*_H &= \overline{F} \circ G^*_E \circ \overline{F}^{-1} \cup \overline{S} \circ G^*_E \circ \overline{S}^{-1} \subseteq \overline{F} \circ G_E \circ \overline{F}^{-1} \cup \overline{S} \circ G_E \circ \overline{S}^{-1} = \overline{F} \circ G \circ \overline{F}^{-1} \cup \overline{S} \circ G \circ \overline{S}^{-1} \\
&\subseteq G \circ \overline{F} \circ \overline{F}^{-1} \cup G \circ \overline{S} \circ \overline{S}^{-1} \subseteq G \circ id_H \cup G \circ id_H = G_H \\
G^*_N &= id_N \circ \overline{A} \circ G^*_H \circ \overline{A}^{-1} \subseteq id_N \circ \overline{A} \circ G_H \circ \overline{A}^{-1} = id_N \circ \overline{A} \circ G \circ \overline{A}^{-1} \subseteq id_N \circ G \circ \overline{A} \circ \overline{A}^{-1} \\
&\subseteq id_N \circ G \circ id_{N \cup \Sigma} = G_N \\
G^*_\Sigma &= \overline{W} \circ G^*_N \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G^*_H \circ \overline{A}^{-1} \subseteq \overline{W} \circ G_N \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1} \\
&\subseteq \overline{W} \circ G \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G \circ \overline{A}^{-1} \subseteq G \circ \overline{W} \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ G \circ \overline{A} \circ \overline{A}^{-1} \\
&\subseteq G \circ id_\Sigma \ \cup \ id_\Sigma \circ G \circ id_{N \cup \Sigma} = G_\Sigma \cup G_\Sigma = G_\Sigma.
\end{aligned}$$

Thus $G^* \subseteq G$ as claimed. Note also that G^*_K is on K, G^*_E is on E, G^*_H is on H, G^*_N is on N, and G^*_Σ is on Σ.

Secondly I shall show that G^* is homomorphic. This follows from

$$\begin{aligned}
\overline{W} \circ G^* &= \overline{W} \circ (G^*_\Sigma \cup G^*_N) \circ id_{\Sigma \cup N} \\
&\subseteq \overline{W} \circ (G^*_\Sigma \cup G^*_N) \circ \overline{W}^{-1} \circ \overline{W} && \text{by theorem 2(ii)} \\
&= \overline{W} \circ G^*_\Sigma \circ \overline{W}^{-1} \circ \overline{W} \ \cup \ \overline{W} \circ G^*_N \circ \overline{W}^{-1} \circ \overline{W} && \text{by theorem 3(i),(ii)} \\
&\subseteq id_\Sigma \circ G^*_\Sigma \circ id_\Sigma \circ \overline{W} \ \cup \ G^*_\Sigma \circ \overline{W} && \text{since } \overline{W|_\Sigma} = id_\Sigma \text{ and } \overline{W} \circ G^*_N \circ \overline{W}^{-1} \subseteq G^*_\Sigma \\
&= G^*_\Sigma \circ \overline{W} \ \cup \ G^*_\Sigma \circ \overline{W} \\
&= G^* \circ \overline{W}
\end{aligned}$$

and

$$\begin{aligned}
\overline{A} \circ G^* &= id_{N \cup \Sigma} \circ \overline{A} \circ G^*_H \circ id_H \\
&\subseteq id_{N \cup \Sigma} \circ \overline{A} \circ G^*_H \circ \overline{A}^{-1} \circ \overline{A} && \text{by theorem 2(ii)} \\
&\subseteq id_N \circ \overline{A} \circ G^*_H \circ \overline{A}^{-1} \circ \overline{A} \ \cup \ id_\Sigma \circ \overline{A} \circ G^*_H \circ \overline{A}^{-1} \circ \overline{A} && \text{by theorem 3(i)} \\
&\subseteq G^*_N \circ \overline{A} \ \cup \ G^*_\Sigma \circ \overline{A} && \text{by definition of } G^*_N \text{ and } G^*_\Sigma \\
&= G^* \circ \overline{A}
\end{aligned}$$

and

$$\begin{aligned}
\overline{P} \circ G^* &\subseteq \overline{P} \circ G \subseteq \overline{P} \\
\overline{F} \circ G^* &= \overline{F} \circ G^*_E \circ id_E \subseteq \overline{F} \circ G^*_E \circ \overline{F}^{-1} \circ \overline{F} \subseteq G^*_H \circ \overline{F} = G^* \circ \overline{F} \\
\overline{S} \circ G^* &= \overline{S} \circ G^*_E \circ id_E \subseteq \overline{S} \circ G^*_E \circ \overline{S}^{-1} \circ \overline{S} \subseteq G^*_H \circ \overline{S} = G^* \circ \overline{S} \\
\overline{C} \circ G^* &= \overline{C} \circ G^*_K \circ id_K \subseteq \overline{C} \circ G^*_K \circ \overline{C}^{-1} \circ \overline{C} = G^*_E \circ \overline{C} = G^* \circ \overline{C}.
\end{aligned}$$

The relation G^* satisfies the condition

$$id_K \subseteq G^*_K \circ {G^*_K}^{-1} \cup {G^*_K}^{-1} \circ G^*_K$$

just like G. Hence, by the minimality condition, $G^* = G$. This shows that G satisfies the equations $(*)$.

(ii) Suppose that $G_K \circ G_K^{-1} \subseteq id_K \subseteq G_K \circ G_K^{-1} \cup G_K^{-1} \circ G_K$ and G satisfies the equations $(*)$; we must show that G is minimal relative to $\mathcal{N}$. Consider any homomorphic relation $G^* \subseteq G$; we can split G^* into its five parts, $G^*_K = G^* \circ id_K \subseteq G_K$, $G^*_E = G^* \circ id_E \subseteq G_E$, $G^*_H = G^* \circ id_H \subseteq G_H$, $G^*_N = G^* \circ id_N \subseteq G_N$, and $G^*_\Sigma = G^* \circ id_\Sigma \subseteq G_\Sigma$, which are on K, E, H, N and Σ respectively. Assume that $id_K \subseteq G^*_K \circ {G^*_K}^{-1} \cup {G^*_K}^{-1} \circ G^*_K$. We must show $G^* = G$.

First we show $G^*_K = G_K$. This follows from $G^*_K \subseteq G_K$ and

$$\begin{aligned}
G_K &= G_K \circ id_K \subseteq G_K \circ (G^*_K \circ {G^*_K}^{-1} \cup {G^*_K}^{-1} \circ G^*_K) && \text{since } id_K \subseteq G^*_K \circ {G^*_K}^{-1} \cup {G^*_K}^{-1} \circ G^*_K \\
&= G_K \circ G^*_K \circ {G^*_K}^{-1} \ \cup \ G_K \circ {G^*_K}^{-1} \circ G^*_K \\
&\subseteq G \circ G \circ {G^*_K}^{-1} \ \cup \ G_K \circ G_K^{-1} \circ G^*_K \\
&\subseteq \bot \ \cup \ id_K \circ G^*_K && \text{since } G \circ G = \bot \text{ and } G_K \circ G_K^{-1} \subseteq id_K \\
&= G^*_K.
\end{aligned}$$

Next we have $G^*_E = G_E$, since $G^*_E \subseteq G_E$ and

$$G_E = \overline{C} \circ G_K \circ \overline{C}^{-1} = \overline{C} \circ G^*_K \circ \overline{C}^{-1} = \overline{C} \circ G^* \circ \overline{C}^{-1} \subseteq G^* \circ \overline{C} \circ \overline{C}^{-1} \subseteq G^* \circ id_E = G^*_E.$$

Then $G^*_H = G_H$, since $G^*_H \subseteq G_H$ and

$$\begin{aligned}
G_H &= \overline{F} \circ G_E \circ \overline{F}^{-1} \cup \overline{S} \circ G_E \circ \overline{S}^{-1} = \overline{F} \circ G^*_E \circ \overline{F}^{-1} \cup \overline{S} \circ G^*_E \circ \overline{S}^{-1} = \overline{F} \circ G^* \circ \overline{F}^{-1} \cup \overline{S} \circ G^* \circ \overline{S}^{-1} \\
&\subseteq G^* \circ \overline{F} \circ \overline{F}^{-1} \cup G^* \circ \overline{S} \circ \overline{S}^{-1} \subseteq G^* \circ id_H \cup G^* \circ id_H = G^*_H.
\end{aligned}$$

Then $G^*_N = G_N$, since $G^*_N \subseteq G_N$ and

$$\begin{aligned}
G_N &= id_N \circ \overline{A} \circ G_H \circ \overline{A}^{-1} = id_N \circ \overline{A} \circ G^*_H \circ \overline{A}^{-1} = id_N \circ \overline{A} \circ G^* \circ \overline{A}^{-1} \subseteq id_N \circ G^* \circ \overline{A} \circ \overline{A}^{-1} \\
&\subseteq id_N \circ G^* \circ id_{N \cup \Sigma} = G^*_N.
\end{aligned}$$

Then $G^*_\Sigma = G_\Sigma$, since $G^*_\Sigma \subseteq G_\Sigma$ and

$$\begin{aligned}
G_\Sigma &= \overline{W} \circ G_N \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1} = \overline{W} \circ G^*_N \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G^*_H \circ \overline{A}^{-1} \\
&\subseteq \overline{W} \circ G^* \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G^* \circ \overline{A}^{-1} \subseteq G^* \circ \overline{W} \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ G^* \circ \overline{A} \circ \overline{A}^{-1} \\
&\subseteq G^* \circ id_\Sigma \ \cup \ id_\Sigma \circ G^* \circ id_{N \cup \Sigma} = G^*_\Sigma \ \cup G^*_\Sigma = G^*_\Sigma.
\end{aligned}$$

This completes the proof that $G^* = G$, and so verifies that G is minimal relative to $\mathcal{N}$. ▮

3.4 Semi-definite and definite networks

Definition. A network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is *semi-definite* iff

(1a) $W \circ A \circ F = W \circ A \circ S$,

(3) G_N is acyclic,

(4a) $H = F(E) \cup S(E)$,

(5a) $G_K \circ G_K^{-1} \subseteq id_K$,

(6) G is minimal relative to $\mathcal{N}$,

(7a) $\overline{F} \circ G = G \circ \overline{F}$ and $\overline{S} \circ G = G \circ \overline{S}$,

(8a) $\forall R \ \ (\overline{A}^{-1} \circ R = (\overline{F} \cup \overline{S}) \circ R \ \Rightarrow \ G_\Sigma \circ R \subseteq \overline{W} \circ G_N \circ R \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1} \circ R)$.

Definition. A network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is *definite* iff

(1b) $E \underset{A \circ S}{\overset{A \circ F}{\rightrightarrows}} N \cup \Sigma \xrightarrow{W} \Sigma$ is a coequaliser diagram in the category of sets, (2b) G_N is connected relative to P,

(3) G_N is acyclic,

(4b) $E \overset{F}{\to} H \overset{S}{\leftarrow} E$ is a sum diagram in the category of sets,

(5b) $id_K = G_K \circ G_K^{-1} \cup G_K^{-1} \circ G_K$,

(6) G is minimal relative to $\mathcal{N}$.

The grammar is required to be semi-definite, whereas the pattern is required to be definite.

The clauses of these two definitions may be paraphrased informally as follows.

(1a) The two nodes at the ends of any edge belong to the same whole.

(1b) The nodes and edges belonging to any whole form a *connected* graph.

(2b) Two nodes share the same part only when they are glued together (directly or indirectly). (This condition prevents the same symbol from being interpreted as part of two unrelated wholes.)

(3) There is no cyclic sequence of gluings. This is a technical condition for ensuring that condition (2b) holds in the pattern at the end of recognition (see §3.6).

(4a) Every hook has at least one incident edge.

(4b) Every hook has exactly one incident edge.

(5a) Every facet has at most one superfacet.

(5b) Every facet has exactly one subfacet or superfacet.

(6) Every facet is glued to another facet, but the gluings are minimal subject to this constraint.

(7a) If any hook is a subhook then all its incident edges are subedges.

(8a) Whenever a subsymbol is glued to a supersymbol, there are sufficiently many nodes and hooks belonging to the subsymbol that are glued to nodes and hooks belonging to the supersymbol. This is another technical condition for ensuring that condition (2b) holds in the pattern at the end of recognition (it is used directly in theorem 22 below).

The definiteness conditions subsume the semi-definiteness conditions, as the following theorem shows.

Theorem 18. Any definite network is semi-definite.

Proof. Consider any definite network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$. We shall verify each of the conditions for semi-definiteness in turn (reordering them to leave the harder ones until last).

$\underline{W \circ A \circ F = W \circ A \circ S}$. This follows from the fact that $E \overset{A\circ F}{\underset{A\circ S}{\rightrightarrows}} N \cup \Sigma \overset{W}{\to} \Sigma$ is a coequaliser diagram.

$\underline{G_N \text{ is acyclic.}}$ This is given.

$\underline{H = F(E) \cup S(E).}$ This follows from the fact that $E \overset{F}{\to} H \overset{S}{\leftarrow} E$ is a sum diagram.

$\underline{G_K \circ G_K^{-1} \subseteq id_K}$. This follows from the given condition $id_K = G_K \circ G_K^{-1} \cup G_K^{-1} \circ G_K$.

$\underline{G \text{ is minimal relative to } \mathcal{N}.}$ This is given.

$\underline{\overline{F} \circ G = G \circ \overline{F} \text{ and } S \circ G = G \circ \overline{S}.}$ This follows by

$$
\begin{aligned}
G \circ \overline{F} &= G \circ id_H \circ \overline{F} \\
&= G_H \circ \overline{F} \\
&= \left(\overline{F} \circ G_E \circ \overline{F}^{-1} \ \cup \ \overline{S} \circ G_E \circ \overline{S}^{-1}\right) \circ \overline{F} && \text{by theorem 17 (G is minimal relative to $\mathcal{N}$)} \\
&= \overline{F} \circ G_E \circ \overline{F}^{-1} \circ \overline{F} \ \cup \ \overline{S} \circ G_E \circ \overline{S}^{-1} \circ \overline{F} \\
&= \overline{F} \circ G_E \circ id_E \ \cup \ \overline{S} \circ G_E \circ \bot && \text{by theorem 7 ($E \xrightarrow{F} H \xleftarrow{S} E$ is a sum)} \\
&= \overline{F} \circ G
\end{aligned}
$$

with a similar proof for $\overline{S} \circ G = G \circ \overline{S}$.

$\underline{\forall R \ \ (\overline{A}^{-1} \circ R = (\overline{F} \cup \overline{S}) \circ R \ \Rightarrow \ G_\Sigma \circ R \subseteq \overline{W} \circ G_N \circ R \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1} \circ R).}$ Consider any relation R such that $\overline{A}^{-1} \circ R = (\overline{F} \cup \overline{S}) \circ R$. Then

$$
\begin{aligned}
\overline{F}^{-1} \circ \overline{A}^{-1} \circ R &= \overline{F}^{-1} \circ (\overline{F} \cup \overline{S}) \circ R \\
&= \overline{F}^{-1} \circ \overline{F} \circ R \ \cup \ \overline{F}^{-1} \circ \overline{S} \circ R \\
&= id_E \circ R \ \cup \ \bot \circ R && \text{by theorem 7 ($E \xrightarrow{F} H \xleftarrow{S} E$ is a sum)} \\
&= id_E \circ R
\end{aligned}
$$

and similarly $\overline{S}^{-1} \circ \overline{A}^{-1} \circ R = id_E \circ R$. Consequently $\overline{F}^{-1} \circ \overline{A}^{-1} \circ R = \overline{S}^{-1} \circ \overline{A}^{-1} \circ R$.

Define a function $f{:}\, N \cup \Sigma \to \mathcal{P}(\mathrm{dom}(R))$ (where $\mathcal{P}(\mathrm{dom}(R))$ is the power-set of $\mathrm{dom}(R)$) by

$$\forall x{\in}N \cup \Sigma \quad f(x) = \{\, y \mid R(x,y)\,\}.$$

Then

$$
\begin{aligned}
\forall e{\in}E \quad f(A(F(e))) &= \{\, y \mid R(A(F(e)),y)\,\} = \{\, y \mid (\overline{F}^{-1} \circ \overline{A}^{-1} \circ R)(e,y)\,\} \\
&= \{\, y \mid (\overline{S}^{-1} \circ \overline{A}^{-1} \circ R)(e,y)\,\} = \{\, y \mid R(A(S(e)),y)\,\} = f(A(S(e)))
\end{aligned}
$$

i.e., $f \circ A \circ F = f \circ A \circ S$. Since $E \underset{A \circ S}{\overset{A \circ F}{\rightrightarrows}} N \cup \Sigma \xrightarrow{W} \Sigma$ is a coequaliser diagram, there exists a unique function $i{:}\, \Sigma \to \mathcal{P}(\mathrm{dom}(R))$ such that $i \circ W = f$. Bearing in mind that $\forall \sigma{\in}\Sigma \ W(\sigma) = \sigma$, this implies

$$
\begin{aligned}
\forall n{\in}N \quad \{\, y \mid (\overline{W}^{-1} \circ R)(n,y)\,\} &= \{\, y \mid R(W(n),y)\,\} = f(W(n)) = i(W(W(n))) = i(W(n)) \\
&= f(n) = \{\, y \mid R(n,y)\,\}
\end{aligned}
$$

so $id_N \circ \overline{W}^{-1} \circ R = id_N \circ R$, so $\overline{W} \circ G \circ id_N \circ \overline{W}^{-1} \circ R = \overline{W} \circ G \circ id_N \circ R$, i.e., $\overline{W} \circ G_N \circ \overline{W}^{-1} \circ R = \overline{W} \circ G_N \circ R$.

Also, G is minimal relative to $\mathcal{N}$, and therefore, by theorem 17,

$$G_\Sigma = \overline{W} \circ G_N \circ \overline{W}^{-1} \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1}.$$

Hence

$$G_\Sigma \circ R = \overline{W} \circ G_N \circ \overline{W}^{-1} \circ R \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1} \circ R = \overline{W} \circ G_N \circ R \ \cup \ id_\Sigma \circ \overline{A} \circ G_H \circ \overline{A}^{-1} \circ R$$

which is stronger than the required condition. ▮

3.5 Properties preserved by homomorphisms

Parsing involves constructing a pattern network $\mathcal{N}_1$ and a homomorphism p from $\mathcal{N}_1$ to the grammar network $\mathcal{N}_0$. The fact that $\mathcal{N}_1$ and $\mathcal{N}_0$ are linked by a homomorphism implies some tight relationships between the two networks; these relationships are stated in theorem 19 below. In particular, the fact that $\mathcal{N}_0$ is semi-definite implies that some of the semi-definiteness properties are carried over to $\mathcal{N}_1$, as shown in theorem 20; these properties therefore hold true for the pattern throughout recognition, not just at the end.

Theorem 19. If $p{:}\mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism from a network $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ to a network $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ then

(i) $\overline{p}^{-1} \circ \overline{p} \;\cap\; G_1 \circ G_1^{-1} \subseteq id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1}$,

(ii) $\overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; \overline{A_1}^{-1} \circ (G_{1N} \cup G_{1\Sigma}) \circ \overline{A_1} = G_{1H}$,

(iii) $\overline{p}^{-1} \circ G_{0K} \circ \overline{p} \;\cap\; \overline{C_1}^{-1} \circ G_{1E} \circ \overline{C_1} = G_{1K}$,

(iv) $\overline{A_0}^{-1} \circ \overline{p} = \overline{p} \circ \overline{A_1}^{-1}$,

(v) $\overline{C_0}^{-1} \circ \overline{p} = \overline{p} \circ \overline{C_1}^{-1}$.

Proof. (i) This follows by theorem 14 since G_1 is minimal relative to p (by the definition of a homomorphism).

(ii) I shall use the abbreviations $G_{1N\Sigma} = G_{1N} \cup G_{1\Sigma}$ and $p_H = p|_{H_1}$ (so $\overline{p_H} = \overline{p} \circ id_{H_1} = id_{H_0} \circ \overline{p}$).

$\underline{G_{1H} \subseteq \overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ \overline{A_1}}$. The definition of a homomorphism gives $\overline{p} \circ G_1 = G_0 \circ \overline{p}$ and hence $\overline{p} \circ G_{1H} = G_{0H} \circ \overline{p}$ (in detail, $\overline{p} \circ G_{1H} = \overline{p} \circ id_{H_1} \circ G_1 = id_{H_0} \circ \overline{p} \circ G_1 = id_{H_0} \circ G_0 \circ \overline{p} = G_{0H} \circ \overline{p}$). Hence by theorem 2(iv) $G_{1H} \subseteq \overline{p}^{-1} \circ G_{0H} \circ \overline{p}$. Also, by the definition of a network, $\overline{A_1} \circ G_1 \subseteq G_1 \circ \overline{A_1}$, hence by theorem 2(iv) $id_{H_1} \circ G_1 \subseteq \overline{A_1}^{-1} \circ G_1 \circ \overline{A_1}$, i.e., $G_{1H} \subseteq \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ \overline{A_1}$. Combining these two gives $G_{1H} \subseteq \overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ \overline{A_1}$.

$\underline{\overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ \overline{A_1} \subseteq G_{1H}}$. Now,

$$
\begin{aligned}
\overline{p}^{-1} \circ G_{0H} \circ \overline{p} &= \overline{p}^{-1} \circ id_{H_0} \circ G_0 \circ \overline{p} \\
&= \overline{p_H}^{-1} \circ id_{H_0} \circ \overline{p} \circ G_1 \qquad \text{since } G_0 \circ \overline{p} = \overline{p} \circ G_1 \ (p \text{ is a homomorphism}) \\
&= \overline{p_H}^{-1} \circ \overline{p_H} \circ id_{H_1} \circ G_1 \\
&= \overline{p_H}^{-1} \circ \overline{p_H} \circ G_{1H}.
\end{aligned} \tag{1}
$$

Also, from the definition of a network, $\overline{A_1} \circ G_1 \subseteq G_1 \circ \overline{A_1}$, so by theorem 2(iii),(iv) $id_{H_1} \circ G_1 \circ \overline{A_1}^{-1} \subseteq \overline{A_1}^{-1} \circ G_1 \circ id_{N_1 \cup \Sigma_1}$, i.e., $G_{1H} \circ \overline{A_1}^{-1} \subseteq \overline{A_1}^{-1} \circ G_{1N\Sigma}$. Inverting this gives

$$\overline{A_1} \circ G_{1H}^{-1} \subseteq G_{1N\Sigma}^{-1} \circ \overline{A_1}. \tag{2}$$

Furthermore,

$$
\begin{aligned}
\overline{p_H}^{-1} \circ \overline{p_H} &= \overline{p}^{-1} \circ id_{H_0} \circ \overline{p} \\
&\subseteq \overline{p}^{-1} \circ \overline{A_0}^{-1} \circ \overline{A_0} \circ \overline{p} \qquad \text{by theorem 2(ii)} \\
&= \overline{A_1}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \circ \overline{A_1} \qquad \text{since } A_0 \circ p = p \circ A_1 \ (p \text{ is a homomorphism}).
\end{aligned} \tag{3}
$$

Using (1)–(3),

$$
\begin{aligned}
&\overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ \overline{A_1} \\
&\quad = \overline{p_H}^{-1} \circ \overline{p_H} \circ G_{1H} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ \overline{A_1} && \text{by (1)} \\
&\quad \subseteq (\overline{p_H}^{-1} \circ \overline{p_H} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ \overline{A_1} \circ G_{1H}^{-1}) \circ G_{1H} && \text{by theorem 4(i)} \\
&\quad \subseteq (\overline{p_H}^{-1} \circ \overline{p_H} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ G_{1N\Sigma}^{-1} \circ \overline{A_1}) \circ G_{1H} && \text{by (2)} \\
&\quad = (\overline{p_H}^{-1} \circ \overline{p_H} \;\cap\; \overline{A_1}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \circ \overline{A_1} \;\cap\; \overline{A_1}^{-1} \circ G_{1N\Sigma} \circ G_{1N\Sigma}^{-1} \circ \overline{A_1}) \circ G_{1H} && \text{by (3)} \\
&\quad = \left(\overline{p_H}^{-1} \circ \overline{p_H} \;\cap\; \overline{A_1}^{-1} \circ (\overline{p}^{-1} \circ \overline{p} \;\cap\; G_{1N\Sigma} \circ G_{1N\Sigma}^{-1}) \circ \overline{A_1}\right) \circ G_{1H} && \text{by theorem 3(v),(vi)} \\
&\quad \subseteq \left(\overline{p_H}^{-1} \circ \overline{p_H} \;\cap\; \overline{A_1}^{-1} \circ (\overline{p}^{-1} \circ \overline{p} \;\cap\; G_1 \circ G_1^{-1}) \circ \overline{A_1}\right) \circ G_{1H} \\
&\quad \subseteq (\overline{p_H}^{-1} \circ \overline{p_H} \;\cap\; \overline{A_1}^{-1} \circ id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1} \circ \overline{A_1}) \circ G_{1H} && \text{by part (i)} \\
&\quad = (\overline{p_H}^{-1} \circ \overline{p_H} \;\cap\; \overline{A_1}^{-1} \circ \overline{A_1}) \circ G_{1H} \\
&\quad \subseteq id_{H_1} \circ G_{1H} && \text{by theorem 6 (p is a homomorphism)} \\
&\quad = G_{1H}
\end{aligned}
$$

as required.

(iii) The argument is as in part (ii), with hooks replaced by facets, $G_{1N\Sigma}$ replaced by G_{1E}, and $id_{N_1 \cup \Sigma_1}$ replaced by id_{E_1}.

(iv) We have

$$
\begin{aligned}
\overline{A_0}^{-1} \circ \overline{p} &= \overline{A_0}^{-1} \circ id_{N_0 \cup \Sigma_0} \circ \overline{p} && \text{since } A_0 \colon H_0 \to N_0 \cup \Sigma_0 \\
&= \overline{A_0}^{-1} \circ \overline{p} \circ id_{N_1 \cup \Sigma_1} && \text{since } N_1 \cup \Sigma_1 = p^{-1}(N_0 \cup \Sigma_0) \\
&= \overline{A_0}^{-1} \circ \overline{p|_{N_1 \cup \Sigma_1}} \\
&= \overline{p|_{H_1}} \circ \overline{A_1}^{-1} && \text{by theorem 6 (p is a homomorphism)} \\
&= \overline{p} \circ id_{H_1} \circ \overline{A_1}^{-1} \\
&= \overline{p} \circ \overline{A_1}^{-1} && \text{since } A_1 \colon H_1 \to N_1 \cup \Sigma_1.
\end{aligned}
$$

(v) The argument is as in part (iv). ∎

Theorem 20. If $p\colon \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism from a network $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ to a *semi-definite* network $\mathcal{N}_0$, then

(i) $G_{1K} \circ G_{1K}^{-1} \subseteq id_{K_1}$,

(ii) $\overline{F_1} \circ G_1 = G_1 \circ \overline{F_1}$ and $\overline{S_1} \circ G_1 = G_1 \circ \overline{S_1}$,

(iii) $G_{1N} = id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}$,

(iv) $G_{1E} = \overline{C_1} \circ G_{1K} \circ \overline{C_1}^{-1}$.

Proof. We shall use the usual notation, $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$.

(i) Now,

$$\begin{aligned}
G_{1K} \circ G_{1K}^{-1} &= id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1} \circ G_1 \circ id_{K_1} \circ G_1^{-1} \circ id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1} \\
&\subseteq \overline{p}^{-1} \circ \overline{p} \circ G_1 \circ id_{K_1} \circ G_1^{-1} \circ \overline{p}^{-1} \circ \overline{p} && \text{by theorem 2(ii)} \\
&= \overline{p}^{-1} \circ G_0 \circ \overline{p} \circ id_{K_1} \circ \overline{p}^{-1} \circ G_0^{-1} \circ \overline{p} && \text{since } \overline{p} \circ G_1 = G_0 \circ \overline{p} \text{ (p is a homomorphism)} \\
&= \overline{p}^{-1} \circ G_0 \circ id_{K_0} \circ \overline{p} \circ \overline{p}^{-1} \circ G_0^{-1} \circ \overline{p} && \text{since } K_1 = p^{-1}(K_0) \\
&\subseteq \overline{p}^{-1} \circ G_0 \circ id_{K_0} \circ id_{\Sigma_0 \cup N_0 \cup H_0 \cup E_0 \cup K_0} \circ G_0^{-1} \circ \overline{p} && \text{by theorem 2(i)} \\
&= \overline{p}^{-1} \circ G_{0K} \circ G_{0K}^{-1} \circ \overline{p} \\
&\subseteq \overline{p}^{-1} \circ \overline{p} && \text{since } G_{0K} \circ G_{0K}^{-1} \subseteq id_{K_0} \text{ ($\mathcal{N}_0$ is semi-definite)}
\end{aligned}$$

so

$$\begin{aligned}
G_{1K} \circ G_{1K}^{-1} &= \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1K} \circ G_{1K}^{-1} \\
&= id_{K_1} \circ \left(\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ G_1^{-1}\right) \circ id_{K_1} && \text{by theorem 4(iii),(iv)} \\
&\subseteq id_{K_1} \circ id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1} \circ id_{K_1} && \text{by theorem 19(i)} \\
&= id_{K_1}
\end{aligned}$$

as required.

(ii) I shall just show $\overline{F_1} \circ G_1 = G_1 \circ \overline{F_1}$; the proof for $\overline{S_1} \circ G_1 = G_1 \circ \overline{S_1}$ is similar. First,

$$\begin{aligned}
G_1 \circ \overline{F_1} &= id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1} \circ G_1 \circ \overline{F_1} \\
&\subseteq \overline{p}^{-1} \circ \overline{p} \circ G_1 \circ \overline{F_1} && \text{by theorem 2(ii)} \\
&= \overline{p}^{-1} \circ G_0 \circ \overline{p} \circ \overline{F_1} && \text{since } \overline{p} \circ G_1 = G_0 \circ \overline{p} \text{ (p is a homomorphism)} \\
&= \overline{p}^{-1} \circ G_0 \circ \overline{F_0} \circ \overline{p} && \text{since } p \circ F_1 = F_0 \circ p \text{ (p is a homomorphism)} \\
&= \overline{p}^{-1} \circ \overline{F_0} \circ G_0 \circ \overline{p} && \text{since } G_0 \circ \overline{F_0} = \overline{F_0} \circ G_0 \text{ ($\mathcal{N}_0$ is semi-definite)} \\
&= \overline{p}^{-1} \circ \overline{F_0} \circ \overline{p} \circ G_1 && \text{since } G_0 \circ \overline{p} = \overline{p} \circ G_1 \text{ (p is a homomorphism)} \\
&= \overline{p}^{-1} \circ \overline{p} \circ \overline{F_1} \circ G_1 && \text{since } F_0 \circ p = p \circ F_1 \text{ (p is a homomorphism).}
\end{aligned}$$

Hence

$$\begin{aligned}
G_1 \circ \overline{F_1} &= \overline{p}^{-1} \circ \overline{p} \circ \overline{F_1} \circ G_1 \ \cap \ G_1 \circ \overline{F_1} \\
&\subseteq (\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ \overline{F_1} \circ (\overline{F_1} \circ G_1)^{-1}) \circ \overline{F_1} \circ G_1 && \text{by theorem 4(i)} \\
&\subseteq (\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ \overline{F_1} \circ (G_1 \circ \overline{F_1})^{-1}) \circ \overline{F_1} \circ G_1 && \text{since } \overline{F_1} \circ G_1 \subseteq G_1 \circ \overline{F_1} \text{ ($\mathcal{N}_1$ is a network)} \\
&= (\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ \overline{F_1} \circ \overline{F_1}^{-1} \circ G_1^{-1}) \circ \overline{F_1} \circ G_1 \\
&\subseteq (\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ id_{H_1} \circ G_1^{-1}) \circ \overline{F_1} \circ G_1 && \text{by theorem 2(i)} \\
&\subseteq (\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ G_1^{-1}) \circ \overline{F_1} \circ G_1 \\
&\subseteq id_{\Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1} \circ \overline{F_1} \circ G_1 && \text{by theorem 19(i)} \\
&= \overline{F_1} \circ G_1.
\end{aligned}$$

Since the converse $\overline{F_1} \circ G_1 \subseteq G_1 \circ \overline{F_1}$ is given by the definition of a network, this establishes that $\overline{F_1} \circ G_1 = G_1 \circ \overline{F_1}$, as required.

(iii) The definition of a network gives $\overline{A_1} \circ G_1 \subseteq G_1 \circ \overline{A_1}$, and hence by theorem 2(iii) $\overline{A_1} \circ G_1 \circ \overline{A_1}^{-1} \subseteq G_1 \circ id_{N_1 \cup \Sigma_1}$, and hence $id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \subseteq G_{1N}$.

For the converse, the fact that p is a homomorphism gives $\overline{p} \circ G_1 = G_0 \circ \overline{p}$, and hence $\overline{p} \circ G_{1N} = G_{0N} \circ \overline{p}$. Thus

$$\begin{aligned} G_{1N} &\subseteq \overline{p}^{-1} \circ G_{0N} \circ \overline{p} && \text{by theorem 2(iv)} \\ &= \overline{p}^{-1} \circ id_{N_0} \circ \overline{A_0} \circ G_{0H} \circ \overline{A_0}^{-1} \circ \overline{p} && \text{by theorem 17 ($\mathcal{N}_0$ is semi-definite)} \\ &\subseteq \overline{p}^{-1} \circ \overline{A_0} \circ G_{0H} \circ \overline{A_0}^{-1} \circ \overline{p} \\ &= \overline{A_1} \circ \overline{p}^{-1} \circ G_{0H} \circ \overline{p} \circ \overline{A_1}^{-1} && \text{by theorem 19(iv).} \end{aligned}$$

Hence

$$\begin{aligned} G_{1N} &= \overline{A_1} \circ \overline{p}^{-1} \circ G_{0H} \circ \overline{p} \circ \overline{A_1}^{-1} \;\cap\; G_{1N} \\ &\subseteq \left(\overline{A_1} \circ \overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; G_{1N} \circ \overline{A_1}\right) \circ \overline{A_1}^{-1} && \text{by theorem 4(i)} \\ &\subseteq \overline{A_1} \circ \left(\overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; \overline{A_1}^{-1} \circ G_{1N} \circ \overline{A_1}\right) \circ \overline{A_1}^{-1} && \text{by theorem 4(ii)} \\ &\subseteq \overline{A_1} \circ \left(\overline{p}^{-1} \circ G_{0H} \circ \overline{p} \;\cap\; \overline{A_1}^{-1} \circ (G_{1N} \cup G_{1\Sigma}) \circ \overline{A_1}\right) \circ \overline{A_1}^{-1} \\ &= \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} && \text{by theorem 19(ii)} \end{aligned}$$

so $G_{1N} = id_{N_1} \circ G_{1N} \subseteq id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}$, as required.

(iv) The argument is similar to part (iii), with hooks replaced by facets, nodes replaced by edges, A_1, A_0 replaced by C_1, C_0, and with id_{N_1}, id_{N_0} and $G_{1\Sigma}$ removed. ∎

3.6 Theorems on definiteness

The recognition process aims to achieve a definite pattern $\mathcal{N}_1$. Theorem 23 below provides sufficient conditions through which definiteness is ensured. Theorem 21 and theorem 22 are more general results, with weaker hypotheses and conclusions, which are used in the proof of theorem 23 and also later in §8.

Theorem 21. If $p: \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism from a network $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ to a semi-definite network $\mathcal{N}_0$, and Y is a set, such that

(a) $Y \subseteq \Sigma_1$ and $E_1 \overset{F_1}{\to} {A_1}^{-1}(N_1 \cup Y) \overset{S_1}{\leftarrow} E_1$ is a sum diagram in the category of sets,
(b) $\exists f: K_1 \backslash dom(G_1) \to K_1 \; G_{1K}^{-1} = \overline{f}$,

then

(i) $G_1 \circ id_{{A_1}^{-1}(N_1 \cup Y)} = \overline{F_1} \circ G_{1E} \circ \overline{F_1}^{-1} \cup \overline{S_1} \circ G_{1E} \circ \overline{S_1}^{-1}$,
(ii) $id_{K_1} = G_{1K} \circ G_{1K}^{-1} \;\cup\; G_{1K}^{-1} \circ G_{1K}$,
(iii) $\overline{p}^{-1} \circ \overline{p} \;\cap\; G_{1E}^{-1} \circ G_{1E} \subseteq id_{E_1}$,
(iv) $\overline{p}^{-1} \circ \overline{p} \;\cap\; id_{{A_1}^{-1}(N_1)} \circ G_1^{-1} \circ G_1 \circ id_{{A_1}^{-1}(N_1)} \subseteq id_{{A_1}^{-1}(N_1)}$,
(v) $\overline{p}^{-1} \circ \overline{p} \;\cap\; G_{1N}^{-1} \circ G_{1N} \subseteq id_{N_1}$,
(vi) G_{1N} is acyclic.

Proof. We shall use the usual notation, $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$.

(i) We have

$$\begin{aligned} G_1 \circ id_{{A_1}^{-1}(N_1 \cup Y)} &= G_1 \circ \left(\overline{F_1} \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ \overline{S_1}^{-1}\right) && \text{by theorem 7 and hypothesis (a)} \\ &= G_1 \circ \overline{F_1} \circ \overline{F_1}^{-1} \;\cup\; G_1 \circ \overline{S_1} \circ \overline{S_1}^{-1} && \text{by theorem 3(ii)} \\ &= \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ G_1 \circ \overline{S_1}^{-1} && \text{by theorem 20(ii)} \\ &= \overline{F_1} \circ G_{1E} \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ G_{1E} \circ \overline{S_1}^{-1} && \text{since } F_1, S_1: E_1 \to H_1. \end{aligned}$$

(ii) By hypothesis (b) there is a function $f: K_1 \setminus \mathrm{dom}(G_1) \to K_1$ such that $G_{1K}^{-1} = \overline{f}$. Hence

$$G_{1K}^{-1} \circ G_{1K} \subseteq id_{K_1},$$
$$id_{K_1 \setminus \mathrm{dom}(G_1)} \subseteq G_{1K} \circ G_{1K}^{-1}$$

by theorem 2(i),(ii). But we also have

$$id_{\mathrm{dom}(G_{1K})} \subseteq G_{1K}^{-1} \circ G_{1K} \quad \text{since this holds for all relations,}$$
$$G_{1K} \circ G_{1K}^{-1} \subseteq id_{K_1} \quad \text{by theorem 20(i).}$$

Combining these four in pairs gives

$$G_{1K} \circ G_{1K}^{-1} \;\cup\; G_{1K}^{-1} \circ G_{1K} \subseteq id_{K_1},$$
$$id_{K_1} = id_{K_1 \setminus \mathrm{dom}(G_1)} \cup id_{\mathrm{dom}(G_{1K})} \subseteq G_{1K} \circ G_{1K}^{-1} \;\cup\; G_{1K}^{-1} \circ G_{1K}$$

and hence $id_{K_1} = G_{1K} \circ G_{1K}^{-1} \;\cup\; G_{1K}^{-1} \circ G_{1K}$, as required.

(iii) Theorem 20(iv) says $G_{1E} = \overline{C_1} \circ G_{1K} \circ \overline{C_1}^{-1}$, and hence

$$G_{1E}^{-1} = \overline{C_1} \circ G_{1K}^{-1} \circ \overline{C_1}^{-1}. \tag{1}$$

Also, from the definition of a homomorphism, $\overline{p} \circ G_{1K} = G_{0K} \circ \overline{p}$, and hence by theorem 2(iii),(iv),

$$G_{1K} \circ \overline{p}^{-1} \subseteq \overline{p}^{-1} \circ G_{0K}. \tag{2}$$

This gives

$$\begin{aligned}
&\overline{p}^{-1} \circ \overline{p} \;\cap\; G_{1E}^{-1} \circ G_{1E} \\
&\quad = \overline{p}^{-1} \circ \overline{p} \;\cap\; \overline{C_1} \circ G_{1K}^{-1} \circ \overline{C_1}^{-1} \circ G_{1E} && \text{by (1)} \\
&\quad \subseteq \overline{C_1} \circ \left(\overline{C_1}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \;\cap\; G_{1K}^{-1} \circ \overline{C_1}^{-1} \circ G_{1E}\right) && \text{by theorem 4(ii)} \\
&\quad = \overline{C_1} \circ \left(\overline{p}^{-1} \circ \overline{C_0}^{-1} \circ \overline{p} \;\cap\; G_{1K}^{-1} \circ \overline{C_1}^{-1} \circ G_{1E}\right) && \text{since } p \circ C_1 = C_0 \circ p \text{ (p is a homomorphism)} \\
&\quad = \overline{C_1} \circ \left(\overline{p}^{-1} \circ \overline{p} \circ \overline{C_1}^{-1} \;\cap\; G_{1K}^{-1} \circ \overline{C_1}^{-1} \circ G_{1E}\right) && \text{by theorem 19(v)} \\
&\quad \subseteq \overline{C_1} \circ \left(\overline{p}^{-1} \circ \overline{p} \;\cap\; G_{1K}^{-1} \circ \overline{C_1}^{-1} \circ G_{1E} \circ \overline{C_1}\right) \circ \overline{C_1}^{-1} && \text{by theorem 4(i)} \\
&\quad \subseteq \overline{C_1} \circ G_{1K}^{-1} \circ \left(G_{1K} \circ \overline{p}^{-1} \circ \overline{p} \;\cap\; \overline{C_1}^{-1} \circ G_{1E} \circ \overline{C_1}\right) \circ \overline{C_1}^{-1} && \text{by theorem 4(ii)} \\
&\quad \subseteq \overline{C_1} \circ G_{1K}^{-1} \circ \left(\overline{p}^{-1} \circ G_{0K} \circ \overline{p} \;\cap\; \overline{C_1}^{-1} \circ G_{1E} \circ \overline{C_1}\right) \circ \overline{C_1}^{-1} && \text{by (2)} \\
&\quad = \overline{C_1} \circ G_{1K}^{-1} \circ G_{1K} \circ \overline{C_1}^{-1} && \text{by theorem 19(iii)} \\
&\quad \subseteq \overline{C_1} \circ id_{K_1} \circ \overline{C_1}^{-1} && \text{by part (ii)} \\
&\quad = \overline{C_1} \circ \overline{C_1}^{-1} && \text{since } C_1: K_1 \to E_1 \\
&\quad \subseteq id_{E_1} && \text{by theorem 2(i).}
\end{aligned}$$

(iv) First, the definition of a homomorphism gives $\overline{p} \circ G_{1E} = G_{0E} \circ \overline{p}$, and so by theorem 2(iii),(iv)

$$G_{1E} \circ \overline{p}^{-1} \subseteq \overline{p}^{-1} \circ G_{0E}. \tag{3}$$

This is used in the following:

$$
\begin{aligned}
&\overline{F_1}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \circ \overline{F_1} \;\cap\; G_{1E}^{-1} \circ G_{1E} \\
&\quad = \overline{p}^{-1} \circ \overline{F_0}^{-1} \circ \overline{F_0} \circ \overline{p} \;\cap\; G_{1E}^{-1} \circ G_{1E} \quad \text{since } p \circ F_1 = F_0 \circ p \text{ (}p\text{ is a homomorphism)} \\
&\quad \subseteq \overline{p}^{-1} \circ \left(\overline{F_0}^{-1} \circ \overline{F_0} \;\cap\; \overline{p} \circ G_{1E}^{-1} \circ G_{1E} \circ \overline{p}^{-1}\right) \circ \overline{p} && \text{by theorem 4(i),(ii)} \\
&\quad \subseteq \overline{p}^{-1} \circ \left(\overline{F_0}^{-1} \circ \overline{F_0} \;\cap\; G_{0E}^{-1} \circ \overline{p} \circ \overline{p}^{-1} \circ G_{0E}\right) \circ \overline{p} && \text{by (3)} \\
&\quad \subseteq \overline{p}^{-1} \circ \left(\overline{F_0}^{-1} \circ \overline{F_0} \;\cap\; G_{0E}^{-1} \circ id_{\Sigma_0 \cup N_0 \cup H_0 \cup E_0 \cup K_0} \circ G_{0E}\right) \circ \overline{p} && \text{by theorem 2(i)} \\
&\quad = \overline{p}^{-1} \circ \left(\overline{F_0}^{-1} \circ \overline{F_0} \;\cap\; G_{0E}^{-1} \circ G_{0E}\right) \circ \overline{p} \\
&\quad \subseteq \overline{p}^{-1} \circ id_{E_0} \circ \overline{p} \qquad\qquad \text{by theorem 14 (}G_{0E}^{-1}\text{ is minimal relative to }F_0\text{)} \\
&\quad \subseteq \overline{p}^{-1} \circ \overline{p}. && (4)
\end{aligned}
$$

Using this we derive

$$
\begin{aligned}
&\overline{p}^{-1} \circ \overline{p} \;\cap\; \overline{F_1} \circ G_{1E}^{-1} \circ G_{1E} \circ \overline{F_1}^{-1} \\
&\quad \subseteq \overline{F_1} \circ \left(\overline{F_1}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \circ \overline{F_1} \;\cap\; G_{1E}^{-1} \circ G_{1E}\right) \circ \overline{F_1}^{-1} && \text{by theorem 4(i),(ii)} \\
&\quad = \overline{F_1} \circ \left(\overline{F_1}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \circ \overline{F_1} \;\cap\; G_{1E}^{-1} \circ G_{1E} \;\cap\; G_{1E}^{-1} \circ G_{1E}\right) \circ \overline{F_1}^{-1} \\
&\quad \subseteq \overline{F_1} \circ (\overline{p}^{-1} \circ \overline{p} \;\cap\; G_{1E}^{-1} \circ G_{1E}) \circ \overline{F_1}^{-1} && \text{by (4)} \\
&\quad \subseteq \overline{F_1} \circ id_{E_1} \circ \overline{F_1}^{-1} && \text{by part (iii)} \\
&\quad = \overline{F_1} \circ \overline{F_1}^{-1} && \text{since } F_1 \colon E_1 \to H_1 \\
&\quad \subseteq id_{H_1} && \text{by theorem 2(i).} \quad (5)
\end{aligned}
$$

By a similar argument,

$$
\overline{p}^{-1} \circ \overline{p} \;\cap\; \overline{S_1} \circ G_{1E}^{-1} \circ G_{1E} \circ \overline{S_1}^{-1} \subseteq id_{H_1}. \tag{6}
$$

We also have, using the abbreviation $I = id_{A_1{}^{-1}(N_1)}$,

$$
\begin{aligned}
I \circ G_1^{-1} \circ G_1 \circ I &\subseteq id_{A_1{}^{-1}(N_1 \cup Y)} \circ G_1^{-1} \circ G_1 \circ id_{A_1{}^{-1}(N_1 \cup Y)} \\
&= \left(\overline{F_1} \circ G_{1E}^{-1} \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ G_{1E}^{-1} \circ \overline{S_1}^{-1}\right) \circ \left(\overline{F_1} \circ G_{1E} \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ G_{1E} \circ \overline{S_1}^{-1}\right) \\
&\qquad\qquad \text{by part (i)} \\
&= \overline{F_1} \circ G_{1E}^{-1} \circ id_{E_1} \circ G_{1E} \circ \overline{F_1}^{-1} \;\cup\; \perp \;\cup\; \perp \;\cup\; \overline{S_1} \circ G_{1E}^{-1} \circ id_{E_1} \circ G_{1E} \circ \overline{S_1}^{-1} \\
&\qquad\qquad \text{by theorem 7 and hypothesis (a)} \\
&= \overline{F_1} \circ G_{1E}^{-1} \circ G_{1E} \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ G_{1E}^{-1} \circ G_{1E} \circ \overline{S_1}^{-1}. && (7)
\end{aligned}
$$

Then, by (5), (6) and (7),

$$
\overline{p}^{-1} \circ \overline{p} \;\cap\; I \circ G_1^{-1} \circ G_1 \circ I \subseteq id_{H_1} \tag{8}
$$

and so

$$
\begin{aligned}
\overline{p}^{-1} \circ \overline{p} \;\cap\; I \circ G_1^{-1} \circ G_1 \circ I &= \overline{p}^{-1} \circ \overline{p} \;\cap\; I \circ G_1^{-1} \circ G_1 \circ I \circ I \\
&= \left(\overline{p}^{-1} \circ \overline{p} \;\cap\; I \circ G_1^{-1} \circ G_1 \circ I\right) \circ I && \text{by theorem 4(iii)} \\
&\subseteq id_{H_1} \circ I && \text{by (8)} \\
&= I && \text{since } A_1{}^{-1}(N_1) \subseteq H_1
\end{aligned}
$$

as required.

(v) From the definition of a homomorphism, $\overline{p} \circ G_{1H} = G_{0H} \circ \overline{p}$, and hence by theorem 2(iii),(iv),

$$G_{1H} \circ \overline{p}^{-1} \subseteq \overline{p}^{-1} \circ G_{0H}. \tag{9}$$

Also (with $I = id_{A_1{}^{-1}(N_1)}$ again),

$$\begin{aligned}
I \circ G_1 &= I \circ id_{H_1} \circ G_1 && \text{since } A_1{}^{-1}(N_1) \subseteq H_1 \\
&\subseteq I \circ \overline{A_1}^{-1} \circ \overline{A_1} \circ G_1 && \text{by theorem 2(ii)} \\
&\subseteq I \circ \overline{A_1}^{-1} \circ G_1 \circ \overline{A_1} && \text{since } \overline{A_1} \circ G_1 \subseteq G_1 \circ \overline{A_1} \text{ } (\mathcal{N}_1 \text{ is a network}) \\
&= \overline{A_1}^{-1} \circ id_{N_1} \circ G_1 \circ \overline{A_1} && \text{since } I = id_{A_1{}^{-1}(N_1)} \\
&= \overline{A_1}^{-1} \circ G_1 \circ id_{N_1} \circ \overline{A_1} && \\
&= \overline{A_1}^{-1} \circ G_1 \circ \overline{A_1} \circ I && \text{since } I = id_{A_1{}^{-1}(N_1)}
\end{aligned}$$

so by theorem 4(v)

$$I \circ G_1 \subseteq G_1 \circ I$$

and inverting this gives

$$G_1^{-1} \circ I \subseteq I \circ G_1^{-1}. \tag{10}$$

Also, theorem 20(iii) says $G_{1N} = id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}$, and hence

$$G_{1N}^{-1} = \overline{A_1} \circ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ id_{N_1}. \tag{11}$$

Then,

$$\begin{aligned}
&\overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ G_{1N} \circ \overline{A_1} \\
&\quad = \overline{p}^{-1} \circ \overline{p} \ \cap \ \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ id_{N_1} \circ G_{1N} \circ id_{N_1} \circ \overline{A_1} \\
&\quad = \overline{p}^{-1} \circ \overline{p} \ \cap \ \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ I \circ \overline{A_1}^{-1} \circ G_{1N} \circ \overline{A_1} \circ I && \text{since } I = id_{A_1{}^{-1}(N_1)} \\
&\quad \subseteq \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ \left(G_{1H} \circ \overline{p}^{-1} \circ \overline{p} \ \cap \ I \circ \overline{A_1}^{-1} \circ G_{1N} \circ \overline{A_1} \circ I\right) && \text{by theorem 4(ii)} \\
&\quad = \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ I \circ \left(G_{1H} \circ \overline{p}^{-1} \circ \overline{p} \ \cap \ \overline{A_1}^{-1} \circ G_{1N} \circ \overline{A_1}\right) \circ I && \text{by theorem 4(iii),(iv)} \\
&\quad \subseteq \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ I \circ \left(\overline{p}^{-1} \circ G_{0H} \circ \overline{p} \ \cap \ \overline{A_1}^{-1} \circ G_{1N} \circ \overline{A_1}\right) \circ I && \text{by (9)} \\
&\quad \subseteq \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ I \circ G_{1H} \circ I && \text{by theorem 19(ii)} \\
&\quad = \overline{p}^{-1} \circ \overline{p} \ \cap \ G_1^{-1} \circ I \circ G_1 \circ I && \text{since } id_{H_1} \circ I = I \\
&\quad \subseteq \overline{p}^{-1} \circ \overline{p} \ \cap \ I \circ G_1^{-1} \circ G_1 \circ I && \text{by (10)} \\
&\quad \subseteq I && \text{by part (iv).}
\end{aligned} \tag{12}$$

This gives

$$
\begin{aligned}
&\overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1N}^{-1} \circ G_{1N} \\
&\quad = \overline{p}^{-1} \circ \overline{p} \ \cap \ \overline{A_1} \circ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ id_{N_1} \circ G_{1N} && \text{by (11)} \\
&\quad = \overline{p}^{-1} \circ \overline{p} \ \cap \ \overline{A_1} \circ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ G_{1N} \\
&\quad \subseteq \overline{A_1} \circ \big(\overline{A_1}^{-1} \circ \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ G_{1N}\big) && \text{by theorem 4(ii)} \\
&\quad = \overline{A_1} \circ \big(\overline{p}^{-1} \circ \overline{A_0}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ G_{1N}\big) && \text{since } p \circ A_1 = A_0 \circ p \ (p \text{ is a homomorphism}) \\
&\quad = \overline{A_1} \circ \big(\overline{p}^{-1} \circ \overline{p} \circ \overline{A_1}^{-1} \ \cap \ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ G_{1N}\big) && \text{by theorem 19(iv)} \\
&\quad \subseteq \overline{A_1} \circ \big(\overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1H}^{-1} \circ \overline{A_1}^{-1} \circ G_{1N} \circ \overline{A_1}\big) \circ \overline{A_1}^{-1} && \text{by theorem 4(i)} \\
&\quad \subseteq \overline{A_1} \circ I \circ \overline{A_1}^{-1} && \text{by (12)} \\
&\quad = id_{N_1} \circ \overline{A_1} \circ \overline{A_1}^{-1} && \text{since } I = id_{A_1{}^{-1}(N_1)} \\
&\quad \subseteq id_{N_1} \circ id_{N_1 \cup \Sigma_1} && \text{by theorem 2(i)} \\
&\quad = id_{N_1}
\end{aligned}
$$

as required.

(vi) Define $p_N = p|_{N_1}$. We already know the following.

$$
\begin{aligned}
&\overline{p_N} \circ G_{1N} = G_{0N} \circ \overline{p_N} && \text{since } p \text{ is a homomorphism;} \\
&\overline{p_N}^{-1} \circ \overline{p_N} \ \cap \ G_{1N}^{-1} \circ G_{1N} \subseteq \overline{p}^{-1} \circ \overline{p} \ \cap \ G_{1N}^{-1} \circ G_{1N} \\
&\qquad \subseteq id_{N_1} && \text{by part (v);} \\
&\overline{p_N}^{-1} \circ \overline{p_N} \ \cap \ G_{1N} \circ G_{1N}^{-1} = id_{N_1} \circ \big(\overline{p}^{-1} \circ \overline{p} \ \cap \ G_1 \circ G_1^{-1}\big) \circ id_{N_1} && \text{by theorem 4(iii),(iv)} \\
&\qquad \subseteq id_{N_1} && \text{by theorem 19(i);} \\
&G_{0N} \text{ is acyclic} && \text{since } \mathcal{N}_0 \text{ is semi-definite.}
\end{aligned}
$$

Then by theorem 12 G_{1N} is also acyclic. ∎

Theorem 22. If $p: \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism from a network $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ to a semi-definite network $\mathcal{N}_0$, and Y is a set, such that

(a) $W_1(N_1) \subseteq Y \subseteq \Sigma_1$ and $E_1 \xrightarrow{F_1} A_1{}^{-1}(N_1 \cup Y) \xleftarrow{S_1} E_1$ is a sum diagram in the category of sets,

(b) $W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1$,

then

(i) $id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} = \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_{\Sigma_1}$,

(ii) $G_1 \circ id_Y = \big(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup \ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}\big) \circ id_Y$,

(iii) $G_1 \circ id_Y \subseteq id_Y \circ G_1$.

Proof. We shall use the usual notation, $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$. We begin with two observations.

First, note that $W_1(N_1) \subseteq Y$ and $W_1(Y) = Y$, so $N_1 \cup Y \subseteq W_1{}^{-1}(Y)$. Conversely, any element of $W_1{}^{-1}(Y)$ is either a node or a symbol in Y, so $W_1{}^{-1}(Y) \subseteq N_1 \cup Y$. Hence

$$W_1{}^{-1}(Y) = N_1 \cup Y. \tag{1}$$

Secondly, from $\overline{A_1} \circ G_1 \subseteq G_1 \circ \overline{A_1}$, theorem 2(iii) implies $\overline{A_1} \circ G_1 \circ \overline{A_1}^{-1} \subseteq G_1 \circ id_{N_1 \cup \Sigma_1}$, and hence

$$id_{\Sigma_1} \circ \overline{A_1} \circ G_1 \circ \overline{A_1}^{-1} \subseteq id_{\Sigma_1} \circ G_1 \circ id_{N_1 \cup \Sigma}.$$

which may be rewritten as

$$id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \subseteq G_1 \circ id_{\Sigma_1}. \tag{2}$$

(i) Applying theorem 4(v) to (2),

$$id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \subseteq \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_{\Sigma_1}.$$

The converse is proved similarly, and hence

$$id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} = \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_{\Sigma_1}$$

as required.

(ii) From $\overline{W_1} \circ G_1 = G_1 \circ \overline{W_1}$, theorem 2(iii) implies

$$\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \subseteq \overline{W_1} \circ G_1 \circ \overline{W_1}^{-1} = G_1 \circ \overline{W_1} \circ \overline{W_1}^{-1} \subseteq G_1 \circ id_{\Sigma_1}.$$

Combining this with (2),

$$\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \;\cup\; id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \subseteq G_1 \circ id_{\Sigma_1} \subseteq G_1. \tag{3}$$

Define a relation $R = \overline{p} \circ \overline{W_1}^{-1} \circ id_Y \cup \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1}$. Then

$$\begin{aligned}
id_{N_0 \cup \Sigma_0} \circ R &= id_{N_0 \cup \Sigma_0} \circ \overline{p} \circ \overline{W_1}^{-1} \circ id_Y \;\cup\; id_{N_0 \cup \Sigma_0} \circ \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} \\
&= \overline{p} \circ id_{N_1 \cup \Sigma_1} \circ id_{N_1 \cup Y} \circ \overline{W_1}^{-1} \;\cup\; \overline{p} \circ id_{N_1 \cup \Sigma_1} \circ id_{E_1} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} && \text{by (1)} \\
&= \overline{p} \circ id_{N_1 \cup Y} \circ \overline{W_1}^{-1} \;\cup\; \perp \\
&= \overline{p} \circ \overline{W_1}^{-1} \circ id_Y && \text{by (1).}
\end{aligned} \tag{4}$$

$$\begin{aligned}
id_{N_0} \circ R &= id_{N_0} \circ id_{N_0 \cup \Sigma_0} \circ R \\
&= id_{N_0} \circ \overline{p} \circ \overline{W_1}^{-1} \circ id_Y \quad \text{by (4)} \\
&= \overline{p} \circ id_{N_1} \circ \overline{W_1}^{-1} \circ id_Y.
\end{aligned} \tag{5}$$

$$\begin{aligned}
id_{\Sigma_0} \circ R &= id_{\Sigma_0} \circ id_{N_0 \cup \Sigma_0} \circ R \\
&= id_{\Sigma_0} \circ \overline{p} \circ \overline{W_1}^{-1} \circ id_Y \quad \text{by (4)} \\
&= \overline{p} \circ id_{\Sigma_1} \circ \overline{W_1}^{-1} \circ id_Y \\
&= \overline{p} \circ id_{\Sigma_1} \circ id_Y \qquad \text{since } \forall \sigma \in \Sigma_1 \; W_1(\sigma) = \sigma \\
&= \overline{p} \circ id_Y.
\end{aligned} \tag{6}$$

$$\begin{aligned}
id_{E_0} \circ R &= id_{E_0} \circ \overline{p} \circ \overline{W_1}^{-1} \circ id_Y \;\cup\; id_{E_0} \circ \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} \\
&= \overline{p} \circ id_{E_1} \circ id_{N_1 \cup Y} \circ \overline{W_1}^{-1} \;\cup\; \overline{p} \circ id_{E_1} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} \quad \text{by (1)} \\
&= \perp \;\cup\; \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} \\
&= \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1}.
\end{aligned} \tag{7}$$

Hence

$$
\begin{aligned}
\overline{A_0}^{-1} \circ R &= \overline{A_0}^{-1} \circ id_{N_0 \cup \Sigma_0} \circ R \\
&= \overline{A_0}^{-1} \circ \overline{p} \circ \overline{W_1}^{-1} \circ id_Y && \text{by (4)} \\
&= \overline{p} \circ \overline{A_1}^{-1} \circ id_{N_1 \cup Y} \circ \overline{W_1}^{-1} && \text{by theorem 19(iv) and (1)} \\
&= \overline{p} \circ id_{A_1{}^{-1}(N_1 \cup Y)} \circ \overline{A_1}^{-1} \circ \overline{W_1}^{-1} \\
&= \overline{p} \circ \left(\overline{F_1} \circ \overline{F_1}^{-1} \cup \overline{S_1} \circ \overline{S_1}^{-1}\right) \circ \overline{A_1}^{-1} \circ \overline{W_1}^{-1} && \text{by theorem 7 and (a)} \\
&= \overline{p} \circ \overline{F_1} \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} \circ \overline{W_1}^{-1} \ \cup\ \overline{p} \circ \overline{S_1} \circ \overline{S_1}^{-1} \circ \overline{A_1}^{-1} \circ \overline{W_1}^{-1} \\
&= \overline{F_0} \circ \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} \ \cup\ \overline{S_0} \circ \overline{p} \circ \overline{W_1 \circ A_1 \circ S_1}^{-1} && \text{since } p \text{ is a homomorphism} \\
&= \overline{F_0} \circ \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} \ \cup\ \overline{S_0} \circ \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} && \text{by hypothesis (b)} \\
&= \left(\overline{F_0} \cup \overline{S_0}\right) \circ \overline{p} \circ \overline{W_1 \circ A_1 \circ F_1}^{-1} \\
&= \left(\overline{F_0} \cup \overline{S_0}\right) \circ id_{E_0} \circ R && \text{by (7)} \\
&= \left(\overline{F_0} \cup \overline{S_0}\right) \circ R.
\end{aligned}
$$

Since $\mathcal{N}_0$ is semi-definite we may infer

$$
G_{0\Sigma} \circ R \subseteq \overline{W_0} \circ G_{0N} \circ R \ \cup\ id_{\Sigma_0} \circ \overline{A_0} \circ G_{0H} \circ \overline{A_0}^{-1} \circ R. \tag{8}
$$

Then

$$
\begin{aligned}
&\overline{p} \circ G_1 \circ id_Y \\
&\quad = G_0 \circ \overline{p} \circ id_Y && \text{since } p \text{ is a homomorphism} \\
&\quad = G_0 \circ id_{\Sigma_0} \circ R && \text{by (6)} \\
&\quad = G_{0\Sigma} \circ R \\
&\quad \subseteq \overline{W_0} \circ G_{0N} \circ R \ \cup\ id_{\Sigma_0} \circ \overline{A_0} \circ G_{0H} \circ \overline{A_0}^{-1} \circ R && \text{by (8)} \\
&\quad = \overline{W_0} \circ G_0 \circ id_{N_0} \circ R \ \cup\ id_{\Sigma_0} \circ \overline{A_0} \circ G_0 \circ \overline{A_0}^{-1} \circ id_{N_0 \cup \Sigma_0} \circ R \\
&\quad = \overline{W_0} \circ G_0 \circ \overline{p} \circ id_{N_1} \circ \overline{W_1}^{-1} \circ id_Y \ \cup\ id_{\Sigma_0} \circ \overline{A_0} \circ G_0 \circ \overline{A_0}^{-1} \circ \overline{p} \circ \overline{W_1}^{-1} \circ id_Y \\
&\quad && \text{by (5) and (4)} \\
&\quad = \overline{W_0} \circ G_0 \circ \overline{p} \circ id_{N_1} \circ \overline{W_1}^{-1} \circ id_Y \ \cup\ id_{\Sigma_0} \circ \overline{A_0} \circ G_0 \circ \overline{p} \circ \overline{A_1}^{-1} \circ \overline{W_1}^{-1} \circ id_Y \\
&\quad && \text{by theorem 19(iv)} \\
&\quad = \overline{p} \circ \overline{W_1} \circ G_1 \circ id_{N_1} \circ \overline{W_1}^{-1} \circ id_Y \ \cup\ \overline{p} \circ id_{\Sigma_1} \circ \overline{A_1} \circ G_1 \circ \overline{A_1}^{-1} \circ \overline{W_1}^{-1} \circ id_Y \\
&\quad && \text{since } p \text{ is a homomorphism} \\
&\quad = \overline{p} \circ \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup\ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ \overline{W_1}^{-1}\right) \circ id_Y \\
&\quad = \overline{p} \circ \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup\ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_{\Sigma_1} \circ \overline{W_1}^{-1}\right) \circ id_Y && \text{by part (i)} \\
&\quad = \overline{p} \circ \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup\ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_{\Sigma_1}\right) \circ id_Y && \text{since } \forall \sigma \in \Sigma_1\ W_1(\sigma) = \sigma \\
&\quad = \overline{p} \circ \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup\ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}\right) \circ id_Y && \text{by part (i).}
\end{aligned} \tag{9}
$$

Define a relation

$$
G^* = \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup\ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}\right) \circ id_Y \ \cup\ G_1 \circ id_{(\Sigma_1 \setminus Y) \cup N_1 \cup H_1 \cup E_1 \cup K_1}.
$$

Comparing G^* with $G_1 = G_1 \circ id_Y \;\cup\; G_1 \circ id_{(\Sigma_1 \backslash Y) \cup N_1 \cup H_1 \cup E_1 \cup K_1}$, we have $G^* \subseteq G_1$ by (3), and hence $\overline{p} \circ G^* \subseteq \overline{p} \circ G_1$, but also $\overline{p} \circ G_1 \subseteq \overline{p} \circ G^*$ by (9), and hence $\overline{p} \circ G^* = \overline{p} \circ G_1$. Since p is a homomorphism, G_1 is minimal relative to p, so $G^* = G_1$. Hence

$$G_1 \circ id_Y = G^* \circ id_Y = \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \;\cup\; id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}\right) \circ id_Y$$

as required.

(iii) We have

$$\begin{aligned} G_1 \circ id_Y &= \overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \circ id_Y \;\cup\; id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_Y && \text{by part (ii)} \\ &= \overline{W_1} \circ G_{1N} \circ id_{N_1 \cup Y} \circ \overline{W_1}^{-1} \;\cup\; id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ id_{A_1{}^{-1}(Y)} \circ \overline{A_1}^{-1} && \text{by (1)} \\ &\subseteq \overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \;\cup\; id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{F_1} \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} \\ &\quad \cup\; id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{S_1} \circ \overline{S_1}^{-1} \circ \overline{A_1}^{-1} && \text{by theorem 7 and (a).} \end{aligned} \tag{10}$$

Let us consider each of the three terms on the right-hand side in turn. First,

$$\begin{aligned} \overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} &= \overline{W_1} \circ id_{N_1 \cup Y} \circ G_{1N} \circ \overline{W_1}^{-1} \\ &= id_Y \circ \overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} && \text{by (1).} \end{aligned}$$

Secondly,

$$\begin{aligned} &id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{F_1} \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} \\ &\quad = id_{\Sigma_1} \circ \overline{A_1} \circ G_1 \circ \overline{F_1} \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} && \text{since } F_1{:}\, E_1 \to H_1 \\ &\quad = id_{\Sigma_1} \circ \overline{A_1} \circ \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} && \text{by theorem 20(ii)} \\ &\quad = id_{\Sigma_1} \circ \overline{A_1} \circ id_{A_1{}^{-1}(N_1 \cup Y)} \circ \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} && \text{since } \mathrm{ran}(F_1) \subseteq A_1{}^{-1}(N_1 \cup Y) \text{, by (a)} \\ &\quad = id_{\Sigma_1} \circ id_{N_1 \cup Y} \circ \overline{A_1} \circ \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} \\ &\quad = id_Y \circ \overline{A_1} \circ \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1}. \end{aligned}$$

By a similar argument,

$$id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{S_1} \circ \overline{S_1}^{-1} \circ \overline{A_1}^{-1} = id_Y \circ \overline{A_1} \circ \overline{S_1} \circ G_1 \circ \overline{S_1}^{-1} \circ \overline{A_1}^{-1}.$$

Using these last three equations in (10),

$$G_1 \circ id_Y \subseteq id_Y \circ \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \;\cup\; \overline{A_1} \circ \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \circ \overline{A_1}^{-1} \;\cup\; \overline{A_1} \circ \overline{S_1} \circ G_1 \circ \overline{S_1}^{-1} \circ \overline{A_1}^{-1}\right).$$

Hence by theorem 4(vi)

$$G_1 \circ id_Y \subseteq id_Y \circ G_1$$

as required. ▮

Theorem 23. If $p{:}\, \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism from a network $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ satisfying

(a) $\forall X \subseteq \Sigma_1 \; |P_1{}^{-1}(X)| - |G_{1N} \circ id_{P_1{}^{-1}(X)}| \leq |X|$,

(b) $E_1 \overset{F_1}{\to} H_1 \overset{S_1}{\leftarrow} E_1$ is a sum diagram in the category of sets,

(c) $\exists f{:}\, K_1 \backslash dom(G_1) \to K_1 \; G_{1K}^{-1} = \overline{f}$,

(d) $E_1 \underset{A_1 \circ S_1}{\overset{A_1 \circ F_1}{\rightrightarrows}} N_1 \cup \Sigma_1 \overset{W_1}{\to} \Sigma_1$ is a coequaliser diagram in the category of sets,

to a semi-definite network $\mathcal{N}_0$, then $\mathcal{N}_1$ is definite.

Proof. We shall use the usual notation, $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$.

First note that the hypotheses of theorem 21 and theorem 22 hold, taking $Y = \Sigma_1$ (since $H_1 = A_1{}^{-1}(N_1 \cup \Sigma_1)$), so we can apply these theorems in what follows. In particular, theorem 22 gives

$$\begin{aligned} G_{1\Sigma} &= G_1 \circ id_{\Sigma_1} \\ &= \left(\overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup \ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1}\right) \circ id_{\Sigma_1} && \text{by theorem 22(ii)} \\ &= \overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \circ id_{\Sigma_1} \ \cup \ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_{\Sigma_1} \\ &= \overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup \ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} && \text{by theorem 22(i)} \qquad (1) \end{aligned}$$

and theorem 21(i) gives

$$G_{1H} = G_1 \circ id_{H_1} = \overline{F_1} \circ G_{1E} \circ \overline{F_1}^{-1} \ \cup \ \overline{S_1} \circ G_{1E} \circ \overline{S_1}^{-1}. \qquad (2)$$

Next we shall show that G_{1N} is connected relative to P_1. We already know the following.

$$\begin{aligned} G_{1N} &\text{ is finite} && \text{since } N_1 \text{ is finite} \\ G_{1N} \circ G_{1N} &\subseteq G_1 \circ G_1 = \bot && \text{since } \mathcal{N}_1 \text{ is a network} \\ G_{1N} &\text{ is acyclic} && \text{by theorem 21(vi)} \\ \overline{P_1} \circ G_{1N} &\subseteq \overline{P_1} \circ G_1 \subseteq \overline{P_1} && \text{since } \mathcal{N}_1 \text{ is a network} \\ \forall X \subseteq \Sigma_1 \ |P_1{}^{-1}(X)| - |G_{1N} &\circ id_{P_1{}^{-1}(X)}| \leq |X| && \text{by hypothesis (a)} \end{aligned}$$

So G_{1N} is connected relative to P_1, by theorem 13(iii).

Next we shall show that G_1 is minimal relative to $\mathcal{N}_1$. We already know the following.

$$\begin{aligned} G_{1K} \circ G_{1K}^{-1} &\subseteq id_{K_1} \subseteq G_{1K} \circ G_{1K}^{-1} \cup G_{1K}^{-1} \circ G_{1K} && \text{by theorem 21(ii)} \\ G_{1\Sigma} &= \overline{W_1} \circ G_{1N} \circ \overline{W_1}^{-1} \ \cup \ id_{\Sigma_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} && \text{by (1)} \\ G_{1N} &= id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} && \text{by theorem 20(iii)} \\ G_{1H} &= \overline{F_1} \circ G_{1E} \circ \overline{F_1}^{-1} \cup \overline{S_1} \circ G_{1E} \circ \overline{S_1}^{-1} && \text{by (2)} \\ G_{1E} &= \overline{C_1} \circ G_{1K} \circ \overline{C_1}^{-1} && \text{by theorem 20(iv)} \end{aligned}$$

Then G_1 is minimal relative to $\mathcal{N}_1$ by theorem 17.

Finally we shall check the conditions for $\mathcal{N}_1$ to be definite.

$$\begin{aligned} &E_1 \underset{A_1 \circ S_1}{\overset{A_1 \circ F_1}{\rightrightarrows}} N_1 \cup \Sigma_1 \overset{W_1}{\to} \Sigma_1 \text{ is a coequaliser diagram} && \text{by hypothesis (d)} \\ &G_{1N} \text{ is connected relative to } P_1 && \text{as just shown} \\ &G_{1N} \text{ is acyclic} && \text{by theorem 21(vi)} \\ &E_1 \overset{F_1}{\to} H_1 \overset{S_1}{\leftarrow} E_1 \text{ is a sum diagram} && \text{by hypothesis (b)} \\ &id_{K_1} = G_{1K} \circ G_{1K}^{-1} \cup G_{1K}^{-1} \circ G_{1K} && \text{by theorem 21(ii)} \\ &G_1 \text{ is minimal relative to } \mathcal{N}_1 && \text{as just shown} \end{aligned}$$

Thus $\mathcal{N}_1$ is definite. ∎

3.7 Subnetworks

Definition. If $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is a network, a *subnetwork* of $\mathcal{N}$ is a 12-tuple $(\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, where

$$\Sigma' \subseteq \Sigma, \quad N' \subseteq N, \quad H' \subseteq H, \quad E' \subseteq E, \quad K' \subseteq K$$
$$W(N') \subseteq \Sigma', \quad P(N') \subseteq \Sigma', \quad H' = A^{-1}(N' \cup \Sigma'), \quad F(E') \subseteq H', \quad S(E') \subseteq H', \quad K' = C^{-1}(E')$$
$$W' = W|_{N' \cup \Sigma'}, \quad P' = P|_{N'}, \quad A' = A|_{H'}, \quad F' = F|_{E'}, \quad S' = S|_{E'}, \quad C' = C|_{K'}$$
$$G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \subseteq id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G, \quad G' = id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}.$$

A *proper* subnetwork of $\mathcal{N}$ is a subnetwork $\mathcal{N}'$ such that $\mathcal{N}' \neq \mathcal{N}$.

Theorem 24. If $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is a network and $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$ is a subnetwork of $\mathcal{N}$ then

(i) $G' = G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$,
(ii) $\mathcal{N}'$ is a network,
(iii) $K' \setminus \mathrm{dom}(G') \subseteq K \setminus \mathrm{dom}(G)$,
(iv) the function $f: \Sigma' \cup N' \cup H' \cup E' \cup K' \to \Sigma \cup N \cup H \cup E \cup K$ defined by $\forall x \in \Sigma' \cup N' \cup H' \cup E' \cup K'\ f(x) = x$ is a homomorphism from $\mathcal{N}'$ to $\mathcal{N}$, called the *inclusion homomorphism*.

Proof. Define $I = id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$.

(i) By definition, $G' = I \circ G \circ I$, which equals $G \circ I$ by

$$I \circ G \circ I \subseteq G \circ I = G \circ I \circ I \subseteq I \circ G \circ I.$$

(ii) The conditions

$$W(N') \subseteq \Sigma', \quad P(N') \subseteq \Sigma', \quad H' = A^{-1}(N' \cup \Sigma'), \quad F(E') \subseteq H', \quad S(E') \subseteq H', \quad K' = C^{-1}(E'),$$

together with $\forall \sigma \in \Sigma\ W(\sigma) = \sigma$, give us

$$W' : N' \cup \Sigma' \to \Sigma', \quad P' : N' \to \Sigma', \quad A' : H' \to N' \cup \Sigma', \quad F', S' : E' \to H', \quad C' : K' \to E'$$

and $\forall \sigma \in \Sigma'\ W'(\sigma) = \sigma$. Note for later use that these conditions can also be expressed in relation notation as

$$\overline{W'} = \overline{W} \circ I \subseteq I \circ \overline{W}, \quad \overline{P'} = \overline{P} \circ I \subseteq I \circ \overline{P}, \quad \overline{A'} = \overline{A} \circ I = I \circ \overline{A},$$
$$\overline{F'} = \overline{F} \circ I \subseteq I \circ \overline{F}, \quad \overline{S'} = \overline{S} \circ I \subseteq I \circ \overline{S}, \quad \overline{C'} = \overline{C} \circ I = I \circ \overline{C}.$$

Hence

$$I \circ \overline{W} \circ I \subseteq \overline{W} \circ I = \overline{W} \circ I \circ I \subseteq I \circ \overline{W} \circ I,$$

so $W' = \overline{W} \circ I = I \circ \overline{W} \circ I$.

Since $G' = I \circ G \circ I$, G' is a relation on $\Sigma' \cup N' \cup H' \cup E' \cup K'$.

To verify $id_{\Sigma'} \circ G' = G' \circ id_{\Sigma'}$:

$$id_{\Sigma'} \circ G' = id_{\Sigma'} \circ I \circ G \circ I = I \circ id_{\Sigma} \circ G \circ I = I \circ G \circ id_{\Sigma} \circ I = I \circ G \circ I \circ id_{\Sigma'} = G' \circ id_{\Sigma'}$$

and similarly for the conditions $id_{N'} \circ G' = G' \circ id_{N'}$, $id_{H'} \circ G' = G' \circ id_{H'}$, $id_{E'} \circ G' = G' \circ id_{E'}$ and $id_{K'} \circ G' = G' \circ id_{K'}$.

To verify that G' preserves incidence:

$$\overline{W'} \circ G' = \overline{W} \circ I \circ G \circ I = \overline{W} \circ G' = \overline{W} \circ G \circ I = G \circ \overline{W} \circ I = G \circ I \circ \overline{W} \circ I = G' \circ \overline{W'}$$

and

$$\overline{P'} \circ G' = \overline{P} \circ I \circ G \circ I = \overline{P} \circ G' = \overline{P} \circ G \circ I \subseteq \overline{P} \circ I = \overline{P'}$$

and

$$\overline{F'} \circ G' = \overline{F} \circ I \circ G \circ I = \overline{F} \circ G' = \overline{F} \circ G \circ I \subseteq G \circ \overline{F} \circ I = G \circ \overline{F} \circ I \circ I \subseteq G \circ I \circ \overline{F} \circ I = G' \circ \overline{F'}$$

and similarly for the conditions for A, S and C.

Next we verify the minimality conditions. We have

$$\begin{aligned}
\overline{A'}^{-1} \circ \overline{A'} \;\cap\; G'_H \circ {G'}_H^{-1} &= id_{H'} \circ \overline{A}^{-1} \circ \overline{A} \circ id_{H'} \;\cap\; G'_H \circ {G'}_H^{-1} \\
&= id_{H'} \circ \left(\overline{A}^{-1} \circ \overline{A} \;\cap\; G'_H \circ {G'}_H^{-1}\right) \circ id_{H'} && \text{by theorem 4(iii),(iv)} \\
&\subseteq id_{H'} \circ \left(\overline{A}^{-1} \circ \overline{A} \;\cap\; G_H \circ G_H^{-1}\right) \circ id_{H'} \\
&\subseteq id_{H'} \circ id_H \circ id_{H'} && \text{by theorem 14} \\
&= id_{H'}
\end{aligned}$$

so by theorem 14 G'_H is minimal relative to A'. The other minimality conditions follow similarly.

For the final condition, $G' \circ G' \subseteq G \circ G = \bot$.

(iii) Using $G' = G \circ I$,

$$\mathrm{dom}(G') \cap K' = \mathrm{dom}(G' \circ id_{K'}) = \mathrm{dom}(G \circ I \circ id_{K'}) = \mathrm{dom}(G \circ id_{K'}) = \mathrm{dom}(G) \cap K'$$

so $K' \setminus \mathrm{dom}(G') = K' \setminus \mathrm{dom}(G) \subseteq K \setminus \mathrm{dom}(G)$, as required.

(iv) The conditions

$$f(\Sigma') \subseteq \Sigma, \quad f(N') \subseteq N, \quad f(H') \subseteq H, \quad f(E') \subseteq E, \quad f(K') \subseteq K$$

are immediate, as are

$$W \circ f = f \circ W', \quad P \circ f = f \circ P', \quad F \circ f = f \circ F', \quad S \circ f = f \circ S'.$$

The pullback $\begin{array}{ccc} N \cup \Sigma & \xleftarrow{f|_{N' \cup \Sigma'}} & N' \cup \Sigma' \\ {\scriptstyle A}\uparrow & & \uparrow{\scriptstyle A'} \\ H & \xleftarrow{f|_{H'}} & H' \end{array}$ is verified as follows. First, $A \circ f|_{H'} = A|_{H'} = A' = f|_{N' \cup \Sigma'} \circ A'$. Secondly, given any set X and functions $p{:}X \to H$ and $q{:}X \to N' \cup \Sigma'$ such that $A \circ p = f|_{N' \cup \Sigma'} \circ q$, this means $A \circ p = q$, so $A(p(X)) \subseteq N' \cup \Sigma'$, so $p(X) \subseteq A^{-1}(N' \cup \Sigma') = H'$, so $p{:}X \to H'$ and $A' \circ p = q$, so there exists a unique function $i{:}X \to H'$ such that $f|_{H'} \circ i = p$ and $A' \circ i = q$, namely $i = p$.

The pullback $\begin{array}{ccc} E & \xleftarrow{f|_{E'}} & E' \\ {\scriptstyle C}\uparrow & & \uparrow{\scriptstyle C'} \\ K & \xleftarrow{f|_{K'}} & K' \end{array}$ is verified similarly.

Note that $\overline{f} = I$, and so, using $G' = G \circ I = I \circ G \circ I$,

$$\overline{f} \circ G' = I \circ G \circ I = G \circ I = G \circ \overline{f}$$

as required.

To verify that G' is minimal relative to f, consider any relation $R \subseteq G'$ such that $\overline{f} \circ R = \overline{f} \circ G'$. Now, G' is on $\Sigma' \cup N' \cup H' \cup E' \cup K'$ and therefore so is R. Hence $R = I \circ R = \overline{f} \circ R = \overline{f} \circ G' = I \circ G' = G'$, as required. ∎

Definition. If $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$ is a subnetwork of $\mathcal{N}$, with inclusion homomorphism $f{:}\mathcal{N}' \to \mathcal{N}$, the *restriction* of a homomorphism $p{:}\mathcal{N} \to \mathcal{N}^*$ to $\mathcal{N}'$, denoted $p|_{\mathcal{N}'}{:}\mathcal{N}' \to \mathcal{N}^*$, is defined as $p \circ f = p|_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$; this is a homomorphism by theorem 15.

3.8 The recognition problem and process (initial statement)

A grammar and a pattern are both represented as networks. Given a semi-definite network $\mathcal{N}_0$ (representing a grammar) and an image, the recognition problem is to construct a definite network $\mathcal{N}_1$ (representing a pattern) and a homomorphism $p\colon\mathcal{N}_1 \to \mathcal{N}_0$ (called the *parse*). (This statement will be refined in §4.7.)

The pattern is constructed by a process of successive extension, pruning, merging and partitioning; the pattern is not definite during this process but at the end the hypotheses of theorem 23 are satisfied and so the pattern is definite.

4. Embeddings and the Definite Match Function

4.1 Introduction

The account of the recognition problem given in the previous sections has one very obvious deficiency: it does not say what relation should hold between the pattern and the image to make the pattern a good interpretation of the image. Indeed, it does not make any use of the image at all, nor does it involve any geometry.

In this section this deficiency will be rectified. Each symbol (and node) in the pattern must be embedded in the image plane using an affine transformation; the collection of all these affine transformations is called an *embedding token*. The grammar imposes constraints on the embedding token, stipulating that if two symbols are grammatically related then their embedding transformations must be geometrically related; the collection of all these constraints is called an *embedding type*. Embedding types make use of the concept of a *fleximap*, introduced in (2004, §5).

The aim of recognition is to construct a definite pattern and embedding token that *match*, or satisfy the constraints of, the image, the grammar and embedding type. The degree of match is expressed by the *definite match function DM*.

This section defines embedding tokens and types (slightly modifying the definitions in (2004, §6.4)), and the *DM* function. It also refines the use of templates (introduced in (2004, §4)), defines a symmetry of the grammar, and demonstrates the affine invariance of the whole framework.

4.2 Embedding tokens and types

As in (2004), $\mathcal{G}$ is the group of affine transformations on the plane, a six-dimensional Lie group. The composition or product of two affine transformations is written as $g \cdot g'$. Composition of two higher-level functions such as homomorphisms, embedding tokens or embedding types will be written using the '$\circ$' symbol.

Definition. An *embedding token* for a network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is a function $u{:}\,\Sigma \cup N \to \mathcal{G}$ such that

$$\forall n, n^* \in N \ (G(n, n^*) \Rightarrow u(n) = u(n^*)).$$

The product $u_1 \cdot u_2$ of embedding tokens u_1 and u_2 is defined as in (2004, §6.4):

$$\forall x \in \Sigma \cup N \ (u_1 \cdot u_2)(x) = u_1(x) \cdot u_2(x).$$

It follows immediately that $u_1 \cdot u_2$ is an embedding token. As before we can also define the induced embedding token $u \circ f$ on $\mathcal{N}'$, where u is an embedding token on $\mathcal{N}$ and $f{:}\,\mathcal{N}' \to \mathcal{N}$ is a homomorphism. However, we must check that $u \circ f$ satisfies the definition of an embedding token, as follows.

Theorem 25. If u is an embedding token for $\mathcal{N}$ and $f{:}\,\mathcal{N}' \to \mathcal{N}$ is a homomorphism then $u \circ f$ is an embedding token for $\mathcal{N}'$.

Proof. We shall use the notation $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ and $u' = u \circ f$. Clearly $u'{:}\,\Sigma' \cup N' \to \mathcal{G}$. Consider any $n, n^* \in N'$ such that $G'(n, n^*)$. We have $\overline{f} \circ G' = G \circ \overline{f}$, and hence $G'(n, n^*)$ implies $(\overline{f} \circ G')(f(n), n^*)$, which implies $(G \circ \overline{f})(f(n), n^*)$, i.e., $G(f(n), f(n^*))$. Since u is an embedding token for $\mathcal{N}$ this implies $u(f(n)) = u(f(n^*))$, i.e., $u'(n) = u'(n^*)$. Thus u' is an embedding token for $\mathcal{N}'$. ∎

Definition. An *embedding type* for a network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is a sextuple $(sub, con, rel, symm, tem, in)$, in which $sub\colon \{(\sigma, \sigma^*) \in \Sigma \times \Sigma \mid G(\sigma, \sigma^*)\} \to Flex$, $con\colon N \to Flex$, $rel\colon E \to Flex$, $symm\colon \Sigma \to Sub(\mathcal{G})$, $tem\colon \Sigma \to Tem$, and $in\colon \Sigma \to Flex$ where $Flex$ is the set of all fleximaps, $Sub(\mathcal{G})$ is the set of all subgroups of $\mathcal{G}$, and Tem is the set of all templates; moreover, for every $\sigma \in \Sigma$, the fleximap $in(\sigma)$ must have nominal part equal to the identity transformation.

This definition of embedding type is just as in (2004, §6.4), except for the presence of sub. The meaning of sub is that if σ^* is a subsymbol of σ then $sub(\sigma, \sigma^*)$ is a fleximap describing the relationship between the embeddings of σ^* and σ.

Embedding tokens are constrained by a symmetry condition. If u is an embedding token for $\mathcal{N}$ and v is an embedding type for $\mathcal{N}$, where $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, the *symmetry condition* for $\mathcal{N}, u, v$ is

$$\forall n \in N \quad u(P(n))^{-1} \cdot u(n) \in symm(P(n)).$$

This states that each node n is embedded in the image plane in the same way as the part $P(n)$, up to a symmetry of $P(n)$.

Given an embedding type $v = (sub_1, con_1, rel_1, symm_1, tem_1, in_1)$ and an embedding token u for a network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, we can define an embedding type $v \cdot u$ for the same network by $v \cdot u = (sub_2, con_2, rel_2, symm_1, tem_2, in_2)$, where

$$\begin{aligned}
\forall \sigma, \sigma^* \in \Sigma \quad (G(\sigma, \sigma^*) \ \Rightarrow \ sub_2(\sigma, \sigma^*) &= u(\sigma)^{-1} \cdot sub_1(\sigma, \sigma^*) \cdot u(\sigma^*)) \\
\forall n \in N \quad con_2(n) &= u(W(n))^{-1} \cdot con_1(n) \cdot u(n) \\
\forall e \in E \quad rel_2(e) &= u(A(S(e)))^{-1} \cdot rel_1(e) \cdot u(A(F(e))) \\
\forall \sigma \in \Sigma \quad tem_2(\sigma) &= tem_1(\sigma) \circ u(\sigma) \\
\forall \sigma \in \Sigma \quad in_2(\sigma) &= u(\sigma)^{-1} \cdot in_1(\sigma) \cdot u(\sigma)
\end{aligned}$$

Given an embedding type $v = (sub, con, rel, symm, tem, in)$ for a network $\mathcal{N}$ and a homomorphism $f\colon \mathcal{N}' \to \mathcal{N}$, there is an induced embedding type $v \circ f$ for $\mathcal{N}'$ defined by

$$v \circ f = (sub', con \circ f, rel \circ f, symm \circ f, tem \circ f, in \circ f).$$

where

$$\forall \sigma, \sigma^* \in \Sigma \quad (G(\sigma, \sigma^*) \ \Rightarrow \ sub'(\sigma, \sigma^*) = sub(f(\sigma), f(\sigma^*))).$$

Theorem 26. (Lemma 1 of (2004, §6.4)) If $f\colon \mathcal{N}' \to \mathcal{N}$ is a homomorphism, v is an embedding type for $\mathcal{N}$, and u is an embedding token for $\mathcal{N}$, then

$$(v \cdot u) \circ f = (v \circ f) \cdot (u \circ f).$$

Proof. The proof is a slightly adapted version of the one in (2004, §6.4). Let $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ and $v = (sub_1, con_1, rel_1,$

$symm_1, tem_1, in_1)$. Then $v \cdot u = (sub_2, con_2, rel_2, symm_1, tem_2, in_2)$, where

$$\begin{aligned}
\forall \sigma, \sigma^* \in \Sigma \quad (G(\sigma, \sigma^*) \;\Rightarrow\; sub_2(\sigma, \sigma^*) &= u(\sigma)^{-1} \cdot sub_1(\sigma, \sigma^*) \cdot u(\sigma^*)) \\
\forall n \in N \quad con_2(n) &= u(W(n))^{-1} \cdot con_1(n) \cdot u(n) \\
\forall e \in E \quad rel_2(e) &= u(A(S(e)))^{-1} \cdot rel_1(e) \cdot u(A(F(e))) \\
\forall \sigma \in \Sigma \quad tem_2(\sigma) &= tem_1(\sigma) \circ u(\sigma) \\
\forall \sigma \in \Sigma \quad in_2(\sigma) &= u(\sigma)^{-1} \cdot in_1(\sigma) \cdot u(\sigma)
\end{aligned}$$

and hence

$$(v \cdot u) \circ f = (sub', con_2 \circ f, rel_2 \circ f, symm_1 \circ f, tem_2 \circ f, in_2 \circ f)$$

where

$$\forall \sigma, \sigma^* \in \Sigma' \quad (G'(\sigma, \sigma^*) \;\Rightarrow\; sub'(\sigma, \sigma^*) = sub_2(f(\sigma), f(\sigma^*))).$$

We also have $v \circ f = (sub'', con_1 \circ f, rel_1 \circ f, symm_1 \circ f, tem_1 \circ f, in_1 \circ f)$, where

$$\forall \sigma, \sigma^* \in \Sigma' \quad (G'(\sigma, \sigma^*) \;\Rightarrow\; sub''(\sigma, \sigma^*) = sub_1(f(\sigma), f(\sigma^*))).$$

Hence $(v \circ f) \cdot (u \circ f) = (sub_3, con_3, rel_3, symm_1 \circ f, tem_3, in_3)$, where

$$\begin{aligned}
\forall \sigma, \sigma^* \in \Sigma' \quad (G'(\sigma, \sigma^*) \;\Rightarrow\; sub_3(\sigma, \sigma^*) &= (u \circ f)(\sigma)^{-1} \cdot sub''(\sigma, \sigma^*) \cdot (u \circ f)(\sigma^*) \\
&= u(f(\sigma))^{-1} \cdot sub_1(f(\sigma), f(\sigma^*)) \cdot u(f(\sigma^*)) \\
&= sub_2(f(\sigma), f(\sigma^*)) = sub'(\sigma, \sigma^*)) \\
\forall n \in N' \quad con_3(n) &= (u \circ f)(W'(n))^{-1} \cdot (con_1 \circ f)(n) \cdot (u \circ f)(n) \\
&= u(f(W'(n)))^{-1} \cdot con_1(f(n)) \cdot u(f(n)) \\
&= u(W(f(n)))^{-1} \cdot con_1(f(n)) \cdot u(f(n)) \\
&= con_2(f(n)) = (con_2 \circ f)(n) \\
\forall e \in E' \quad rel_3(e) &= (u \circ f)(A'(S'(e)))^{-1} \cdot (rel_1 \circ f)(e) \cdot (u \circ f)(A'(F'(e))) \\
&= u(f(A'(S'(e))))^{-1} \cdot rel_1(f(e)) \cdot u(f(A'(F'(e)))) \\
&= u(A(S(f(e))))^{-1} \cdot rel_1(f(e)) \cdot u(A(F(f(e)))) \\
&= rel_2(f(e)) = (rel_2 \circ f)(e) \\
\forall \sigma \in \Sigma' \quad tem_3(\sigma) &= (tem_1 \circ f)(\sigma) \circ (u \circ f)(\sigma) \\
&= tem_1(f(\sigma)) \circ u(f(\sigma)) \\
&= tem_2(f(\sigma)) = (tem_2 \circ f)(\sigma) \\
\forall \sigma \in \Sigma' \quad in_3(\sigma) &= (u \circ f)(\sigma)^{-1} \cdot (in_1 \circ f)(\sigma) \cdot (u \circ f)(\sigma) \\
&= u(f(\sigma))^{-1} \cdot in_1(f(\sigma)) \cdot u(f(\sigma)) \\
&= in_2(f(\sigma)) = (in_2 \circ f)(\sigma)
\end{aligned}$$

which shows that $(v \cdot u) \circ f = (v \circ f) \cdot (u \circ f)$. ∎

Definition. If $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$ is a subnetwork of $\mathcal{N}$, with inclusion homomorphism $f : \mathcal{N}' \to \mathcal{N}$,

(i) the *restriction* of u, an embedding token for $\mathcal{N}$, to $\mathcal{N}'$, denoted $u|_{\mathcal{N}'}$, is defined as $u \circ f = u|_{\Sigma' \cup N'}$; this is an embedding token for $\mathcal{N}'$ by theorem 25;

(ii) the *restriction* of v, an embedding type for $\mathcal{N}$, to $\mathcal{N}'$, denoted $v|_{\mathcal{N}'}$, is defined as $v \circ f$, an embedding type for $\mathcal{N}'$.

4.3 Symmetries

Definition. A *symmetry* of a network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, with respect to the embedding type $v = (sub, con, rel, symm, tem, in)$, is a pair (a, s), where $a: \mathcal{N} \to \mathcal{N}$ is an automorphism of $\mathcal{N}$ and s is an embedding token for $\mathcal{N}$, such that

- $\forall \sigma \in \Sigma \; s(\sigma) \in symm(\sigma)$,
- $\forall n \in N \; s(n) \in symm(P(n))$,
- $v \cdot s = v \circ a$.

This is just as in (2004, §7.5). However, the operation of global application of a symmetry of the grammar to the pattern, in lemma 2, needs to be generalised to *local* symmetries, in which different symmetries are applied to different symbols of the pattern

Definition. Given networks $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ and $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$, a homomorphism $p: \mathcal{N}_1 \to \mathcal{N}_0$, and an embedding token v on $\mathcal{N}_0$, define $\mathcal{S}_0$ as the group of symmetries of $\mathcal{N}_0$ with respect to v. A *local symmetry* of $\mathcal{N}_1$ with respect to $\mathcal{N}_0, p, v$ is a function $\pi: \Sigma_1 \to \mathcal{S}_0$ such that

- $\forall \sigma, \sigma^* \in \Sigma_1 \; (G_1(\sigma, \sigma^*) \Rightarrow \pi(\sigma) = \pi(\sigma^*))$,
- $\forall e \in E_1 \; \pi(W_1(A_1(F_1(e)))) = \pi(W_1(A_1(S_1(e))))$,
- $\forall n \in N_1 \; a(P_1(n))(p(P_1(n))) = a(W_1(n))(p(P_1(n)))$,

where π is decomposed into components by $\forall \sigma \in \Sigma_1 \; \pi(\sigma) = (a(\sigma), s(\sigma))$, for functions a, s.

Given a local symmetry π and an embedding token u for $\mathcal{N}_1$, the *application of* π *to* p, u produces a new homomorphism $p': \mathcal{N}_1 \to \mathcal{N}_0$, defined by

$$\begin{aligned}
\forall \sigma \in \Sigma_1 &\quad p'(\sigma) = a(\sigma)(p(\sigma)) \\
\forall n \in N_1 &\quad p'(n) = a(W_1(n))(p(n)) \\
\forall h \in H_1 &\quad p'(h) = a(W_1(A_1(h)))(p(h)) \\
\forall e \in E_1 &\quad p'(e) = a(W_1(A_1(F_1(e))))(p(e)) \\
\forall k \in K_1 &\quad p'(k) = a(W_1(A_1(F_1(C_1(k)))))(p(k)),
\end{aligned}$$

and a new embedding token $u' = u \cdot s'$ for $\mathcal{N}_1$, where the embedding token s' is defined by

$$\begin{aligned}
\forall \sigma \in \Sigma_1 &\quad s'(\sigma) = s(\sigma)(p(\sigma)) \\
\forall n \in N_1 &\quad s'(n) = s(W_1(n))(p(n)).
\end{aligned}$$

Theorem 27. In the above definition, $p': \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism and s', u' are embedding tokens for $\mathcal{N}_1$, as claimed.

Proof. We shall continue to use the above notation.

First we verify that $p': \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism. The conditions

$$p'(\Sigma_1) \subseteq \Sigma_0, \quad p'(N_1) \subseteq N_0, \quad p'(H_1) \subseteq H_0, \quad p'(E_1) \subseteq E_0, \quad p'(K_1) \subseteq K_0$$

are immediate.

Next we check $W_0 \circ p' = p' \circ W_1$:

$$
\begin{aligned}
\forall n \in N_1 \quad W_0(p'(n)) &= W_0\big(a(W_1(n))(p(n))\big) \\
&= a(W_1(n))\big(W_0(p(n))\big) && \text{since } a(W_1(n)) \text{ is an automorphism} \\
&= a(W_1(n))\big(p(W_1(n))\big) && \text{since } p \text{ is an homomorphism} \\
&= p'(W_1(n)),
\end{aligned}
$$

$$\forall \sigma \in \Sigma_1 \quad W_0(p'(\sigma)) = p'(\sigma) = p'(W_1(\sigma)).$$

Next we check $P_0 \circ p' = p' \circ P_1$:

$$
\begin{aligned}
\forall n \in N_1 \quad P_0(p'(n)) &= P_0\big(a(W_1(n))(p(n))\big) \\
&= a(W_1(n))\big(P_0(p(n))\big) && \text{since } a(W_1(n)) \text{ is an automorphism} \\
&= a(W_1(n))\big(p(P_1(n))\big) && \text{since } p \text{ is an homomorphism} \\
&= a(P_1(n))\big(p(P_1(n))\big) && \text{since } \pi \text{ is a local symmetry} \\
&= p'(P_1(n)).
\end{aligned}
$$

Next we check $F_0 \circ p' = p' \circ F_1$:

$$
\begin{aligned}
\forall e \in E_1 \quad F_0(p'(e)) &= F_0\big(a(W_1(A_1(F_1(e))))(p(e))\big) \\
&= a(W_1(A_1(F_1(e))))\big(F_0(p(e))\big) && \text{since } a(W_1(A_1(F_1(e)))) \text{ is an automorphism} \\
&= a(W_1(A_1(F_1(e))))\big(p(F_1(e))\big) && \text{since } p \text{ is a homomorphism} \\
&= p'(F_1(e)).
\end{aligned}
$$

Next we check $S_0 \circ p' = p' \circ S_1$:

$$
\begin{aligned}
\forall e \in E_1 \quad S_0(p'(e)) &= S_0\big(a(W_1(A_1(F_1(e))))(p(e))\big) \\
&= a(W_1(A_1(F_1(e))))\big(S_0(p(e))\big) && \text{since } a(W_1(A_1(F_1(e)))) \text{ is an automorphism} \\
&= a(W_1(A_1(F_1(e))))\big(p(S_1(e))\big) && \text{since } p \text{ is a homomorphism} \\
&= a(W_1(A_1(S_1(e))))\big(p(S_1(e))\big) && \text{since } \pi \text{ is a local symmetry} \\
&= p'(S_1(e)).
\end{aligned}
$$

Next we check that $\begin{array}{ccc} N_0 \cup \Sigma_0 & \xleftarrow{p'|_{N_1\cup\Sigma_1}} & N_1 \cup \Sigma_1 \\ {\scriptstyle A_0}\uparrow & & \uparrow{\scriptstyle A_1} \\ H_0 & \xleftarrow{p'|_{H_1}} & H_1 \end{array}$ is a pullback in the category of sets. The equation $A_0 \circ p'|_{H_1} = p'|_{N_1 \cup \Sigma_1} \circ A_1$ follows by

$$
\begin{aligned}
\forall h \in H_1 \quad A_0(p'(h)) &= A_0\big(a(W_1(A_1(h)))(p(h))\big) \\
&= a(W_1(A_1(h)))\big(A_0(p(h))\big) && \text{since } a(W_1(A_1(h))) \text{ is an automorphism} \\
&= a(W_1(A_1(h)))\big(p(A_1(h))\big) && \text{since } p \text{ is an homomorphism} \\
&= p'(A_1(h)).
\end{aligned}
$$

The other half of the pullback condition is

$$\forall h_0 \in H_0 \ \forall x_1 \in N_1 \cup \Sigma_1 \quad (A_0(h_0) = p'(x_1) \ \Rightarrow \ \exists! h_1 \in H_1 \ (A_1(h_1) = x_1 \wedge p'(h_1) = h_0)). \qquad (1)$$

To prove this, consider any $h_0 \in H_0$ and any $x_1 \in N_1 \cup \Sigma_1$. Define $a^* = a(W_1(x_1))$, an automorphism of $\mathcal{N}_0$. By theorem 15, $a^* \circ p : \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, so the pullback $\begin{tikzcd} N_0 \cup \Sigma_0 & N_1 \cup \Sigma_1 \arrow[l, "(a^* \circ p)|_{N_1 \cup \Sigma_1}"'] \\ H_0 \arrow[u, "A_0"] & H_1 \arrow[l, "(a^* \circ p)|_{H_1}"] \arrow[u, "A_1"'] \end{tikzcd}$ implies

$$A_0(h_0) = a^*(p(x_1)) \;\Rightarrow\; \exists! h_1 \in H_1 \; (A_1(h_1) = x_1 \wedge a^*(p(h_1)) = h_0). \tag{2}$$

Now, $p'(x_1) = a^*(p(x_1))$; and, for any $h_1 \in H_1$, if $A_1(h_1) = x_1$ then $p'(h_1) = a(W_1(A_1(h_1)))(p(h_1)) = a^*(p(h_1))$. Thus (2) may be rewritten, and quantifiers over h_0 and x_1 added, to give (1). This completes the proof of the pullback.

The pullback $\begin{tikzcd} E_0 & E_1 \arrow[l, "p'|_{E_1}"'] \\ K_0 \arrow[u, "C_0"] & K_1 \arrow[l, "p'|_{K_1}"] \arrow[u, "C_1"'] \end{tikzcd}$ is verified similarly.

Next we check that $G_0 \circ \overline{p'} = \overline{p'} \circ G_1$ and G_1 is minimal relative to p'. Let us restrict our attention to nodes in the first instance. In view of theorem 14, we have to show

$$\begin{aligned} &\forall n_0 \in N_0 \; \forall n_1^* \in N_1 \;\; (G_0(n_0, p'(n_1^*)) \;\Leftrightarrow\; \exists n_1 \in N_1 \; (n_0 = p'(n_1) \wedge G_1(n_1, n_1^*))); \\ &\forall n_0 \in N_0 \; \forall n_1^* \in N_1 \;\; \exists^{\leq 1} n_1 \in N_1 \; (n_0 = p'(n_1) \wedge G_1(n_1, n_1^*)). \end{aligned} \tag{3}$$

To prove these, consider any $n_0 \in N_0$ and any $n_1^* \in N_1$. Define $a^* = a(W_1(n_1^*))$, an automorphism of $\mathcal{N}_0$. By theorem 15, $a^* \circ p : \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, so

$$\begin{aligned} &G_0(n_0, a^*(p(n_1^*))) \;\Leftrightarrow\; \exists n_1 \in N_1 \; (n_0 = a^*(p(n_1)) \wedge G_1(n_1, n_1^*)); \\ &\exists^{\leq 1} n_1 \in N_1 \; (n_0 = a^*(p(n_1)) \wedge G_1(n_1, n_1^*)). \end{aligned} \tag{4}$$

Now, the condition $G_0(n_0, a^*(p(n_1^*)))$ is equivalent to $G_0(n_0, p'(n_1^*))$, since $a^*(p(n_1^*)) = p'(n_1^*)$. Moreover, for any $n_1 \in N_1$, if $G_1(n_1, n_1^*)$ then $G_1(W_1(n_1), W_1(n_1^*))$, by $\overline{W_1} \circ G_1 \subseteq G_1 \circ \overline{W_1}$, so $\pi(W_1(n_1)) = \pi(W_1(n_1^*))$, so $a(W_1(n_1)) = a(W_1(n_1^*)) = a^*$, so $p'(n_1) = a(W_1(n_1))(p(n_1)) = a^*(p(n_1))$. Thus the condition $n_0 = a^*(p(n_1)) \wedge G_1(n_1, n_1^*)$ is equivalent to $n_0 = p'(n_1) \wedge G_1(n_1, n_1^*)$. Hence (4) may be rewritten (quantifying over n_0 and n_1^*) as (3), as required. This deals with the node case. Statements similar to (3) for symbols, hooks, edges and facets can be proved in the same way, completing the proof that $G_0 \circ \overline{p'} = \overline{p'} \circ G_1$ and G_1 is minimal relative to p'.

This completes the proof that $p' : \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism.

Next we verify that s' is an embedding token for $\mathcal{N}_1$. Consider any $n, n^* \in N_1$ such that $G_1(n, n^*)$ holds. We must show that $s'(n) = s'(n^*)$.

First, $\pi(W_1(n))$ is a symmetry of $\mathcal{N}_0$, so $s(W_1(n))$ is an embedding token for $\mathcal{N}_0$; moreover, $G_0(p(n), p(n^*))$ holds; so $s(W_1(n))(p(n)) = s(W_1(n))(p(n^*))$.

Secondly, we have $G_1(W_1(n), W_1(n^*))$, so $\pi(W_1(n)) = \pi(W_1(n^*))$, so $s(W_1(n)) = s(W_1(n^*))$.

Putting these two observations together,

$$s'(n) = s(W_1(n))(p(n)) = s(W_1(n))(p(n^*)) = s(W_1(n^*))(p(n^*)) = s'(n^*)$$

as required. It follows now that $u' = u \cdot s'$ is an embedding token for $\mathcal{N}_1$. ∎

4.4 Templates and saturation

The concepts of an image, a template, and the correlation function $\rho_{I,T}$ are carried over from (2004, §4), with two modifications.

The first modification is intended to prevent two identical symbol tokens from forming at the same place in the image, or more generally to prevent two or more symbol tokens from claiming credit for the same patch of the image. If two or more symbol tokens' templates overlap in the image plane, i.e., if there are points x in the image plane where $tem(\sigma)(u(\sigma)^{-1}(x)) > 0$ for two or more symbol tokens σ, then the point x should become *saturated* and the contribution it makes to the correlation function should be reduced. For this it is necessary to calculate the saturation $sat(x)$ of each point x, which is roughly the sum of $tem(\sigma)(u(\sigma)^{-1}(x))$ over each symbol token σ. (However, this needs modification for subsymbols to avoid double-counting.) The formal definition of saturation follows.

Suppose we are given a definite network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, an embedding token u for $\mathcal{N}$, and a function $tem\colon \Sigma \to Tem$. The *(definite) saturation function*, $sat\colon \mathbb{R}^2 \to \mathbb{R}$, is defined by

$$\forall x \in \mathbb{R}^2 \quad sat(x) = \sum_{\sigma \in \Sigma} (1 - g_\sigma)\, tem(\sigma)(u(\sigma)^{-1}(x))$$

where $g_\sigma = \big|\{\, \sigma^* \in \Sigma \mid G(\sigma^*, \sigma) \,\}\big|$. We call $sat(x)$ the *saturation at* x.

The correlation function $\rho_{I,T}$, between an image I and a template T, was defined in (2004, §4.2) by

$$\rho_{I,T}(g) = |det(\overline{g})| \int T(u)\,(I(g(u)) - I_0)\,\mathrm{d}^2u = \int T(g^{-1}(x))\,(I(x) - I_0)\,\mathrm{d}^2x$$

for any affine transformation g, where $\overline{g}$ is the matrix representation of g and I_0 is a positive real constant associated with T. This must now be modified to take account of saturation:

$$\rho_{I,T,sat}(g) = |det(\overline{g})| \int w(sat(g(u)))\,T(u)\,(I(g(u)) - I_0)\,\mathrm{d}^2u = \int w(sat(x))\,T(g^{-1}(x))\,(I(x) - I_0)\,\mathrm{d}^2x$$

where w is a weighting function that suppresses the integrand at points x where $sat(x)$ is above 1. A suitable definition of w is

$$\forall s \in \mathbb{R} \quad w(s) = \begin{cases} 1 & \text{if } s \le 1, \\ (1.6 - s)/0.6 & \text{if } 1 < s < 1.6, \\ 0 & \text{if } 1.6 \le s. \end{cases}$$

The second modification is to give the factor of $|det(\overline{g})|$ that occurs in the formula for $\rho_{I,T,sat}(g)$ an adjustable exponent:

$$\begin{aligned} \rho_{I,T,sat}(g) &= |det(\overline{g})|^{1-k} \int w(sat(g(u)))\,T(u)\,(I(g(u)) - I_0)\,\mathrm{d}^2u \\ &= |det(\overline{g})|^{-k} \int w(sat(x))\,T(g^{-1}(x))\,(I(x) - I_0)\,\mathrm{d}^2x \end{aligned}$$

where k is a constant characteristic of the template, with $0 \le k \le 1$ normally.

The calculation of template force in (2004, §4.3) is affected by these two changes as follows. For any $g \in \mathcal{G}$ and $A \in \mathcal{A}$,

$$
\begin{aligned}
\rho_{I,T,sat}(g \cdot \exp A) &= |\,det(\overline{g \cdot \exp A})|^{-k} \int w(sat(x))\, T((g \cdot \exp A)^{-1}(x))(I(x) - I_0)\, \mathrm{d}^2 x \\
&= |\,det(\overline{g} \cdot \exp \overline{A})|^{-k} \int w(sat(x))\, \overline{T}((\overline{g} \exp \overline{A})^{-1} \overline{x})(I(x) - I_0)\, \mathrm{d}^2 x \\
&= |\,det(\overline{g})|^{-k} e^{-k\, \mathrm{tr}(\overline{A})} \int w(sat(x))\, \overline{T}(\exp(-\overline{A})\overline{g}^{-1}\overline{x})(I(x) - I_0)\, \mathrm{d}^2 x \\
&= |\,det(\overline{g})|^{-k}(\mathbb{I} - k\, \mathrm{tr}(\overline{A}) + o(A)) \int w(sat(x))\, \overline{T}((\mathbb{I} - \overline{A} + o(A))\overline{g}^{-1}\overline{x})(I(x) - I_0)\, \mathrm{d}^2 x \\
&= |\,det(\overline{g})|^{-k}(\mathbb{I} - k\, \mathrm{tr}(\overline{A}) + o(A)) \cdot \\
&\qquad \int w(sat(x))\, \big(\overline{T}(\overline{g}^{-1}\overline{x}) - (\overline{\nabla} T(g^{-1}x))\overline{A}\,\overline{g}^{-1}\overline{x} + o(A)\big)(I(x) - I_0)\, \mathrm{d}^2 x \\
&= \rho_{I,T,sat}(g) - k\, \mathrm{tr}(\overline{A})\rho_{I,T,sat}(g) \\
&\qquad - |\,det(\overline{g})|^{-k} \int w(sat(x))\, \big((\overline{\nabla} T(g^{-1}x))\overline{A}\,\overline{g}^{-1}\overline{x}\big)(I(x) - I_0)\, \mathrm{d}^2 x + o(A)
\end{aligned}
$$

where $\mathbb{I}$ is the 6×6 identity matrix. Hence the derivative $\rho_{I,T,sat*} \colon \mathcal{G} \to (\mathcal{A} \to \mathbb{R})$ is defined by

$$
\begin{aligned}
\forall g \in \mathcal{G}\ \forall A \in \mathcal{A}\quad \rho_{I,T,sat*}(g)(A) &= -\,k\, \mathrm{tr}(\overline{A})\rho_{I,T,sat}(g) \\
&\qquad - |\,det(\overline{g})|^{-k} \int w(sat(x))\, \big((\overline{\nabla} T(g^{-1}x))\overline{A}\,\overline{g}^{-1}\overline{x}\big)(I(x) - I_0)\, \mathrm{d}^2 x \\
&= -\,k\, \mathrm{tr}(\overline{A})\rho_{I,T,sat}(g) \\
&\qquad - |\,det(\overline{g})|^{1-k} \int w(sat(g(u)))\, \big((\overline{\nabla} T(u))\overline{A}\,\overline{u}\big)(I(g(u)) - I_0)\, \mathrm{d}^2 u.
\end{aligned}
$$

This gives a matrix representation for the force,

$$
\begin{aligned}
\forall g \in \mathcal{G}\quad \overline{\rho_{I,T,sat*}(g)} &= -\,k\rho_{I,T,sat}(g)\mathbb{I} - |\,det(\overline{g})|^{-k} \int w(sat(x))\, \big(\overline{g}^{-1}\overline{x}(\overline{\nabla} T(g^{-1}(x)))\big)(I(x) - I_0)\, \mathrm{d}^2 x \\
&= -\,k\rho_{I,T,sat}(g)\mathbb{I} - |\,det(\overline{g})|^{1-k} \int w(sat(g(u)))\, \big(\overline{u}(\overline{\nabla} T(u))\big)(I(g(u)) - I_0)\, \mathrm{d}^2 u.
\end{aligned}
$$

4.5 The definite match function

The *definite match function DM* measures how well a definite network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ matches an image I, given an embedding token u for $\mathcal{N}$, and an embedding type $v = (sub, con, rel, symm, tem, in)$ for $\mathcal{N}$. It is defined by

$$
\begin{aligned}
DM(I, \mathcal{N}, u, v) = \sum_{\sigma \in \Sigma} \rho_{I,tem(\sigma),sat}(u(\sigma)) \;-\; \theta|\Sigma \backslash P(N)| \;-\; \sum_{(\sigma,\sigma^*)|G(\sigma,\sigma^*)} E_{sub(\sigma,\sigma^*)}\big(u(\sigma)^{-1} \cdot u(\sigma^*)\big) \\
-\; \sum_{n \in N} E_{con(n)}\big(u(W(n))^{-1} \cdot u(n)\big) \;-\; \sum_{e \in E} E_{rel(e)}\big(u(A(S(e)))^{-1} \cdot u(A(F(e)))\big)
\end{aligned}
$$

where θ is a positive real constant and *sat* is the definite saturation function, as defined in the previous section.

The first term on the right-hand side is a sum over all symbols; $tem(\sigma)$ is the template for σ, and $\rho_{I,tem(\sigma),sat}(u(\sigma))$ measures the correlation between the template (embedded in the image using $u(\sigma)$) and the image I.

The second term on the right-hand side applies a fixed penalty of θ for each 'bare' symbol, i.e., each symbol that is not a part of another symbol. This term encourages the symbols to connect themselves together rather than remaining separate.

The third term measures how well each subsymbol matches its supersymbol geometrically. The summation is over all pairs (σ,σ^*) such that σ is a supersymbol of σ^*; $sub(\sigma,\sigma^*)$ is a fleximap that defines what the geometric relationship between σ^* and σ should be; $u(\sigma)^{-1}\cdot u(\sigma^*)$ is the actual relationship between them; the quadratic penalty function $E_\tau(G)$ calculates the penalty for the deviation between an affine transformation G and a fleximap τ, as defined in (2004, §5.3).

The fourth term measures how well each part matches its whole geometrically. The summation is over all nodes n; $P(n)$ is the part symbol and $W(n)$ is the whole symbol. The node n has its own embedding $u(n)$, which equals $u(P(n))$ up to a symmetry; $u(W(n))^{-1}\cdot u(n)$ is the actual geometric relationship between the part (or rather the node) and the whole; $con(n)$ is a fleximap specifying what the relationship should be.

The final term measures how well each pair of siblings match geometrically. The sum is over every edge e, representing a sibling relationship between two nodes $A(F(e))$ and $A(S(e))$; $u(A(S(e)))^{-1}\cdot u(A(F(e)))$ is the actual relationship and $rel(e)$ is the fleximap specifying what the relationship should be.

All these fleximaps are provided by the embedding type v; the final component in of v is not used yet.

The way DM will be used in the statement of the recognition problem is as follows. An embedding type v for the grammar $\mathcal{N}_0$ is given at the outset. The parsing homomorphism $p{:}\mathcal{N}_1\to\mathcal{N}_0$ induces an embedding type $v\circ p$ for $\mathcal{N}_1$. An embedding token u for $\mathcal{N}_1$ is constructed by the parser, satisfying the symmetry condition for $\mathcal{N}_1,u,v\circ p$. At the end of recognition, when $\mathcal{N}_1$ is definite, we measure how well u matches $v\circ p$ by calculating $DM(I,\mathcal{N}_1,u,v\circ p)$.

4.6 Invariance theorems

Lemmas 1 and 2 from (2004, §7) hold for the definite match function DM; I shall restate them here, but with lemma 2 generalised from global symmetries to local symmetries. I shall also add two new theorems on the affine invariance of the ρ and match functions.

Theorem 28. (Lemma 1 of (2004, §7.3): invariance under affine transformation of the symbols' internal frames of reference.) For any image I, definite network $\mathcal{N}$, embedding tokens u,u' for $\mathcal{N}$, and embedding type v for $\mathcal{N}$, then

$$DM(I,\mathcal{N},u\cdot u',v\cdot u')=DM(I,\mathcal{N},u,v);$$

Proof. The proof is very similar to the one in (2004), but I shall restate it in full. Let $v=(sub_1,con_1,rel_1,symm_1,tem_1,in_1)$; then $v\cdot u'=(sub_2,con_2,rel_2,symm_1,tem_2,in_2)$, where

$$\begin{aligned}
\forall\sigma,\sigma^*\in\Sigma\quad (G(\sigma,\sigma^*) \;\Rightarrow\; sub_2(\sigma,\sigma^*) &= u'(\sigma)^{-1}\cdot sub_1(\sigma,\sigma^*)\cdot u'(\sigma^*))\\
\forall n\in N\quad con_2(n) &= u'(W(n))^{-1}\cdot con_1(n)\cdot u'(n)\\
\forall e\in E\quad rel_2(e) &= u'(A(S(e)))^{-1}\cdot rel_1(e)\cdot u'(A(F(e)))\\
\forall\sigma\in\Sigma\quad tem_2(\sigma) &= tem_1(\sigma)\circ u'(\sigma)\\
\forall\sigma\in\Sigma\quad in_2(\sigma) &= u'(\sigma)^{-1}\cdot in_1(\sigma)\cdot u'(\sigma)
\end{aligned}$$

Now, first, for any $\sigma \in \Sigma$,

$$\rho_{I,tem_2(\sigma),sat}((u \cdot u')(\sigma)) = \rho_{I,tem_1(\sigma)\circ u'(\sigma),sat}(u(\sigma) \cdot u'(\sigma)) = \rho_{I,tem_1(\sigma),sat}(u(\sigma))$$

using lemma 1 of (2004, §4.2). Note that the saturation function *sat* is unchanged by the application of u'. This is because

$$\forall \sigma \in \Sigma \ \forall x \in \mathbb{R}^2 \quad tem_2(\sigma)((u \cdot u')(\sigma)^{-1}(x)) = (tem_1(\sigma) \circ u'(\sigma))(u'(\sigma)^{-1}(u(\sigma)^{-1}(x))) = tem_1(\sigma)(u(\sigma)^{-1}(x))$$

and so the calculation of $sat(x)$ is unaffected.

Secondly, for any $\sigma, \sigma^* \in \Sigma$ such that $G(\sigma, \sigma^*)$,

$$\begin{aligned} E_{sub_2(\sigma,\sigma^*)}((u \cdot u')(\sigma)^{-1} \cdot (u \cdot u')(\sigma^*)) &= E_{u'(\sigma)^{-1} \cdot sub_1(\sigma,\sigma^*) \cdot u'(\sigma^*)}(u'(\sigma)^{-1} \cdot u(\sigma)^{-1} \cdot u(\sigma^*) \cdot u'(\sigma^*)) \\ &= E_{sub_1(\sigma,\sigma^*)}(u(\sigma)^{-1} \cdot u(\sigma^*)) \end{aligned}$$

by lemma 1 of (2004, §5.3).

Thirdly, for any $n \in N$,

$$\begin{aligned} &E_{con_2(n)}((u \cdot u')(W(n))^{-1} \cdot (u \cdot u')(n)) \\ &\qquad = E_{u'(W(n))^{-1} \cdot con_1(n) \cdot u'(n)}(u'(W(n))^{-1} \cdot u(W(n))^{-1} \cdot u(n) \cdot u'(n)) \\ &\qquad = E_{con_1(n)}(u(W(n))^{-1} \cdot u(n)) \end{aligned}$$

by lemma 1 of (2004, §5.3).

Fourthly, for any $e \in E$, using the abbreviations $n_1 = A(F(e))$ and $n_2 = A(S(e))$,

$$\begin{aligned} E_{rel_2(e)}((u \cdot u')(n_2)^{-1} \cdot (u \cdot u')(n_1)) &= E_{u'(n_2)^{-1} \cdot rel_1(e) \cdot u'(n_1)}(u'(n_2)^{-1} \cdot u(n_2)^{-1} \cdot u(n_1) \cdot u'(n_1)) \\ &= E_{rel_1(e)}(u(n_2)^{-1} \cdot u(n_1)) \end{aligned}$$

and, using the abbreviations $\sigma_1 = W(A(F(e)))$ and $\sigma_2 = W(A(S(e)))$,

$$\begin{aligned} E_{in_2(\sigma_1)}((u \cdot u')(\sigma_2)^{-1} \cdot (u \cdot u')(\sigma_1)) &= E_{u'(\sigma_1)^{-1} \cdot in_1(\sigma_1) \cdot u'(\sigma_1)}(u'(\sigma_2)^{-1} \cdot u(\sigma_2)^{-1} \cdot u(\sigma_1) \cdot u'(\sigma_1)) \\ &= E_{in_1(\sigma_1)}(u(\sigma_2)^{-1} \cdot u(\sigma_1)) \end{aligned}$$

by lemma 1 of (2004, §5.3) again, using the assumption that $u'(\sigma_1) = u'(\sigma_2)$ (which holds under the conditions of (i) or (ii)).

Thus each term of $DM(I, \mathcal{N}, u \cdot u', v \cdot u')$ equals the corresponding term of $DM(I, \mathcal{N}, u, v)$. ▮

Theorem 29. (Invariance of *DM* under a local symmetry.) Given an image I, a semi-definite network $\mathcal{N}_0$, a definite network $\mathcal{N}_1$, a homomorphism $p: \mathcal{N}_1 \to \mathcal{N}_0$, an embedding token u for $\mathcal{N}_1$, an embedding type v for $\mathcal{N}_0$, and a local symmetry π of $\mathcal{N}_1$ with respect to $\mathcal{N}_0, p, v$, the application of π to p, u produces a homomorphism $p': \mathcal{N}_1 \to \mathcal{N}_0$ and an embedding token u' for $\mathcal{N}_1$.

Then

(i) $DM(I, \mathcal{N}_1, u', v \circ p') = DM(I, \mathcal{N}_1, u, v \circ p)$,

(ii) The symmetry condition holds for $\mathcal{N}_1, u', v \circ p'$ iff it holds for $\mathcal{N}_1, u, v \circ p$.

Proof. We begin by recapitulating the relevant definitions and notation. Define functions a, s by $\forall \sigma \in \Sigma_1 \ \pi(\sigma) = (a(\sigma), s(\sigma))$. Then p' is given by

$$\begin{aligned} &\forall \sigma \in \Sigma_1 \quad p'(\sigma) = a(\sigma)(p(\sigma)) \\ &\forall n \in N_1 \quad p'(n) = a(W_1(n))(p(n)) \\ &\forall h \in H_1 \quad p'(h) = a(W_1(A_1(h)))(p(h)) \\ &\forall e \in E_1 \quad p'(e) = a(W_1(A_1(F_1(e))))(p(e)) \\ &\forall k \in K_1 \quad p'(k) = a(W_1(A_1(F_1(C_1(k)))))(p(k)) \end{aligned} \tag{1}$$

and $u' = u \cdot s'$, where s' is an embedding token for $\mathcal{N}_1$ defined by

$$\begin{aligned} \forall \sigma \in \Sigma_1 \quad & s'(\sigma) = s(\sigma)(p(\sigma)) \\ \forall n \in N_1 \quad & s'(n) = s(W_1(n))(p(n)). \end{aligned} \tag{2}$$

Also, $v = (sub, con, rel, symm, tem, in)$, and, for any $\sigma^\dagger \in \Sigma_1$, $(a(\sigma^\dagger), s(\sigma^\dagger))$ is a symmetry of $\mathcal{N}_0$ with respect to v, so $v \circ a(\sigma^\dagger) = v \cdot s(\sigma^\dagger)$; in terms of components this means

$$\begin{aligned} &(sub^\ddagger, con \circ a(\sigma^\dagger), rel \circ a(\sigma^\dagger), symm \circ a(\sigma^\dagger), tem \circ a(\sigma^\dagger), in \circ a(\sigma^\dagger)) = v \circ a(\sigma^\dagger) \\ &= v \cdot s(\sigma^\dagger) = (sub^\dagger, con^\dagger, rel^\dagger, symm, tem^\dagger, in^\dagger) \end{aligned} \tag{3}$$

where $sub^\ddagger, sub^\dagger, con^\dagger, rel^\dagger, tem^\dagger, in^\dagger$ are defined by

$$\begin{aligned} \forall \sigma, \sigma^* \in \Sigma_0 \quad (G_0(\sigma, \sigma^*) \;\Rightarrow\; & sub^\ddagger(\sigma, \sigma^*) = sub(a(\sigma^\dagger)(\sigma), a(\sigma^\dagger)(\sigma^*))) \\ \forall \sigma, \sigma^* \in \Sigma_0 \quad (G_0(\sigma, \sigma^*) \;\Rightarrow\; & sub^\dagger(\sigma, \sigma^*) = (s(\sigma^\dagger)(\sigma))^{-1} \cdot sub(\sigma, \sigma^*) \cdot s(\sigma^\dagger)(\sigma^*)) \\ \forall n \in N_0 \quad & con^\dagger(n) = (s(\sigma^\dagger)(W_0(n)))^{-1} \cdot con(n) \cdot s(\sigma^\dagger)(n) \\ \forall e \in E_0 \quad & rel^\dagger(e) = (s(\sigma^\dagger)(A_0(S_0(e))))^{-1} \cdot rel(e) \cdot s(\sigma^\dagger)(A_0(F_0(e))) \\ \forall \sigma \in \Sigma_0 \quad & tem^\dagger(\sigma) = tem(\sigma) \circ s(\sigma^\dagger)(\sigma) \\ \forall \sigma \in \Sigma_0 \quad & in^\dagger(\sigma) = (s(\sigma^\dagger)(\sigma))^{-1} \cdot in(\sigma) \cdot s(\sigma^\dagger)(\sigma). \end{aligned} \tag{4}$$

I claim that $v \circ p' = (v \circ p) \cdot s'$. To verify the claim we may write the left-hand side as

$$v \circ p' = (sub', con \circ p', rel \circ p', symm \circ p', tem \circ p', in \circ p')$$

where

$$\forall \sigma, \sigma^* \in \Sigma_1 \quad (G_1(\sigma, \sigma^*) \Rightarrow sub'(\sigma, \sigma^*) = sub(p'(\sigma), p'(\sigma^*))) \tag{5}$$

and the right-hand side as

$$(v \circ p) \cdot s' = (sub'', con'', rel'', symm \circ p, tem'', in'')$$

where

$$\begin{aligned} \forall \sigma, \sigma^* \in \Sigma_1 \quad (G_1(\sigma, \sigma^*) \;\Rightarrow\; & sub''(\sigma, \sigma^*) = s'(\sigma)^{-1} \cdot sub(p(\sigma), p(\sigma^*)) \cdot s'(\sigma^*)) \\ \forall n \in N_1 \quad & con''(n) = s'(W_1(n))^{-1} \cdot con(p(n)) \cdot s'(n) \\ \forall e \in E_1 \quad & rel''(e) = s'(A_1(S_1(e)))^{-1} \cdot rel(p(e)) \cdot s'(A_1(F_1(e))) \\ \forall \sigma \in \Sigma_1 \quad & tem''(\sigma) = tem(p(\sigma)) \circ s'(\sigma) \\ \forall \sigma \in \Sigma_1 \quad & in''(\sigma) = s'(\sigma)^{-1} \cdot in(p(\sigma)) \cdot s'(\sigma). \end{aligned} \tag{6}$$

Now we can verify the claim by comparing the six components of $v \circ p'$ and $(v \circ p) \cdot s'$.

$$\begin{aligned} &\forall \sigma, \sigma^* \in \Sigma_1 \quad (G_1(\sigma, \sigma^*) \Rightarrow \\ &\quad sub'(\sigma, \sigma^*) = sub(p'(\sigma), p'(\sigma^*)) && \text{by (5)} \\ &\quad = sub\left(a(\sigma)(p(\sigma)), a(\sigma^*)(p(\sigma^*))\right) && \text{by (1)} \\ &\quad = sub\left(a(\sigma)(p(\sigma)), a(\sigma)(p(\sigma^*))\right) && \text{since } \pi \text{ is a local symmetry} \\ &\quad = sub^\ddagger(p(\sigma), p(\sigma^*)) && \text{by (4) with } \sigma^\dagger = \sigma \\ &\quad = sub^\dagger(p(\sigma), p(\sigma^*)) && \text{by (3)} \\ &\quad = \left(s(\sigma)(p(\sigma))\right)^{-1} \cdot sub(p(\sigma), p(\sigma^*)) \cdot s(\sigma)(p(\sigma^*)) && \text{by (4)} \\ &\quad = \left(s(\sigma)(p(\sigma))\right)^{-1} \cdot sub(p(\sigma), p(\sigma^*)) \cdot s(\sigma^*)(p(\sigma^*)) && \text{since } \pi \text{ is a local symmetry} \\ &\quad = s'(\sigma)^{-1} \cdot sub(p(\sigma), p(\sigma^*)) \cdot s'(\sigma^*) && \text{by (2)} \\ &\quad = sub''(\sigma, \sigma^*)) && \text{by (6)} \end{aligned}$$

$$\begin{aligned}
\forall n \in N_1 \quad (con \circ p')(n) &= con(p'(n)) = con(a(W_1(n))(p(n))) && \text{by (1)}\\
&= con^{\dagger}(p(n)) && \text{by (3) with } \sigma^{\dagger} = W_1(n)\\
&= \big(s(W_1(n))(W_0(p(n)))\big)^{-1} \cdot con(p(n)) \cdot s(W_1(n))(p(n)) && \text{by (4)}\\
&= \big(s(W_1(n))(p(W_1(n)))\big)^{-1} \cdot con(p(n)) \cdot s(W_1(n))(p(n)) && \text{since } W_0 \circ p = p \circ W_1\\
&= s'(W_1(n))^{-1} \cdot con(p(n)) \cdot s'(n) && \text{by (2)}\\
&= con''(n) && \text{by (6)}
\end{aligned}$$

$$\begin{aligned}
\forall e \in E_1 \quad (rel \circ p')(e) &= rel(p'(e)) = rel\big(a(W_1(A_1(F_1(e))))(p(e))\big) && \text{by (1)}\\
&= rel^{\dagger}(p(e)) && \text{by (3) with } \sigma^{\dagger} = W_1(A_1(F_1(e)))\\
&= \big(s(\sigma^{\dagger})(A_0(S_0(p(e))))\big)^{-1} \cdot rel(p(e)) \cdot s(\sigma^{\dagger})(A_0(F_0(p(e)))) && \text{by (4)}\\
&= \big(s(\sigma^{\dagger})(p(A_1(S_1(e))))\big)^{-1} \cdot rel(p(e)) \cdot s(\sigma^{\dagger})(p(A_1(F_1(e)))) && \text{since } p \text{ is a homomorphism}\\
&= \big(s(W_1(A_1(S_1(e))))(p(A_1(S_1(e))))\big)^{-1} \cdot rel(p(e)) \cdot s(\sigma^{\dagger})(p(A_1(F_1(e)))) \\
& && \text{since } \pi \text{ is a local symmetry}\\
&= s'(A_1(S_1(e)))^{-1} \cdot rel(p(e)) \cdot s'(A_1(F_1(e))) && \text{by (2)}\\
&= rel''(e) && \text{by (6)}
\end{aligned}$$

$$\begin{aligned}
\forall \sigma \in \Sigma_1 \quad (symm \circ p')(\sigma) &= symm(p'(\sigma)) = symm(a(\sigma)(p(\sigma))) && \text{by (1)}\\
&= symm(p(\sigma)) && \text{by (3), with } \sigma^{\dagger} = \sigma\\
&= (symm \circ p)(\sigma)
\end{aligned}$$

$$\begin{aligned}
\forall \sigma \in \Sigma_1 \quad (tem \circ p')(\sigma) &= tem(p'(\sigma)) = tem(a(\sigma)(p(\sigma))) && \text{by (1)}\\
&= tem^{\dagger}(p(\sigma)) && \text{by (3) with } \sigma^{\dagger} = \sigma\\
&= tem(p(\sigma)) \circ s(\sigma)(p(\sigma)) && \text{by (4)}\\
&= tem(p(\sigma)) \circ s'(\sigma) && \text{by (2)}\\
&= tem''(\sigma) && \text{by (6)}
\end{aligned}$$

$$\begin{aligned}
\forall \sigma \in \Sigma_1 \quad (in \circ p')(\sigma) &= in(p'(\sigma)) = in(a(\sigma)(p(\sigma))) && \text{by (1)}\\
&= in^{\dagger}(p(\sigma)) && \text{by (3) with } \sigma^{\dagger} = \sigma\\
&= (s(\sigma)(p(\sigma)))^{-1} \cdot in(p(\sigma)) \cdot s(\sigma)(p(\sigma)) && \text{by (4)}\\
&= s'(\sigma)^{-1} \cdot in(p(\sigma)) \cdot s'(\sigma) && \text{by (2)}\\
&= in''(\sigma) && \text{by (6).}
\end{aligned}$$

This completes the verification of the claim that $v \circ p' = (v \circ p) \cdot s'$.

Now we are in a position to prove part (i) of the theorem:

$$\begin{aligned}
DM(I, \mathcal{N}_1, u', v \circ p') &= DM(I, \mathcal{N}_1, u \cdot s', v \circ p')\\
&= DM(I, \mathcal{N}_1, u \cdot s', (v \circ p) \cdot s') && \text{by the claim}\\
&= DM(I, \mathcal{N}_1, u, v \circ p) && \text{by theorem 28(i).}
\end{aligned}$$

For part (ii), consider any $n \in N_1$. Recall that, for any $\sigma^\dagger \in \Sigma_1$, $symm \circ a(\sigma^\dagger) = symm$, by (3), so, taking $\sigma^\dagger = P_1(n)$, $symm(a(P_1(n))(p(P_1(n)))) = symm(p(P_1(n)))$ and hence

$$symm(p'(P_1(n))) = symm(p(P_1(n))). \tag{7}$$

Secondly, since $(a(P_1(n)), s(P_1(n)))$ is a symmetry of $\mathcal{N}_0$ with respect to v,

$$s(P_1(n))(p(P_1(n))) \in symm(p(P_1(n))) \tag{8}$$

and, since $(a(W_1(n)), s(W_1(n)))$ is a symmetry of $\mathcal{N}_0$ with respect to v,

$$s(W_1(n))(p(n)) \in symm(P_0(p(n))) = symm(p(P_1(n))). \tag{9}$$

Now, suppose that the symmetry condition holds for $\mathcal{N}_1, u, v \circ p$. Then, for any $n \in N_1$, $u(P_1(n))^{-1} \cdot u(n) \in symm(p(P_1(n)))$, so

$$\begin{aligned} u'(P_1(n))^{-1} \cdot u'(n) &= \left(s(P_1(n))(p(P_1(n)))\right)^{-1} \cdot u(P_1(n))^{-1} \cdot u(n) \cdot s(W_1(n))(p(n)) \\ &\in symm(p(P_1(n))) && \text{by (8) and (9)} \\ &= symm(p'(P_1(n))) && \text{by (7)} \end{aligned}$$

so the symmetry condition for $\mathcal{N}_1, u', v \circ p'$ holds. Conversely, suppose that the symmetry condition for $\mathcal{N}_1, u', v \circ p'$ holds. Then, for any $n \in N_1$, $u'(P_1(n))^{-1} \cdot u'(n) \in symm(p'(P_1(n))) = symm(p(P_1(n)))$, by (7), so from

$$u'(P_1(n))^{-1} \cdot u'(n) = \left(s(P_1(n))(p(P_1(n)))\right)^{-1} \cdot u(P_1(n))^{-1} \cdot u(n) \cdot s(W_1(n))(p(n))$$

we have

$$\begin{aligned} u(P_1(n))^{-1} \cdot u(n) &= s(P_1(n))(p(P_1(n))) \cdot u'(P_1(n))^{-1} \cdot u'(n) \cdot \left(s(W_1(n))(p(n))\right)^{-1} \\ &\in symm(p(P_1(n))) && \text{by (8) and (9)} \end{aligned}$$

so the symmetry condition for $\mathcal{N}_1, u, v \circ p$ holds. ∎

Theorem 30. (Affine invariance of the ρ function.) For any template T, any image I, and any affine transformations g, g',

$$\rho_{I \circ g^{-1}, T, sat \circ g^{-1}}(g \cdot g') = |\,det(\overline{g})|^{1-k}\, \rho_{I,T,sat}(g').$$

(Recall that $\overline{g}$ is the matrix representation of g.)

Proof. Recall the definition

$$\rho_{I,T,sat}(g) = |det(\overline{g})|^{1-k} \int w(sat(g(u)))\, T(u)\, (I(g(u)) - I_0)\, \mathrm{d}^2 u.$$

Now,

$$\begin{aligned} \rho_{I \circ g^{-1}, T, sat \circ g^{-1}}(g \cdot g') &= |det(\overline{g \cdot g'})|^{1-k} \int w((sat \circ g^{-1})((g \cdot g')(u)))\, T(u)\, ((I \circ g^{-1})((g \cdot g')(u)) - I_0)\, \mathrm{d}^2 u \\ &= |det(\overline{g})|^{1-k}\, |det(\overline{g'})|^{1-k} \int w(sat(g'(u)))\, T(u)\, (I(g'(u)) - I_0)\, \mathrm{d}^2 u \\ &= |det(\overline{g})|^{1-k}\, \rho_{I,T,sat}(g'). \end{aligned}$$

▮

Theorem 31. (Affine invariance of DM.) For any affine g such that $|det(\overline{g})| = 1$, any image I, any definite network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, any embedding token u for $\mathcal{N}$, and any embedding type v for $\mathcal{N}$, define a new, affine-transformed image $I' = I \circ g^{-1}$, and a new, affine-transformed embedding token u' for $\mathcal{N}$ by $\forall x \in \Sigma \cup N\ u'(x) = g \cdot u(x)$. Then

$$DM(I', \mathcal{N}, u', v) = DM(I, \mathcal{N}, u, v).$$

Proof. Let sat be the definite saturation function calculated using the embedding token u. Then the saturation function calculated using u' is $sat' = sat \circ g^{-1}$, by the definition of sat in §4.4. By theorem 30

$$\forall \sigma \in \Sigma \quad \rho_{I', tem(\sigma), sat'}(u'(\sigma)) = \rho_{I, tem(\sigma), sat}(u(\sigma)).$$

Also,

$$\forall \sigma, \sigma^* \in \Sigma \quad u'(\sigma)^{-1} \cdot u'(\sigma^*) = u(\sigma)^{-1} \cdot g^{-1} \cdot g \cdot u(\sigma^*) = u(\sigma)^{-1} \cdot u(\sigma^*)$$

and similarly

$$\forall n \in N \quad u'(W(n))^{-1} \cdot u'(n) = u(W(n))^{-1} \cdot u(n)$$
$$\forall e \in E \quad u'(A(S(e)))^{-1} \cdot u'(A(F(e))) = u(A(S(e)))^{-1} \cdot u(A(F(e)))$$
$$\forall e \in E \quad u'(W(A(S(e))))^{-1} \cdot u'(W(A(F(e)))) = u(W(A(S(e))))^{-1} \cdot u(W(A(F(e))))$$

Thus the each term of $DM(I', \mathcal{N}, u', v)$ equals the corresponding term of $DM(I, \mathcal{N}, u, v)$. ▮

4.7 The recognition problem – definitive statement

With the help of the theory developed so far we can now give a definitive statement of the recognition problem, taking account of embeddings.

Given a semi-definite network $\mathcal{N}_0$ (representing a grammar), an embedding type v for $\mathcal{N}_0$, and an image I, the recognition problem is to construct a definite network $\mathcal{N}_1$ (representing a pattern), a homomorphism $p: \mathcal{N}_1 \to \mathcal{N}_0$, and an embedding token u for $\mathcal{N}_1$, maximising $DM(I, \mathcal{N}_1, u, v \circ p)$, subject to the symmetry condition for $\mathcal{N}_1, u, v \circ p$.

5. Inclusion Functions

5.1 Introduction

During recognition, when many competing grammatical possibilities are being considered in parallel in the same pattern, some possibilities are evaluated as better than others. That is, each symbol token σ in the pattern has associated with it a real number $i(\sigma)\in[0,1]$ (called an *inclusion value*) indicating the algorithm's degree of confidence that σ should be present in the final pattern. If $i(\sigma)=1$ then the algorithm has decided definitely to include it in the final pattern; if $i(\sigma)=0$ the algorithm has decided definitely to prune it (but hasn't yet done so); intermediate values indicate uncertainty about whether to keep it. Similarly each node token n and edge token e has an inclusion value $i(n)$ or $i(e)$ indicating the algorithm's degree of confidence that it should be included in the final pattern. (Hooks and facets do not need their own inclusion values as they stand or fall with the symbol, node or edge they belong to.)

The function i is called an *inclusion function* and is accompanied by a second inclusion function j, defined below.

By the end of the recognition process, each inclusion value have been driven to the extremes, 0 or 1, and the symbols, nodes and edges with inclusion value 0 have all been pruned.

5.2 Definition of inclusion functions

Definition. A pair of *inclusion functions* on a network $\mathcal{N}=(\Sigma,N,H,E,K,W,P,A,F,S,C,G)$ is defined as (i,j), where $i\colon \Sigma\cup N\cup E\to[0,1]$ and $j\colon \Sigma\cup N\cup H\cup(K\backslash dom(G))\to[0,1]$, such that

$$\forall\sigma\in\Sigma\quad i(\sigma)=j(\sigma)\;+\sum_{n\in P^{-1}(\{\sigma\})}(1-g_n)i(n),\qquad\text{where } g_n=\left|\{\,n^*\in N\mid G(n^*,n)\,\}\right| \tag{1}$$

$$\forall n\in N\quad i(W(n))=j(n)+i(n) \tag{2}$$

$$\forall h\in H\quad i(A(h))=j(h)\;+\sum_{e\in F^{-1}(\{h\})}i(e)\;+\sum_{e\in S^{-1}(\{h\})}i(e) \tag{3}$$

$$\forall k\in K\backslash dom(G)\quad i(C(k))=j(k)+\sum_{k^*\mid G(k,k^*)}i(C(k^*)). \tag{4}$$

(This is a slightly modified version of the definition in (2004, §7.2), taking account of subsymbols. The summation notation in (4) means a sum over all k^* such that $G(k,k^*)$.)

This definition may be interpreted informally as follows. (In the following explanation I shall use the future tense to state conditions that will hold at the end of recognition when the pattern is definite, and the present tense for what holds during recognition when the pattern is indefinite.)

Line (1). At the end of recognition each symbol σ will be either a 'bare' symbol (with $P^{-1}(\{\sigma\})=\emptyset$) or a part of one larger symbol (with $|P^{-1}(\{\sigma\})|=1$). Hence the nodes presently in $P^{-1}(\{\sigma\})$ are in competition with one another; at most one will survive to the end. $i(\sigma)$ is interpreted as the degree of confidence in σ, $j(\sigma)$ is the degree of confidence that σ will be a bare symbol, and $i(n)$ is the degree of confidence in n. An exception to this competition is that if n is a subnode of n^* (i.e., $G(n^*,n)$) then they may both be correct; the $1-g_n$ factor allows for this co-existence.

Line (2). $i(W(n))$ is the degree of confidence in $W(n)$, $j(n)$ is the degree of confidence that $W(n)$ will survive to the end but n will not, and $i(n)$ is the degree of confidence in n.

Line (3) expresses the fact that each hook h will have a single edge incident to it (i.e., $|F^{-1}(\{h\})|+|S^{-1}(\{h\})|=1$); hence the edges presently in $F^{-1}(\{h\})\cup S^{-1}(\{h\})$ are in competition with one another. Thus $i(A(h))$ is the degree of confidence in the node or symbol $A(h)$, $i(e)$ is the degree of confidence in e, and $j(h)$ is the degree of confidence that $A(h)$ will survive to the end but that none of its present edges will (i.e., the correct edge has yet to be created).

Line (4) expresses the fact that each facet k that is not itself a sub-facet will be glued to a single sub-facet. Hence if k is presently glued to several sub-facets then they are in competition with one another; $i(C(k^*))$ is the degree of confidence in the edge $C(k^*)$, $j(k)$ is the degree of confidence that $C(k)$ is correct but that none of k's present sub-facets is.

Theorem 32. If $\mathcal{N}=(\Sigma,N,H,E,K,W,P,A,F,S,C,G)$ is a definite network and (i,j) is a pair of inclusion functions on $\mathcal{N}$ satisfying $\forall x\in\Sigma\cup N\cup E\ i(x)=1$, then

$$\forall\sigma\in\Sigma\ j(\sigma)=\begin{cases}0 & \text{if } \sigma\in P(N)\\ 1 & \text{otherwise}\end{cases},\quad \forall n\in N\ j(n)=0,\quad \forall h\in H\ j(h)=0,\quad \forall k\in K\setminus\mathrm{dom}(G)\ j(k)=0.$$

Proof. For any $\sigma\in\Sigma$,

$$i(\sigma)=j(\sigma)\ +\sum_{n\in P^{-1}(\{\sigma\})}(1-g_n)i(n)$$

where $g_n=\big|\{\,n^*\in N\mid G(n^*,n)\,\}\big|$, so

$$1=j(\sigma)+|P^{-1}(\{\sigma\})|-|G_N\circ id_{P^{-1}(\{\sigma\})}|.$$

This implies $j(\sigma)\in\mathbb{Z}$, and since also $j(\sigma)\in[0,1]$ we have $j(\sigma)\in\{0,1\}$. Now, for any $\sigma\in\Sigma$, if $\sigma\notin P(N)$ then $P^{-1}(\{\sigma\})=\emptyset$, which immediately gives $|P^{-1}(\{\sigma\})|-|G_N\circ id_{P^{-1}(\{\sigma\})}|=0$ and so $j(\sigma)=1$. Conversely, if $j(\sigma)=1$ then $|P^{-1}(\{\sigma\})|-|G_N\circ id_{P^{-1}(\{\sigma\})}|=0$, so by theorem 13(ii) (applied to N,Σ,P,G_N) $P^{-1}(\{\sigma\})=\emptyset$, so $\sigma\notin P(N)$. This shows that $j(\sigma)=1$ if $\sigma\notin P(N)$ and $j(\sigma)=0$ if $\sigma\in P(N)$.

For any $n\in N$,

$$i(\sigma)=j(n)+i(n)$$

and hence $j(n)=0$.

For any $h\in H$,

$$i(A(h))=j(h)\ +\sum_{e\in F^{-1}(\{h\})}i(e)\ +\sum_{e\in S^{-1}(\{h\})}i(e)$$

and, thanks to the sum diagram $E\xrightarrow{F}H\xleftarrow{S}E$, there is a unique $e\in E$ such that either $F(e)=h$ or $S(e)=h$; so $j(h)=0$.

For any $k\in K\setminus\mathrm{dom}(G)$,

$$i(C(k))=j(k)+\sum_{k^*|G(k,k^*)}i(C(k^*)).$$

By the definiteness condition $id_K=G_K\circ G_K^{-1}\cup G_K^{-1}\circ G_K$, there exists $k^*\in K$ such that $G(k,k^*)$ (otherwise $(G_K\circ G_K^{-1}\cup G_K^{-1}\circ G_K)(k,k)$ would not hold); moreover, this k^* is unique (for if there were a second, k^{**}, then $(G_K\circ G_K^{-1}\cup G_K^{-1}\circ G_K)(k^*,k^{**})$ would hold). Thus $j(k)=0$. ∎

5.3 The recognition process

With the help of the theory developed so far we can give an initial description of the recognition process. It constructs a pattern $\mathcal{N}_1$, a homomorphism $p{:}\mathcal{N}_1 \to \mathcal{N}_0$ (where $\mathcal{N}_0$ is the grammar), and a pair of inclusion functions (i,j) on $\mathcal{N}_1$, by a sequence of structural operations on the pattern.

The recognition process must ensure that, at the end, the following conditions hold.

- $\forall x \in \Sigma_1 \cup N_1 \cup E_1 \; i(x) = 1$,
- $\mathcal{N}_1$ is definite.

6. The Indefinite Match Function

6.1 Introduction

In section 4 we constructed the definite match function DM, which measures how well a definite pattern (with an embedding token) matches the image, the grammar and embedding type. Now we shall generalise it to the *indefinite match function* IM, which applies to patterns that are not necessarily definite, with inclusion functions. The affine invariance theorems of §4.6 are generalised to IM.

The recognition process attempts to maximise the value of IM during recognition, so that the final, definite pattern will maximise DM.

6.2 Template matching

The template matching function of §4.4 is generalised as follows.

Suppose we are given a network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, an embedding token u for $\mathcal{N}$, a function $tem\colon \Sigma \to Tem$, and a pair (i,j) of inclusion functions on $\mathcal{N}$. The *(indefinite) saturation function*, $sat\colon \mathbb{R}^2 \to \mathbb{R}$, is defined by

$$\forall x \in \mathbb{R}^2 \quad sat(x) = \sum_{\sigma\in\Sigma} (1 - g_\sigma)\, i(\sigma)\, tem(\sigma)(u(\sigma)^{-1}(x))$$

where $g_\sigma = \big|\{\, \sigma^* \in \Sigma \mid G(\sigma^*, \sigma)\,\}\big|$.

In the special case where the inclusion functions satisfy $\forall\sigma\in\Sigma\ i(\sigma) = 1$, as at the end of recognition, the definition of sat reduces to the definite saturation function defined in §4.4.

The correlation function $\rho_{I,T,sat}$ between an image I and a template T is as in §4.4, except for using the indefinite saturation function,

$$\begin{aligned}\rho_{I,T,sat}(g) &= |det(\overline{g})|^{1-k} \int w(sat(g(u)))\, T(u)\, (I(g(u)) - I_0)\, \mathrm{d}^2u \\ &= |\, det(\overline{g})|^{-k} \int w(sat(x))\, T(g^{-1}(x))\, (I(x) - I_0)\, \mathrm{d}^2x\end{aligned}$$

6.3 The indefinite match function

The *indefinite match* function *IM* measures how well a (not necessarily definite) network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ matches an image I, given an embedding token u for $\mathcal{N}$, an embedding type $v = (sub, con, rel, symm, tem, in)$ for $\mathcal{N}$, a pair of inclusion functions (i,j) on $\mathcal{N}$, and a function $B\colon \Sigma \cup H \cup (K \setminus \mathrm{dom}(G)) \to [0,\infty)$. It is defined by

$$\begin{aligned}
&IM(I,\mathcal{N},u,v,i,j,B) = \\
&\quad \sum_{\sigma\in\Sigma} \big(i(\sigma)\rho_{I,tem(\sigma),sat}(u(\sigma)) - j(\sigma)B(\sigma)\big) - \sum_{(\sigma,\sigma^*)|G(\sigma,\sigma^*)} i(\sigma^*)E_{sub(\sigma,\sigma^*)}(u(\sigma)^{-1}\cdot u(\sigma^*)) \\
&\quad - \sum_{n\in N} i(n)E_{con(n)}(u(W(n))^{-1}\cdot u(n)) - \sum_{h\in H} j(h)B(h) - \sum_{k\in K\setminus \mathrm{dom}(G)} j(k)B(k) \\
&\quad - \sum_{e\in E} i(e)\big(E_{rel(e)}(u(A(S(e)))^{-1}\cdot u(A(F(e)))) + E_{in(W(A(F(e))))}(u(W(A(S(e))))^{-1}\cdot u(W(A(F(e)))))\big)
\end{aligned}$$

where *sat* is the indefinite saturation function. Observe that the B function imposes a penalty for each 'bare' symbol, hook and facet, i.e., each symbol that is not a part of any other symbol ($j(\sigma) = 1$), each hook with no incident edges ($j(h) = 1$), and each facet that should be glued to a sub-facet but is not ($j(k) = 1$). A function $B\colon \Sigma \cup H \cup (K \setminus \mathrm{dom}(G)) \to [0,\infty)$ is called a *bareness function* for $\mathcal{N}$. The final term, $E_{in(W(A(F(e))))}(\cdots)$, has no counterpart in *DM*; it is to penalise edges between two nodes belonging to different wholes (see §9.6).

The following theorem shows that *DM* is a special case of *IM*.

Theorem 33. If
(a) $\mathcal{N}$ is definite,
(b) $\forall x \in \Sigma \cup N \cup E\ i(x) = 1$,
(c) $\forall \sigma \in \Sigma \setminus P(N)\ B(\sigma) = \theta$,

then $IM(I,\mathcal{N},u,v,i,j,B) = DM(I,\mathcal{N},u,v)$.

Proof. By theorem 32,

$$\forall\sigma\in\Sigma\ j(\sigma) = \begin{cases} 0 & \text{if } \sigma \in P(N) \\ 1 & \text{otherwise} \end{cases}, \quad \forall n\in N\ j(n) = 0, \quad \forall h\in H\ j(h) = 0, \quad \forall k\in K\setminus \mathrm{dom}(G)\ j(k) = 0.$$

It follows that each term of $IM(I,\mathcal{N},u,v,i,j,B)$ simplifies to the corresponding term of $DM(I, \mathcal{N},u,v)$ or vanishes; in particular, for each $e\in E$, the term

$$E_{in(W(A(F(e))))}(u(W(A(S(e))))^{-1}\cdot u(W(A(F(e)))))$$

vanishes since $W(A(S(e))) = W(A(F(e)))$ by the definiteness conditions. ∎

In the recognition process an embedding token u for $\mathcal{N}_1$ is constructed, along with $\mathcal{N}_1$ and p. An embedding type $v \circ p$ is induced using p from the given embedding type v for $\mathcal{N}_0$. Throughout recognition we measure how well u matches $v \circ p$ by calculating $IM(I,\mathcal{N}_1,u,v \circ p,i,j,B)$, and we seek to maximise this by continually adjusting u, subject to the symmetry condition for $\mathcal{N}_1,u,v\circ p$.

6.4 Invariance theorems

I shall generalise the invariance theorems of §4.6 from DM to IM.

Theorem 34. (Lemma 1 of (2004, §7.3): invariance under affine transformation of the symbols' internal frames of reference.) For any image I, network $\mathcal{N}$, embedding tokens u,u' for $\mathcal{N}$, embedding type v for $\mathcal{N}$, inclusion functions i,j on $\mathcal{N}$, and bareness function B for $\mathcal{N}$,

$$IM(I,\mathcal{N},u\cdot u',v\cdot u',i,j,B) = IM(I,\mathcal{N},u,v,i,j,B),$$

provided $u'\circ W\circ A\circ F = u'\circ W\circ A\circ S$.

Proof. As in theorem 28, each term of $IM(I,\mathcal{N},u\cdot u',v\cdot u',i,j,B)$ equals the corresponding term of $IM(I,\mathcal{N},u,v,i,j,B)$. ▮

Theorem 35. (Invariance of the match functions under a local symmetry.) Given an image I, a semi-definite network $\mathcal{N}_0$, a network $\mathcal{N}_1$, a homomorphism $p{:}\mathcal{N}_1\to\mathcal{N}_0$, an embedding token u for $\mathcal{N}_1$, an embedding type v for $\mathcal{N}_0$, and a local symmetry π of $\mathcal{N}_1$ with respect to $\mathcal{N}_0,p,v$, the application of π to p,u produces a homomorphism $p'{:}\mathcal{N}_1\to\mathcal{N}_0$ and an embedding token u' for $\mathcal{N}_1$.

(i) For any inclusion functions i,j and bareness function B for $\mathcal{N}_1$, $IM(I,\mathcal{N}_1,u',v\circ p',i,j,B)$ $= IM(I,\mathcal{N}_1,u,v\circ p,i,j,B)$.

(ii) The symmetry condition holds for $\mathcal{N}_1,u',v\circ p'$ iff it holds for $\mathcal{N}_1,u,v\circ p$.

Proof. The argument of theorem 29 generalises:

$$\begin{aligned} IM(I,\mathcal{N}_1,u',v\circ p',i,j,B) &= IM(I,\mathcal{N}_1,u\cdot s',(v\circ p)\cdot s',i,j,B) && \text{since } v\circ p' = (v\circ p)\cdot s' \\ &= IM(I,\mathcal{N}_1,u,v\circ p,i,j,B) && \text{by theorem 34.} \end{aligned}$$

To justify this use of theorem 34 we must check that $s'\circ W_1\circ A_1\circ F_1 = s'\circ W_1\circ A_1\circ S_1$. For any $e\in E_1$,

$$\begin{aligned} s'(W_1(A_1(F_1(e)))) &= s(W_1(A_1(F_1(e))))(p(W_1(A_1(F_1(e))))) \\ &= s(W_1(A_1(F_1(e))))(W_0(A_0(F_0(p(e))))) && \text{since } p \text{ is a homomorphism} \\ &= s(W_1(A_1(F_1(e))))(W_0(A_0(S_0(p(e))))) && \text{since } \mathcal{N}_0 \text{ is semi-definite} \\ &= s(W_1(A_1(S_1(e))))(W_0(A_0(S_0(p(e))))) && \text{since } \pi \text{ is a local symmetry} \\ &= s(W_1(A_1(S_1(e))))(p(W_1(A_1(S_1(e))))) && \text{since } p \text{ is a homomorphism} \\ &= s'(W_1(A_1(S_1(e)))) \end{aligned}$$

as required.

For part (ii), the proof in theorem 29 applies (it does not require $\mathcal{N}_1$ to be definite). ▮

Theorem 36. (Affine invariance of the ρ function.) For any template T, any image I, and any affine transformations g,g',

$$\rho_{I\circ g^{-1},T,sat\circ g^{-1}}(g\cdot g') = |\,det(\overline{g})|^{1-k}\,\rho_{I,T,sat}(g').$$

(Recall that $\overline{g}$ is the matrix representation of g.)

Proof. This is identical to theorem 30, except using the indefinite saturation function. The same proof applies. ▮

Theorem 37. (Affine invariance of IM.) For any affine g such that $|det(\overline{g})| = 1$, any image I, any network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, any embedding token u for $\mathcal{N}$, and any embedding type v for $\mathcal{N}$, define a new, affine-transformed image $I' = I \circ g^{-1}$, and a new, affine-transformed embedding token u' for $\mathcal{N}$ by $\forall x \in \Sigma \cup N \; u'(x) = g \cdot u(x)$.

For any inclusion functions i,j and bareness function B for $\mathcal{N}$,

$$IM(I', \mathcal{N}, u', v, i, j, B) = IM(I, \mathcal{N}, u, v, i, j, B).$$

(The condition $|det(\overline{g})| = 1$ may be dropped if $k = 1$.)

Proof. Let sat be the indefinite saturation function calculated using the embedding token u.

As in the proof of theorem 31, each term of $IM(I', \mathcal{N}, u', v, i, j, B)$ equals the corresponding term of $IM(I, \mathcal{N}, u, v, i, j, B)$. ∎

6.5 Adjustment of the embedding token

During recognition the embeddings of all the symbols and nodes are continually adjusted to maximise $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$ (subject to the symmetry condition). This is done by a process of gradient ascent, as defined in (2004, §§7.6–7.8), but with the calculation of the derivative of the match function in §7.6 augmented to allow for the new terms in the match function. To be precise, theorem 3 of §7.6 is amended as follows. We make a small change in the embedding token from u to $u \cdot \Delta u$, where

$$\forall \sigma \in \Sigma \quad \Delta u(\sigma) = \exp(\varepsilon V_\sigma)$$
$$\forall n \in N \quad \Delta u(n) = \exp(\varepsilon V_n)$$

where the increments $V_\sigma, V_n \in \mathcal{A}$ are linked by the symmetry condition; then the value of IM changes to

$$IM(I, \mathcal{N}, u \cdot \Delta u, v, i, j, B) = IM(I, \mathcal{N}, u, v, i, j, B) + \varepsilon \sum_{\sigma \in \Sigma} F_\sigma(V_\sigma) + o(\varepsilon)$$

where, for each $\sigma \in \Sigma$,

$$\begin{aligned} &F_\sigma = F^0_\sigma + F^1_\sigma + F^2_\sigma + F^3_\sigma + F^4_\sigma \\ &F^0_\sigma, F^1_\sigma, F^2_\sigma, F^3_\sigma \text{ are as before} \\ &F^4_\sigma = -\sum_{\sigma^* | G(\sigma, \sigma^*)} Ad(Sub^{-1}_{\sigma,\sigma^*})^\dagger (F_{\sigma,\sigma^*}) + \sum_{\sigma^* | G(\sigma^*, \sigma)} F_{\sigma^*,\sigma} \end{aligned}$$

and, for each $\sigma, \sigma^* \in \Sigma$ such that $G(\sigma, \sigma^*)$,

$$\begin{aligned} F_{\sigma,\sigma^*} &= -i(\sigma^*) E_{sub(\sigma,\sigma^*)*}(Sub_{\sigma,\sigma^*}) \\ Sub_{\sigma,\sigma^*} &= u(\sigma)^{-1} \cdot u(\sigma^*). \end{aligned}$$

The theorem is proved by the same methods as before.

6.6 The recognition process

During the recognition process $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$ is maximised, and at the end the conditions of theorem 33 are satisfied, so $DM(I, \mathcal{N}_1, u, v \circ p)$ is maximised.

7. How the Inclusion Functions are Determined

7.1 Introduction

This section refines the account of the recognition process by describing how the inclusion functions i and j are determined during recognition. The inclusion functions are determined by a simulated annealing process, governed by a *temperature* parameter that varies across the pattern and with time. Where the structure of the pattern is changing, the temperature is high, and this makes the inclusion functions take on mid-range values, so several alternative interpretations can co-exist in parallel; when structural changes stop, the temperature declines and the inclusion functions are pushed towards 0 or 1, and so a choice is forced between alternatives and the pattern becomes definite. The temperature varies across the pattern, as some parts of the pattern may be very active while other parts have settled down.

7.2 The inclusion vector, $\boldsymbol{i}$

Our first step is to reformulate i,j and the constraints on them in a vector notation in order to emphasise their linear nature (this vector notation is for use only in this section). Given a pair of inclusion functions (i,j) on a network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, we shall convert (i,j) into the *inclusion vector* $\boldsymbol{i}$ on $\mathcal{N}$, with components $\boldsymbol{i}_x$, for all $x \in X$, where

$$X = \Sigma \times \{0,1\} \ \cup \ N \times \{0,1\} \ \cup \ H \ \cup \ E \ \cup \ K \backslash dom(G).$$

Each component of $\boldsymbol{i}$ represents one value of i or j, as follows.

$$\begin{aligned}
\forall \sigma \in \Sigma \quad & \boldsymbol{i}_{(\sigma,0)} = i(\sigma), \quad \boldsymbol{i}_{(\sigma,1)} = j(\sigma) \\
\forall n \in N \quad & \boldsymbol{i}_{(n,0)} = i(n), \quad \boldsymbol{i}_{(n,1)} = j(n) \\
\forall h \in H \quad & \boldsymbol{i}_h = j(h) \\
\forall e \in E \quad & \boldsymbol{i}_e = i(e) \\
\forall k \in K \backslash dom(G) \quad & \boldsymbol{i}_k = j(k)
\end{aligned}$$

The constraints on i and j can be expressed as a set of linear conditions on $\boldsymbol{i}$ of the form

$$\forall y \in Y \quad \boldsymbol{c}^y \cdot \boldsymbol{i} = \sum_{x \in X} \boldsymbol{c}^y_x \boldsymbol{i}_x = 0$$

using a set of vectors $\boldsymbol{c}^y$, for all $y \in Y$, where

$$Y = \Sigma \cup N \cup H \cup K \backslash dom(G)$$

and $\boldsymbol{c}^y$ has components $\boldsymbol{c}^y_x$ for $x \in X$, specified as follows. For each $\sigma \in \Sigma$, the constraint $i(\sigma) = j(\sigma) + \sum_{n \in P^{-1}(\{\sigma\})} (1 - g_n) i(n)$ is expressed as $\boldsymbol{c}^\sigma \cdot \boldsymbol{i} = 0$, where

$$\boldsymbol{c}^\sigma_x = \begin{cases} 1 & \text{if } x = (\sigma, 0) \\ -1 & \text{if } x = (\sigma, 1) \\ g_n - 1 & \text{if } x = (n, 0) \text{ for some } n \in P^{-1}(\{\sigma\}) \\ 0 & \text{otherwise} \end{cases}$$

For each $n \in N$, the condition $i(W(n)) = j(n) + i(n)$ is expressed as $\boldsymbol{c}^n \cdot \boldsymbol{i} = 0$, where

$$\boldsymbol{c}^n_x = \begin{cases} 1 & \text{if } x = (W(n), 0) \\ -1 & \text{if } x = (n, 1) \text{ or } x = (n, 0) \\ 0 & \text{otherwise} \end{cases}$$

For each $h \in H$, the condition $i(A(h)) = j(h) + \sum_{e \in F^{-1}(\{h\})} i(e) + \sum_{e \in S^{-1}(\{h\})} i(e)$ is expressed as $\boldsymbol{c}^h \cdot \boldsymbol{i} = 0$, where

$$\boldsymbol{c}^h_x = \begin{cases} 1 & \text{if } x = (A(h), 0) \\ -1 & \text{if } x = h \text{ or } x \in F^{-1}(\{h\}) \text{ or } x \in S^{-1}(\{h\}) \\ 0 & \text{otherwise.} \end{cases}$$

(Note that this assumes that $F(e) \neq S(e)$ for each edge e.) For each $k \in K \backslash dom(G)$, the constraint $i(C(k)) = j(k) + \sum_{k^* | G(k,k^*)} i(C(k^*))$ is expressed as $\boldsymbol{c}^k \cdot \boldsymbol{i} = 0$, where

$$\boldsymbol{c}^k_x = \begin{cases} 1 & \text{if } x = C(k) \\ -1 & \text{if } x = k \\ -1 & \text{if } x = C(k^*), \text{ for some } k^* \text{ such that } G(k, k^*) \\ 0 & \text{otherwise.} \end{cases}$$

This completes the specification of the $\boldsymbol{c}^y$ vectors.

Note that, using this vector notation, the *IM* function is linear in $\boldsymbol{i}$ (if we disregard the dependence of the saturation function on $\boldsymbol{i}$):

$$\begin{aligned}
&IM(I, \mathcal{N}, u, v, i, j, B) \\
&\quad = \sum_{\sigma \in \Sigma} \big(i(\sigma) \rho_{I', tem(\sigma), sat}(u(\sigma)) - j(\sigma) B(\sigma) \big) - \sum_{(\sigma, \sigma^*) | G(\sigma, \sigma^*)} i(\sigma^*) E_{sub(\sigma, \sigma^*)}(u(\sigma)^{-1} \cdot u(\sigma^*)) \\
&\qquad - \sum_{n \in N} i(n) E_{con(n)}(u(W(n))^{-1} \cdot u(n)) - \sum_{h \in H} j(h) B(h) - \sum_{k \in K \setminus \mathrm{dom}(G)} j(k) B(k) \\
&\qquad - \sum_{e \in E} i(e) \big(E_{rel(e)}(u(A(S(e)))^{-1} \cdot u(A(F(e)))) + E_{in(W(A(F(e))))}(u(W(A(S(e))))^{-1} \cdot u(W(A(F(e))))) \big) \\
&\quad = \boldsymbol{i} \cdot \boldsymbol{m}
\end{aligned}$$

where $\boldsymbol{m}$ is a vector whose components $\boldsymbol{m}_x$ are given by

$$\begin{aligned}
&\forall \sigma \in \Sigma \quad \boldsymbol{m}_{(\sigma,0)} = \rho_{I', tem(\sigma), sat}(u(\sigma)) - \sum_{\sigma^* | G(\sigma^*, \sigma)} E_{sub(\sigma^*, \sigma)}(u(\sigma^*)^{-1} \cdot u(\sigma)), \qquad \boldsymbol{m}_{(\sigma,1)} = -B(\sigma) \\
&\forall n \in N \quad \boldsymbol{m}_{(n,0)} = -E_{con(n)}(u(W(n))^{-1} \cdot u(n)), \qquad \boldsymbol{m}_{(n,1)} = 0 \\
&\forall h \in H \quad \boldsymbol{m}_h = -B(h) \\
&\forall e \in E \quad \boldsymbol{m}_e = -E_{rel(e)}(u(A(S(e)))^{-1} \cdot u(A(F(e)))) - E_{in(W(A(F(e))))}(u(W(A(S(e))))^{-1} \cdot u(W(A(F(e))))) \\
&\forall k \in K \backslash dom(G) \quad \boldsymbol{m}_k = -B(k)
\end{aligned}$$

7.3 How $\boldsymbol{i}$ is determined

Recall that recognition is governed by the expression $IM(I,\mathcal{N}_1,u,v\circ p,i,j,B)$. As shown above, this can be expressed in the form $IM(I,\mathcal{N}_1,u,v\circ p,i,j,B)=\boldsymbol{i}\cdot\boldsymbol{m}$, for a suitable vector $\boldsymbol{m}$. The inclusion vector $\boldsymbol{i}$ is determined by maximising the expression

$$E=\sum_{x\in X}\frac{\boldsymbol{i}_x\boldsymbol{m}_x}{T_x}-\sum_{x\in X}\big(\boldsymbol{i}_x\ln\boldsymbol{i}_x+(1-\boldsymbol{i}_x)\ln(1-\boldsymbol{i}_x)\big)$$

subject to the constraints $\forall y\in Y\ \boldsymbol{c}^y\cdot\boldsymbol{i}=0$, where each T_x is a positive number, known as the *temperature* of x. To be precise, every symbol, node, hook, edge and facet has a temperature, and we define $T_{(\sigma,0)}=T_{(\sigma,1)}=T_\sigma$ and $T_{(n,0)}=T_{(n,1)}=T_n$. (If $\boldsymbol{i}_x=0$ we treat $\boldsymbol{i}_x\ln\boldsymbol{i}_x$ as 0, and if $\boldsymbol{i}_x=1$ we treat $(1-\boldsymbol{i}_x)\ln(1-\boldsymbol{i}_x)$ as 0.)

Using the method of Lagrange multipliers, we have to maximise the Lagrangian

$$L=\sum_{x\in X}\frac{\boldsymbol{i}_x\boldsymbol{m}_x}{T_x}-\sum_{x\in X}\big(\boldsymbol{i}_x\ln\boldsymbol{i}_x+(1-\boldsymbol{i}_x)\ln(1-\boldsymbol{i}_x)\big)+\sum_{y\in Y}\lambda_y\boldsymbol{c}^y\cdot\boldsymbol{i}$$

where λ_y is a Lagrange multiplier, for each $y\in Y$. This gives

$$\forall x\in X\quad 0=\frac{\partial L}{\partial\boldsymbol{i}_x}=\frac{\boldsymbol{m}_x}{T_x}-\ln\Big(\frac{\boldsymbol{i}_x}{1-\boldsymbol{i}_x}\Big)+\sum_{y\in Y}\lambda_y\boldsymbol{c}^y_x.$$

This has the solution

$$\boldsymbol{i}_x=sig\Big(\frac{\boldsymbol{m}_x}{T_x}+\sum_{y\in Y}\lambda_y\boldsymbol{c}^y_x\Big)$$

where the sigmoid function $sig\colon\mathbb{R}\to(0,1)$ is defined by $\forall u\in\mathbb{R}\ sig(u)=\frac{1}{1+e^{-u}}$. By examining the second derivatives it can be determined that this solution is a local maximum. Note that this solution satisfies $\forall x\in X\ \boldsymbol{i}_x\in(0,1)$, and so i and j map into $[0,1]$, as required.

If all the temperatures are high, all $\boldsymbol{i}_x$ tend to take mid-range values. If all temperatures are at the minimum allowed value T_{min}, the $\frac{\boldsymbol{m}_x}{T_x}$ terms will become large, and all $\boldsymbol{i}_x$ are likely to become very close to 0 or 1, in which case we will round them to 0 or 1, giving

$$E=\frac{\sum_{x\in X}\boldsymbol{i}_x\boldsymbol{m}_x}{T_{min}}=\frac{IM(I,\mathcal{N},u,v,i,j,B)}{T_{min}}.$$

7.4 Algorithm for finding $\boldsymbol{i}$

The inclusion vector $\boldsymbol{i}$ is determined above by

$$\forall x\in X\quad\boldsymbol{i}_x=sig\Big(\frac{\boldsymbol{m}_x}{T_x}+\sum_{y\in Y}\lambda_y\boldsymbol{c}^y_x\Big)$$

subject to the constraints $\forall y\in Y\ \boldsymbol{c}^y\cdot\boldsymbol{i}=0$, where the λ_y parameters are unknown. We can split each $\boldsymbol{c}^y$ vector into two vectors $\boldsymbol{c}^{y+}$ and $\boldsymbol{c}^{y-}$ by separating positive and negative components:

$$\forall y\in Y\ \forall x\in X\quad\boldsymbol{c}^{y+}_x=\max(\boldsymbol{c}^y_x,0),\quad\boldsymbol{c}^{y-}_x=\max(-\boldsymbol{c}^y_x,0),$$

so that the constraints may be written as $\forall y\in Y\ \boldsymbol{c}^{y+}\cdot\boldsymbol{i}=\boldsymbol{c}^{y-}\cdot\boldsymbol{i}$. (Note that $\forall y\in Y\ \boldsymbol{c}^{y+},\boldsymbol{c}^{y-}\neq\boldsymbol{0}$.) Also define

$$\forall y\in Y\quad C^{y+}=\max_{x\in X}\boldsymbol{c}^{y+}_x,\quad C^{y-}=\max_{x\in X}\boldsymbol{c}^{y-}_x,\quad C^y=C^{y+}+C^{y-}.$$

The following iterative algorithm seeks values of λ_y satisfying the constraints.

For each $y\in Y$, initialise λ_y to its final value last time this algorithm was run;

repeat

for each $x \in X$, do $\boldsymbol{i}_x := sig\left(\boldsymbol{m}_x/T_x + \sum_{y\in Y} \lambda_y \boldsymbol{c}^y_x\right)$ |

for each $y \in Y$, do $\lambda_y := \lambda_y + \dfrac{1}{C^y} \ln\left(\dfrac{\boldsymbol{c}^{y-} \cdot \boldsymbol{i}}{\boldsymbol{c}^{y+} \cdot \boldsymbol{i}}\right)$

until equilibrium;

for each $x \in X$, if $\boldsymbol{i}_x$ is very close to 0 or 1, then round it to 0 or 1.

In this algorithm the semicolon means sequential composition, the '|' symbol means parallel composition, and the 'for each' loops are parallel loops. This means that all the assignment statements in the 'repeat' loop body may be executed concurrently in any fair order. Note that this loop is by far the most computationally expensive part of the entire recognition process, so the high degree of parallelism is relevant from the point of view of time complexity.

The 'for each $y \in Y$' loop may be expressed more explicitly as

for each $\sigma \in \Sigma$, do $\lambda_\sigma := \lambda_\sigma + \dfrac{1}{C} \ln\left(\dfrac{j(\sigma) + \sum_{n\in P^{-1}(\{\sigma\})}(1-g_n)i(n)}{i(\sigma)}\right)$ |

(where $C = \max(2, \max\{g_n \mid n \in P^{-1}(\{\sigma\})\})$) |

for each $n \in N$, do $\lambda_n := \lambda_n + \frac{1}{2} \ln\left(\dfrac{i(n) + j(n)}{i(W(n))}\right)$ |

for each $h \in H$, do $\lambda_h := \lambda_h + \frac{1}{2} \ln\left(\dfrac{j(h) + \sum_{e\in F^{-1}(\{h\})} i(e) + \sum_{e\in S^{-1}(\{h\})} i(e)}{i(A(h))}\right)$ |

for each $k \in K \backslash dom(G)$, do $\lambda_k := \lambda_k + \frac{1}{2} \ln\left(\dfrac{j(k) + \sum_{k^* \mid G(k,k^*)} i(C(k^*))}{i(C(k))}\right)$.

The rationale for this algorithm is as follows. We begin with a lemma.

Lemma. (a) For any $l \in \mathbb{R}$ and any $\delta > 0$,

$$1 < \frac{sig(l+\delta)}{sig(l)} < e^\delta.$$

(b) For any $l \in \mathbb{R}$ and any $\delta < 0$,

$$1 > \frac{sig(l+\delta)}{sig(l)} > e^\delta.$$

Proof. (a) Since $\delta > 0$, and *sig* and the exponential function are strictly increasing,

$$1 < \frac{sig(l+\delta)}{sig(l)} = \frac{1+e^{-l}}{1+e^{-l-\delta}} = \frac{e^l + 1}{e^{l+\delta}+1} e^\delta < e^\delta.$$

(b) Similarly, for $\delta < 0$,

$$1 > \frac{sig(l+\delta)}{sig(l)} = \frac{1+e^{-l}}{1+e^{-l-\delta}} = \frac{e^l + 1}{e^{l+\delta}+1} e^\delta > e^\delta. \blacksquare$$

Now, consider the effect of applying one of the update rules,

$$\lambda_y := \lambda_y + \delta, \quad \text{where } \delta = \frac{1}{C^y} \ln\left(\frac{\boldsymbol{c}^{y-} \cdot \boldsymbol{i}}{\boldsymbol{c}^{y+} \cdot \boldsymbol{i}}\right),$$

on the constraint $\boldsymbol{c}^{y+} \cdot \boldsymbol{i} = \boldsymbol{c}^{y-} \cdot \boldsymbol{i}$.

We have $\boldsymbol{i}_x = sig(l_x)$, where $l_x = \boldsymbol{m}_x/T_x + \sum_{y\in Y}\lambda_y\boldsymbol{c}^y_x$, for each $x\in X$. The update rule $\lambda_y := \lambda_y + \delta$ causes $\boldsymbol{i}_x$ to be updated to $\boldsymbol{i}'_x = sig(l_x + \delta\boldsymbol{c}^y_x)$, for each $x\in X$.

<u>Case 1: $\boldsymbol{c}^{y+}\cdot\boldsymbol{i} < \boldsymbol{c}^{y-}\cdot\boldsymbol{i}$.</u> Then $\delta > 0$.

For each $x\in X$ such that $\boldsymbol{c}^y_x > 0$, part (a) of the lemma gives

$$1 < \frac{sig(l_x + \delta\boldsymbol{c}^y_x)}{sig(l_x)} < e^{\delta\boldsymbol{c}^y_x} \le e^{\delta C^{y+}}$$

i.e., $1 < \boldsymbol{i}'_x/\boldsymbol{i}_x < e^{\delta C^{y+}}$. Hence, taking a weighted sum over all such x,

$$1 < \frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}} < e^{\delta C^{y+}}.$$

For each $x\in X$ such that $\boldsymbol{c}^y_x < 0$, part (b) of the lemma gives

$$1 > \frac{sig(l_x + \delta\boldsymbol{c}^y_x)}{sig(l_x)} > e^{\delta\boldsymbol{c}^y_x} \ge e^{-\delta C^{y-}}$$

i.e., $1 > \boldsymbol{i}'_x/\boldsymbol{i}_x > e^{-\delta C^{y-}}$. Hence, taking a weighted sum over all such x,

$$1 > \frac{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}} > e^{-\delta C^{y-}}.$$

Dividing the inequalities gives

$$1 < \frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}'/\boldsymbol{c}^{y-}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}/\boldsymbol{c}^{y-}\cdot\boldsymbol{i}} < e^{\delta C^{y}} = \frac{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}}{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}}.$$

Hence

$$\frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}}{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}} < \frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}'} < 1.$$

This shows that, after the update, the constraint $\boldsymbol{c}^{y+}\cdot\boldsymbol{i} = \boldsymbol{c}^{y-}\cdot\boldsymbol{i}$ is closer to being satisfied than it was before.

<u>Case 2: $\boldsymbol{c}^{y+}\cdot\boldsymbol{i} > \boldsymbol{c}^{y-}\cdot\boldsymbol{i}$.</u> Then $\delta < 0$.

For each $x\in X$ such that $\boldsymbol{c}^y_x > 0$, part (b) of the lemma gives

$$1 > \frac{sig(l_x + \delta\boldsymbol{c}^y_x)}{sig(l_x)} > e^{\delta\boldsymbol{c}^y_x} \ge e^{\delta C^{y+}}$$

i.e., $1 > \boldsymbol{i}'_x/\boldsymbol{i}_x > e^{\delta C^{y+}}$. Hence, taking a weighted sum over all such x,

$$1 > \frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}} > e^{\delta C^{y+}}.$$

For each $x\in X$ such that $\boldsymbol{c}^y_x < 0$, part (a) of the lemma gives

$$1 < \frac{sig(l_x + \delta\boldsymbol{c}^y_x)}{sig(l_x)} < e^{\delta\boldsymbol{c}^y_x} \le e^{-\delta C^{y-}}$$

i.e., $1 < \boldsymbol{i}'_x/\boldsymbol{i}_x < e^{-\delta C^{y-}}$. Hence, taking a weighted sum over all such x,

$$1 < \frac{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}} < e^{-\delta C^{y-}}.$$

Dividing the inequalities gives

$$1 > \frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}'/\boldsymbol{c}^{y-}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}/\boldsymbol{c}^{y-}\cdot\boldsymbol{i}} > e^{\delta C^{y}} = \frac{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}}{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}}.$$

Hence

$$\frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}}{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}} > \frac{\boldsymbol{c}^{y+}\cdot\boldsymbol{i}'}{\boldsymbol{c}^{y-}\cdot\boldsymbol{i}'} > 1.$$

This shows that, after the update, the constraint $\boldsymbol{c}^{y+}\cdot\boldsymbol{i} = \boldsymbol{c}^{y-}\cdot\boldsymbol{i}$ is closer to being satisfied than it was before.

<u>Case 3: $\boldsymbol{c}^{y+}\cdot\boldsymbol{i} = \boldsymbol{c}^{y-}\cdot\boldsymbol{i}$.</u> Then $\delta = 0$ and no change is made.

Thus, for each constraint $\boldsymbol{c}^{y+} \cdot \boldsymbol{i} = \boldsymbol{c}^{y-} \cdot \boldsymbol{i}$, the update rule for λ_y moves $\boldsymbol{i}$ closer to satisfying the constraint. This does not imply that $\boldsymbol{i}$ moves closer to satisfying *all* the constraints, still less does it prove convergence to an $\boldsymbol{i}$ satisfying all the constraints, but it does provide some motivation for the update rules used.

7.5 The recognition process

We can now refine further the account of the recognition process. During recognition $\boldsymbol{i}$, and hence (i,j), is determined by maximising

$$E = \sum_{x \in X} \frac{\boldsymbol{i}_x \boldsymbol{m}_x}{T_x} - \sum_{x \in X} \big(\boldsymbol{i}_x \ln \boldsymbol{i}_x + (1 - \boldsymbol{i}_x) \ln(1 - \boldsymbol{i}_x)\big)$$

subject to the constraints $\forall y \in Y\ \boldsymbol{c}^y \cdot \boldsymbol{i} = 0$, given temperatures T_x for each $x \in X$. (Here, $\boldsymbol{i}$ is the inclusion vector corresponding to the pair of inclusion functions (i,j) on the pattern $\mathcal{N}_1$, and $\boldsymbol{m}$ is calculated using the image I, the embedding token u on $\mathcal{N}_1$, and the embedding type $v \circ p$ on $\mathcal{N}_1$.)

In the final stages of recognition, all the temperatures will converge to the minimum allowed temperature $T_{min} > 0$ and so each $\boldsymbol{i}_x$ is likely to approach 0 or 1, and is then rounded to 0 or 1, giving

$$E = \frac{IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)}{T_{min}}.$$

Hence maximising E reduces to maximising $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$. As pointed out in §6.6, this maximises $DM(I, \mathcal{N}_1, u, v \circ p)$ for the final pattern.

8. The Structural Operations

8.1 Introduction

In this section I shall define the *structural operations* by which the pattern is incrementally grown during recognition. There are four kinds:

- pruning operations,
- extension operations,
- merging two symbol tokens,
- partitioning a symbol token into two.

8.2 Pruning operations

Definition. Given a network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ and a pair (i,j) of inclusion functions on $\mathcal{N}$, a *pruning operation* is a transformation from $\mathcal{N}$ to a subnetwork $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$ such that

$$\forall \sigma \in \Sigma \backslash \Sigma' \; i(\sigma) = 0, \quad \forall n \in N \backslash N' \; i(n) = 0, \quad \forall e \in E \backslash E' \; i(e) = 0.$$

A pruning operation is *trivial* iff $\mathcal{N}' = \mathcal{N}$.

Theorem 38. For any pruning operation from $\mathcal{N}$ to $\mathcal{N}'$, given a pair (i,j) of inclusion functions on $\mathcal{N}$,

(i) $\forall \sigma \in \Sigma \backslash \Sigma' \; j(\sigma) = 0, \quad \forall h \in H \backslash H' \; j(h) = 0, \quad \forall k \in (K \backslash \operatorname{dom}(G)) \backslash (K' \backslash \operatorname{dom}(G')) \; j(k) = 0,$

(ii) $(i|_{\Sigma' \cup N' \cup E'}, j|_{\Sigma' \cup N' \cup H' \cup (K' \backslash \operatorname{dom}(G'))})$ is a pair of inclusion functions on $\mathcal{N}'$ (called the *restriction of* (i,j) *to* $\mathcal{N}'$).

Proof. (i) For any $\sigma \in \Sigma \backslash \Sigma'$,

$$i(\sigma) = j(\sigma) \; + \sum_{n \in P^{-1}(\{\sigma\})} (1 - g_n) i(n)$$

where $i(\sigma) = 0$; moreover, the condition $P(N') \subseteq \Sigma'$ implies that each $n \in P^{-1}(\{\sigma\})$ is in $N \backslash N'$ and so $i(n) = 0$. Hence $j(\sigma) = 0$.

For any $h \in H \backslash H'$, we have $h \notin H' = A^{-1}(N' \cup \Sigma')$, so $A(h) \notin N' \cup \Sigma'$, which means $A(h) \in N \backslash N' \; \cup \; \Sigma \backslash \Sigma'$, so $i(A(h)) = 0$. By the condition

$$i(A(h)) = j(h) \; + \sum_{e \in F^{-1}(\{h\})} i(e) \; + \sum_{e \in S^{-1}(\{h\})} i(e)$$

this implies $j(h) = 0$.

For any $k \in (K \backslash \operatorname{dom}(G)) \backslash (K' \backslash \operatorname{dom}(G'))$, since $\operatorname{dom}(G') \subseteq \operatorname{dom}(G)$ we have $k \notin K' = C^{-1}(E')$, so $C(k) \in E \backslash E'$, so $i(C(k)) = 0$. By the condition

$$i(C(k)) = j(k) + \sum_{k^* | G(k, k^*)} i(C(k^*))$$

this implies $j(k) = 0$.

(ii) Define $i' = i|_{\Sigma' \cup N' \cup E'}$, $j' = j|_{\Sigma' \cup N' \cup H' \cup (K' \setminus \mathrm{dom}(G'))}$, and $I = id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$. The condition

$$\forall \sigma \in \Sigma \quad i(\sigma) = j(\sigma) \ + \sum_{n \in P^{-1}(\{\sigma\})} (1 - g_n) i(n)$$

implies the same condition for (i', j') on $\mathcal{N}'$. We need to check that the value of g_n is the same in both cases, i.e., $\left|\{ n^* \in N' \mid G'(n^*, n) \}\right| = \left|\{ n^* \in N \mid G(n^*, n) \}\right|$ for every $n \in N'$. In fact, for any $n \in N'$, using $G' = G \circ I$,

$$\left|\{ n^* \in N' \mid G'(n^*, n) \}\right| = |G' \circ id_{\{n\}}| = |G \circ I \circ id_{\{n\}}| = |G \circ id_{\{n\}}| = \left|\{ n^* \in N \mid G(n^*, n) \}\right|$$

as required. Moreover, $\sum_{n \in P^{-1}(\{\sigma\})} (1 - g_n) i(n) = \sum_{n \in P'^{-1}(\{\sigma\})} (1 - g_n) i'(n)$, since the right-hand side merely omits terms $(1 - g_n) i(n)$ for which $i(n) = 0$.

Secondly, the condition

$$\forall n \in N \quad i(W(n)) = j(n) + i(n)$$

implies the same condition for (i', j') on $\mathcal{N}'$.

Thirdly, the condition

$$\forall h \in H \quad i(A(h)) = j(h) \ + \sum_{e \in F^{-1}(\{h\})} i(e) \ + \sum_{e \in S^{-1}(\{h\})} i(e)$$

implies the same condition for (i', j') on $\mathcal{N}'$. We have $\sum_{e \in F^{-1}(\{h\})} i(e) = \sum_{e \in F'^{-1}(\{h\})} i'(e)$, since the right-hand side merely omits terms $i(e)$ that are equal to 0; and similarly $\sum_{e \in S^{-1}(\{h\})} i(e) = \sum_{e \in S'^{-1}(\{h\})} i'(e)$.

Fourthly, the condition

$$\forall k \in K \setminus dom(G) \quad i(C(k)) = j(k) + \sum_{k^* | G(k, k^*)} i(C(k^*))$$

implies the same condition for (i', j') on $\mathcal{N}'$. For any $k \in K' \setminus \mathrm{dom}(G')$, we have $k \in K \setminus dom(G)$ by theorem 24(iii), so $i(C(k)) = j(k) + \sum_{k^* | G(k, k^*)} i(C(k^*))$. Also,

$$\{ k^* \mid G'(k, k^*) \} = \{ k^* \mid G(k, k^*) \wedge k^* \in K' \} = \{ k^* \mid G(k, k^*) \wedge C(k^*) \in E' \}$$

by $G' = G \circ I$. Thus $\{ k^* \mid G'(k, k^*) \} \subseteq \{ k^* \mid G(k, k^*) \}$, and any k^* in the latter set but not in the former has $C(k^*) \notin E'$ and so $i(C(k^*)) = 0$. This shows that $\sum_{k^* | G'(k, k^*)} i'(C'(k^*)) = \sum_{k^* | G(k, k^*)} i(C(k^*))$.

This completes the verification that (i', j') is a pair of inclusion functions on $\mathcal{N}'$. ∎

In practice we may confine ourselves to elementary pruning operations, involving removal of a single symbol, node or edge.

Definition. Given a network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ and a pair of inclusion functions (i, j) on $\mathcal{N}$, an *elementary pruning operation* is one of the following operations.

(i) *Pruning a symbol* $\sigma \in \Sigma$ transforms $\mathcal{N}$ to $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, where
$\Sigma' = \Sigma \setminus \left(\{\sigma\} \cup \{ \sigma^* \mid G(\sigma, \sigma^*) \}\right)$, $\quad N' = W^{-1}(\Sigma') \cap P^{-1}(\Sigma')$, $\quad H' = A^{-1}(N' \cup \Sigma')$,
$E' = F^{-1}(H') \cap S^{-1}(H')$, $\quad K' = C^{-1}(E')$,
$W' = W|_{N' \cup \Sigma'}$, $\quad P' = P|_{N'}$, $\quad A' = A|_{H'}$, $\quad F' = F|_{E'}$, $\quad S' = S|_{E'}$, $\quad C' = C|_{K'}$,

$G' = id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$.
This operation is permitted provided $\forall \sigma^* \in \Sigma \backslash \Sigma'\ i(\sigma^*) = 0$ and $\forall n \in P^{-1}(\Sigma \backslash \Sigma')\ i(n) = 0$.

(ii) *Pruning a node* $n \in N$ transforms $\mathcal{N}$ to $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, where
$\Sigma' = \Sigma$, $\quad N' = N \backslash \big(\{n\} \cup \{ n^* \mid G(n, n^*) \}\big)$, $\quad H' = A^{-1}(N' \cup \Sigma')$,
$E' = F^{-1}(H') \cap S^{-1}(H')$, $\quad K' = C^{-1}(E')$,
$W' = W|_{N' \cup \Sigma'}$, $\quad P' = P|_{N'}$, $\quad A' = A|_{H'}$, $\quad F' = F|_{E'}$, $\quad S' = S|_{E'}$, $\quad C' = C|_{K'}$,
$G' = id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$.
This operation is permitted provided $\forall n^* \in N \backslash N'\ i(n^*) = 0$.

(iii) *Pruning an edge* $e \in E$ transforms $\mathcal{N}$ to $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, where
$\Sigma' = \Sigma$, $\quad N' = N$, $\quad H' = H$, $\quad E' = E \backslash \big(\{e\} \cup \{ e^* \mid G(e, e^*) \}\big)$, $\quad K' = C^{-1}(E')$,
$W' = W$, $\quad P' = P$, $\quad A' = A$, $\quad F' = F|_{E'}$, $\quad S' = S|_{E'}$, $\quad C' = C|_{K'}$,
$G' = id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'}$.
This operation is permitted provided $\forall e^* \in E \backslash E'\ i(e^*) = 0$.

Theorem 39. All elementary pruning operations are pruning operations.

Proof. Let the network be $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ and the inclusion functions be i, j.

(i) Consider pruning a symbol $\sigma \in \Sigma$, producing $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$. We must show that $\mathcal{N}'$ is a subnetwork of $\mathcal{N}$. The only condition that is not obvious is $G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \subseteq id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G$. This is proved in five parts.

First, $G \circ id_{\Sigma'} \subseteq id_{\Sigma'} \circ G$ is verified as follows. For any $\sigma_1, \sigma_2 \in \Sigma$, if $(G \circ id_{\Sigma'})(\sigma_1, \sigma_2)$ then $G(\sigma_1, \sigma_2)$ and $\sigma_2 \in \Sigma'$, so $\sigma_2 \neq \sigma$ and $G(\sigma, \sigma_2)$ does not hold. This implies that $\sigma_1 \neq \sigma$ and $G(\sigma, \sigma_1)$ does not hold (by $G \circ G = \perp$). This gives $\sigma_1 \in \Sigma'$, and hence $(id_{\Sigma'} \circ G)(\sigma_1, \sigma_2)$, as required.

The condition $G \circ id_{N'} \subseteq id_{N'} \circ G$ is verified as follows.

$$
\begin{aligned}
G \circ id_{N'} &= G \circ id_N \circ id_{N'} = id_N \circ G \circ id_{N'} \subseteq id_{N \cup \Sigma} \circ G \circ id_{N'} \\
&\subseteq \overline{W}^{-1} \circ G \circ \overline{W} \circ id_{N'} && \text{from } \overline{W} \circ G \subseteq G \circ \overline{W} \text{ by theorem 2(iv)} \\
&\subseteq \overline{W}^{-1} \circ G \circ id_{\Sigma'} \circ \overline{W} && \text{since } N' \subseteq W^{-1}(\Sigma') \\
&\subseteq \overline{W}^{-1} \circ id_{\Sigma'} \circ G \circ \overline{W} && \text{since } G \circ id_{\Sigma'} \subseteq id_{\Sigma'} \circ G \\
&= id_{W^{-1}(\Sigma')} \circ \overline{W}^{-1} \circ G \circ \overline{W}
\end{aligned}
$$

so by theorem 4(vi)

$$G \circ id_{N'} \subseteq id_{W^{-1}(\Sigma')} \circ G$$

and similarly

$$G \circ id_{N'} \subseteq id_{P^{-1}(\Sigma')} \circ G$$

so

$$
\begin{aligned}
G \circ id_{N'} &\subseteq id_{W^{-1}(\Sigma')} \circ G \ \cap \ id_{P^{-1}(\Sigma')} \circ G \\
&= id_{W^{-1}(\Sigma')} \circ (G \cap id_{P^{-1}(\Sigma')} \circ G) && \text{by theorem 4(iv)} \\
&= id_{W^{-1}(\Sigma')} \circ id_{P^{-1}(\Sigma')} \circ (G \cap G) && \text{by theorem 4(iv)} \\
&= id_{N'} \circ G.
\end{aligned}
$$

The condition $G \circ id_{H'} \subseteq id_{H'} \circ G$ is verified as follows.

$$
\begin{aligned}
G \circ id_{H'} &= G \circ id_H \circ id_{H'} = id_H \circ G \circ id_{H'} \\
&\subseteq \overline{A}^{-1} \circ G \circ \overline{A} \circ id_{H'} && \text{from } \overline{A} \circ G \subseteq G \circ \overline{A} \text{ by theorem 2(iv)} \\
&= \overline{A}^{-1} \circ G \circ id_{N' \cup \Sigma'} \circ \overline{A} && \text{since } H' = A^{-1}(N' \cup \Sigma') \\
&\subseteq \overline{A}^{-1} \circ id_{N' \cup \Sigma'} \circ G \circ \overline{A} && \text{since } G \circ id_{\Sigma'} \subseteq id_{\Sigma'} \circ G \text{ and } G \circ id_{N'} \subseteq id_{N'} \circ G \\
&= id_{H'} \circ \overline{A}^{-1} \circ G \circ \overline{A} && \text{since } H' = A^{-1}(N' \cup \Sigma')
\end{aligned}
$$

so by theorem 4(vi)

$$G \circ id_{H'} \subseteq id_{H'} \circ G.$$

The condition $G \circ id_{E'} \subseteq id_{E'} \circ G$ is verified as follows.

$$
\begin{aligned}
G \circ id_{E'} &= G \circ id_E \circ id_{E'} = id_E \circ G \circ id_{E'} \\
&\subseteq \overline{F}^{-1} \circ G \circ \overline{F} \circ id_{E'} && \text{from } \overline{F} \circ G \subseteq G \circ \overline{F} \text{ by theorem 2(iv)} \\
&\subseteq \overline{F}^{-1} \circ G \circ id_{H'} \circ \overline{F} && \text{since } E' \subseteq F^{-1}(H') \\
&\subseteq \overline{F}^{-1} \circ id_{H'} \circ G \circ \overline{F} && \text{since } G \circ id_{H'} \subseteq id_{H'} \circ G \\
&= id_{F^{-1}(H')} \circ \overline{F}^{-1} \circ G \circ \overline{F}
\end{aligned}
$$

so by theorem 4(vi)

$$G \circ id_{E'} \subseteq id_{F^{-1}(H')} \circ G$$

and similarly

$$G \circ id_{E'} \subseteq id_{S^{-1}(H')} \circ G$$

so

$$
\begin{aligned}
G \circ id_{E'} &\subseteq id_{F^{-1}(H')} \circ G \;\cap\; id_{S^{-1}(H')} \circ G \\
&= id_{F^{-1}(H')} \circ (G \cap id_{S^{-1}(H')} \circ G) && \text{by theorem 4(iv)} \\
&= id_{F^{-1}(H')} \circ id_{S^{-1}(H')} \circ (G \cap G) && \text{by theorem 4(iv)} \\
&= id_{E'} \circ G.
\end{aligned}
$$

Finally, the condition $G \circ id_{K'} \subseteq id_{K'} \circ G$ is verified as follows.

$$
\begin{aligned}
G \circ id_{K'} &= G \circ id_K \circ id_{K'} = id_K \circ G \circ id_{K'} \\
&\subseteq \overline{C}^{-1} \circ G \circ \overline{C} \circ id_{K'} && \text{from } \overline{C} \circ G \subseteq G \circ \overline{C} \text{ by theorem 2(iv)} \\
&= \overline{C}^{-1} \circ G \circ id_{E'} \circ \overline{C} && \text{since } K' = C^{-1}(E') \\
&\subseteq \overline{C}^{-1} \circ id_{E'} \circ G \circ \overline{C} && \text{since } G \circ id_{E'} \subseteq id_{E'} \circ G \\
&= id_{K'} \circ \overline{C}^{-1} \circ G \circ \overline{C} && \text{since } K' = C^{-1}(E')
\end{aligned}
$$

so by theorem 4(vi)

$$G \circ id_{K'} \subseteq id_{K'} \circ G.$$

It follows then that $G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \subseteq id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G$ and hence $\mathcal{N}'$ is a subnetwork of $\mathcal{N}$.

We also need to verify the conditions on the inclusion functions for a pruning operation. The first condition, $\forall \sigma^* \in \Sigma \backslash \Sigma'\ i(\sigma^*) = 0$, is given.

The second condition, $\forall n \in N \backslash N'\ i(n) = 0$, is verified as follows. For any $n \in N \backslash N'$, we have $n \notin W^{-1}(\Sigma')$ or $n \notin P^{-1}(\Sigma')$. In the former case, $W(n) \in \Sigma \backslash \Sigma'$, so $i(W(n)) = 0$, so, by the constraint $i(W(n)) = j(n) + i(n)$, we have $i(n) = 0$ as required. In the latter case, $n \in P^{-1}(\Sigma \backslash \Sigma')$, so we are given that $i(n) = 0$ as required.

The third condition, $\forall e \in E \backslash E'\ i(e) = 0$, is verified as follows. For any $e \in E \backslash E'$, we have $e \notin F^{-1}(H')$ or $e \notin S^{-1}(H')$. In the former case, $F(e) \notin H'$, so $A(F(e)) \notin N' \cup \Sigma'$, so $i(A(F(e))) = 0$. Hence by the constraint $\forall h \in H\ i(A(h)) = j(h) + \sum_{e \in F^{-1}(\{h\})} i(e) + \sum_{e \in S^{-1}(\{h\})} i(e)$ it follows that $i(e) = 0$ as required. The latter case is similar.

This completes the proof that pruning σ is a pruning operation.

(ii) Consider pruning a node $n \in N$, producing $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$. The verification that $\mathcal{N}'$ is a subnetwork of $\mathcal{N}$ is similar to the one in part (i), but simpler. We also need to verify the conditions on the inclusion functions for a pruning operation.

The first condition, $\forall \sigma \in \Sigma \backslash \Sigma'\ i(\sigma) = 0$, is vacuously true because $\Sigma' = \Sigma$.

The second condition, $\forall n^* \in N \backslash N'\ i(n^*) = 0$, is given.

The third condition, $\forall e \in E \backslash E'\ i(e) = 0$, is verified exactly as in part (i).

(iii) Consider pruning an edge $e \in E$, producing $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$. The verification that $\mathcal{N}'$ is a subnetwork of $\mathcal{N}$ is similar to the one in part (i), but simpler. We also need to verify the conditions on the inclusion functions for a pruning operation.

The first two conditions, $\forall \sigma \in \Sigma \backslash \Sigma'\ i(\sigma) = 0$ and $\forall n \in N \backslash N'\ i(n) = 0$, are vacuously true because $\Sigma' = \Sigma$ and $N' = N$.

The third condition, $\forall e^* \in E \backslash E'\ i(e^*) = 0$, is given. ∎

The following theorem shows why it is sufficient to restrict attention to elementary pruning operations.

Theorem 40. If any non-trivial pruning operation is possible on a network then an elementary pruning operation is possible on it.

Proof. Consider a network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ with inclusion functions i, j and suppose that a non-trivial pruning operation is possible on it, producing a proper subnetwork $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$. Then there exists $\sigma \in \Sigma \backslash \Sigma'$ or $n \in N \backslash N'$ or $h \in H \backslash H'$ or $e \in E \backslash E'$ or $k \in K \backslash K'$. However, note that if $h \in H \backslash H'$ then $A(h) \in N \backslash N' \cup \Sigma \backslash \Sigma'$, since $H' = A^{-1}(N' \cup \Sigma')$; and if $k \in K \backslash K'$ then $C(k) \in E \backslash E'$ since $K' = C^{-1}(E')$. So we can infer that there exists $\sigma \in \Sigma \backslash \Sigma'$ or $n \in N \backslash N'$ or $e \in E \backslash E'$.

Consider the case where there exists $\sigma \in \Sigma \backslash \Sigma'$. Then $\forall \sigma^* \in \Sigma\ (G(\sigma, \sigma^*) \Rightarrow \sigma^* \in \Sigma \backslash \Sigma')$, by the condition $G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \subseteq id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G$. Defining a set $\Sigma_0 = \{\sigma\} \cup \{\, \sigma^* \mid G(\sigma, \sigma^*) \,\}$, we then have $\Sigma_0 \subseteq \Sigma \backslash \Sigma'$, so $\forall \sigma^* \in \Sigma_0\ i(\sigma^*) = 0$; moreover the condition $P(N') \subseteq \Sigma'$ implies $P^{-1}(\Sigma_0) \subseteq N \backslash N'$ and hence $\forall n \in P^{-1}(\Sigma_0)\ i(n) = 0$. Hence we can perform on $\mathcal{N}$ the elementary pruning operation of pruning σ, giving a subnetwork $\mathcal{N}'' = (\Sigma'', N'', H'', E'', K'', W'', P'', A'', F'', S'', C'', G'')$, with $\Sigma'' = \Sigma \backslash \Sigma_0$.

Next consider the case where there exists $n \in N \backslash N'$. Then $\forall n^* \in N\ (G(n, n^*) \Rightarrow n^* \in N \backslash N')$, by the condition $G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \subseteq id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G$. Defining a set $N_0 = \{n\} \cup \{\, n^* \mid G(n, n^*) \,\}$, we then have $N_0 \subseteq N \backslash N'$, so $\forall n^* \in N_0\ i(n^*) = 0$. Hence we can perform on $\mathcal{N}$ the elementary pruning operation of pruning n, giving a subnetwork $\mathcal{N}'' = (\Sigma'', N'', H'', E'', K'', W'', P'', A'', F'', S'', C'', G'')$, with $N'' = N \backslash N_0$.

Finally consider the case where there exists $e \in E \backslash E'$. Then $\forall e^* \in E\ (G(e,e^*) \Rightarrow e^* \in E \backslash E')$, by the condition $G \circ id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \subseteq id_{\Sigma' \cup N' \cup H' \cup E' \cup K'} \circ G$. Defining a set $E_0 = \{e\} \cup \{\, e^* \mid G(e,e^*)\,\}$, we then have $E_0 \subseteq E \backslash E'$, so $\forall e^* \in E_0\ i(e^*) = 0$. Hence we can perform on $\mathcal{N}$ the elementary pruning operation of pruning e, giving a subnetwork $\mathcal{N}'' = (\Sigma'', N'', H'', E'', K'', W'', P'', A'', F'', S'', C'', G'')$, with $E'' = E \backslash E_0$.

Hence in all cases an elementary pruning operation is possible. ▮

Theorem 41. If a pruning operation is carried out on a network $\mathcal{N}$, given a pair of inclusion functions (i,j) on $\mathcal{N}$, resulting in a subnetwork $\mathcal{N}'$, then, for any image I and any embedding token u, embedding type v and bareness function B for $\mathcal{N}$,

$$IM(I, \mathcal{N}', u', v', i', j', B') = IM(I, \mathcal{N}, u, v, i, j, B)$$

where u', v', i', j', B' are the restrictions of u, v, i, j, B to $\mathcal{N}'$. Moreover, the symmetry condition holds for $\mathcal{N}', u', v'$ if it holds for $\mathcal{N}, u, v$.

Proof. We use the usual notation, $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, $v = (sub, con, rel, symm, tem, in)$, $v' = (sub', con', rel', symm', tem', in')$. By the definition of a pruning operation and theorem 38(i),

$$\forall \sigma \in \Sigma \backslash \Sigma'\ i(\sigma) = 0 = j(\sigma), \quad \forall n \in N \backslash N'\ i(n) = 0, \quad \forall h \in H \backslash H'\ j(h) = 0,$$
$$\forall e \in E \backslash E'\ i(e) = 0, \quad \forall k \in (K \backslash \operatorname{dom}(G)) \backslash (K' \backslash \operatorname{dom}(G'))\ j(k) = 0.$$

This implies that the pruning operation makes no difference to the value of IM. The only term for which this is not obvious is $\sum_{(\sigma,\sigma^*) \mid G(\sigma,\sigma^*)} i(\sigma^*) E_{sub(\sigma,\sigma^*)}(u(\sigma)^{-1} \cdot u(\sigma^*))$. Since $G'_\Sigma = G_\Sigma \circ id_{\Sigma'}$ (by theorem 24(i)), the difference

$$\sum_{(\sigma,\sigma^*) \mid G(\sigma,\sigma^*)} i(\sigma^*) E_{sub(\sigma,\sigma^*)}(u(\sigma)^{-1} \cdot u(\sigma^*)) \ - \sum_{(\sigma,\sigma^*) \mid G'(\sigma,\sigma^*)} i'(\sigma^*) E_{sub'(\sigma,\sigma^*)}(u'(\sigma)^{-1} \cdot u'(\sigma^*))$$

is the sum of $i(\sigma^*) E_{sub(\sigma,\sigma^*)}(u(\sigma)^{-1} \cdot u(\sigma^*))$ over pairs (σ, σ^*) having $\sigma^* \in \Sigma \backslash \Sigma'$, and for such pairs the term vanishes since $i(\sigma^*) = 0$.

The symmetry condition for $\mathcal{N}', u', v'$ immediately follows from the one for $\mathcal{N}, u, v$. ▮

Pruning is used in the recognition process as follows. We have a pattern $\mathcal{N}_1$, a parse $p\colon \mathcal{N}_1 \to \mathcal{N}_0$, an embedding token u for $\mathcal{N}_1$, and a pair of inclusion functions (i,j) on $\mathcal{N}_1$. We carry out an elementary pruning operation on $\mathcal{N}_1$ to give a subnetwork $\mathcal{N}'_1$, with a new parse $p|_{\mathcal{N}'_1}\colon \mathcal{N}'_1 \to \mathcal{N}_0$ (§3.7), a new embedding token $u|_{\mathcal{N}'_1}$ (§4.2), and a new pair of inclusion functions, the restriction of (i,j) to $\mathcal{N}'_1$ (theorem 38(ii)).

8.3 Unprunability and its consequences

Definition. A network $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is *unprunable*, given a pair of inclusion functions (i,j) on $\mathcal{N}$, iff there is no proper subnetwork $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$ of $\mathcal{N}$ satisfying

$$\forall \sigma \in \Sigma \backslash \Sigma' \; i(\sigma) = 0, \quad \forall n \in N \backslash N' \; i(n) = 0, \quad \forall e \in E \backslash E' \; i(e) = 0.$$

(In other words, a network is unprunable iff no non-trivial pruning operation is possible on it, or, equivalently by theorem 40, iff no elementary pruning operation is possible on it.)

Theorem 42. If

(a) $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is a network,

(b) $p : \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, where $\mathcal{N}_0$ is a semi-definite network,

(c) (i,j) is a pair of inclusion functions on $\mathcal{N}_1$ satisfying $\forall x \in \Sigma_1 \cup N_1 \cup E_1 \; i(x) \in \{0,1\}$,

(d) $\mathcal{N}_1$ is unprunable, given (i,j),

then $\forall e \in E_1 \; i(e) = 1$.

Proof. Consider the pruning operation that transforms $\mathcal{N}_1$ to $\mathcal{N}_2 = (\Sigma_2, N_2, H_2, E_2, K_2, W_2, P_2, A_2, F_2, S_2, C_2, G_2)$, where

$$\Sigma_2 = \Sigma_1, \quad N_2 = N_1, \quad H_2 = H_1, \quad E_2 = \{\, e \in E_1 \mid i(e) = 1 \,\}, \quad K_2 = {C_1}^{-1}(E_2),$$
$$W_2 = W_1, \quad P_2 = P_1, \quad A_2 = A_1, \quad F_2 = F_1|_{E_2}, \quad S_2 = S_1|_{E_2}, \quad C_2 = C_1|_{K_2},$$
$$G_2 = id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2}.$$

I shall show that this is indeed a pruning operation, and then since $\mathcal{N}_1$ is assumed unprunable it will follow that $\mathcal{N}_2 = \mathcal{N}_1$.

First we check that $\mathcal{N}_2$ is a subnetwork of $\mathcal{N}_1$. The conditions

$$W_1(N_2) \subseteq \Sigma_2, \quad P_1(N_2) \subseteq \Sigma_2, \quad H_2 = {A_1}^{-1}(N_2 \cup \Sigma_2), \quad F_1(E_2) \subseteq H_2, \quad S_1(E_2) \subseteq H_2$$

are immediate, since $\Sigma_2 = \Sigma_1$, $N_2 = N_1$ and $H_2 = H_1$. The condition $K_2 = {C_1}^{-1}(E_2)$ holds by definition.

The functions $W_2, P_2, A_2, F_2, S_2, C_2$ are as they ought to be for a subnetwork.

To verify $G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1$ we need the following derivations.

For every $k, k^* \in K_1$ such that $G_1(k, k^*)$, we have $k \in \mathrm{ran}(G_1)$, so $k \notin \mathrm{dom}(G_1)$, so

$$i(C_1(k)) = j(k) + \sum_{k' \mid G_1(k,k')} i(C_1(k')) \geq i(C_1(k^*));$$

thus if $k^* \in K_2$ then $C_1(k^*) \in E_2$, so $i(C_1(k^*)) = 1$, so $i(C_1(k)) = 1$, so $C_1(k) \in E_2$, so $k \in K_2$. This shows

$$G_{1K} \circ id_{K_2} \subseteq id_{K_2} \circ G_{1K}. \tag{1}$$

Next,

$$\begin{aligned}
G_{1E} \circ id_{E_2} &= \overline{C_1} \circ G_{1K} \circ \overline{C_1}^{-1} \circ id_{E_2} && \text{by theorem 20(iv)} \\
&= \overline{C_1} \circ G_{1K} \circ id_{K_2} \circ \overline{C_1}^{-1} && \text{since } K_2 = {C_1}^{-1}(E_2) \\
&\subseteq \overline{C_1} \circ id_{K_2} \circ G_{1K} \circ \overline{C_1}^{-1} && \text{by (1)} \\
&= id_{E_2} \circ \overline{C_1} \circ G_{1K} \circ \overline{C_1}^{-1} && \text{since } K_2 = {C_1}^{-1}(E_2) \\
&= id_{E_2} \circ G_{1E} && \text{by theorem 20(iv).}
\end{aligned} \tag{2}$$

Next,

$$G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2} = G_{1\Sigma} \cup G_{1N} \cup G_{1H} = id_{\Sigma_2 \cup N_2 \cup H_2} \circ G_1 \tag{3}$$

since $\Sigma_2 = \Sigma_1$, $N_2 = N_1$ and $H_2 = H_1$. Combining (1), (2) and (3) gives

$$G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1$$

as required. Thus $\mathcal{N}_2$ is indeed a subnetwork of $\mathcal{N}_1$. The remaining conditions for a pruning operation,

$$\forall \sigma \in \Sigma_1 \backslash \Sigma_2 \ i(\sigma) = 0, \quad \forall n \in N_1 \backslash N_2 \ i(n) = 0, \quad \forall e \in E_1 \backslash E_2 \ i(e) = 0,$$

hold by definition of Σ_2, N_2, E_2.

Since $\mathcal{N}_1$ is assumed unprunable, given (i,j), it follows that $\mathcal{N}_2 = \mathcal{N}_1$. This means that $E_2 = E_1$ and so $\forall e \in E_1 \ i(e) = 1$. ∎

Theorem 43. If

(a) $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is a network,
(b) $p: \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, where $\mathcal{N}_0$ is a semi-definite network,
(c) (i,j) is a pair of inclusion functions on $\mathcal{N}_1$ satisfying
$\forall x \in \Sigma_1 \cup N_1 \cup E_1 \ i(x) \in \{0,1\}, \quad \forall h \in H_1 \ j(h) = 0, \quad \forall k \in K_1 \backslash \operatorname{dom}(G_1) \ j(k) = 0,$
(d) $\mathcal{N}_1$ is unprunable, given (i,j),

then

(i) $\forall e \in E_1 \ i(e) = 1$,
(ii) $\forall n \in N_1 \ i(n) = 1$,
(iii) $E_1 \xrightarrow{F_1} A_1{}^{-1}(N_1 \cup Y) \xleftarrow{S_1} E_1$ is a sum diagram in the category of sets, where $Y = \{\, \sigma \in \Sigma_1 \mid i(\sigma) = 1 \,\}$,
(iv) $\exists f: K_1 \backslash \operatorname{dom}(G_1) \to K_1 \ G_{1K}^{-1} = \overline{f}$,
(v) $\forall n \in N_1 \ i(W_1(n)) = 1 = i(P_1(n))$,
(vi) $\forall X \subseteq \Sigma_1 \ |P_1{}^{-1}(X)| - |G_{1N} \circ id_{P_1{}^{-1}(X)}| \leq |X|$.

Proof. (i) is given by theorem 42.

(ii) Consider the pruning operation that transforms $\mathcal{N}_1$ to $\mathcal{N}_2 = (\Sigma_2, N_2, H_2, E_2, K_2, W_2, P_2, A_2, F_2, S_2, C_2, G_2)$, where

$\Sigma_2 = \Sigma_1, \quad N_2 = \{\, n \in N_1 \mid i(n) = 1 \,\}, \quad H_2 = A_1{}^{-1}(N_2 \cup \Sigma_2), \quad E_2 = E_1, \quad K_2 = K_1,$
$W_2 = W_1|_{N_2 \cup \Sigma_2}, \quad P_2 = P_1|_{N_2}, \quad A_2 = A_1|_{H_2}, \quad F_2 = F_1, \quad S_2 = S_1, \quad C_2 = C_1,$
$G_2 = id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2}.$

I shall show that this is indeed a pruning operation, and then since $\mathcal{N}_1$ is assumed unprunable it will follow that $\mathcal{N}_2 = \mathcal{N}_1$.

First we check that $\mathcal{N}_2$ is a subnetwork of $\mathcal{N}_1$. The conditions $W_1(N_2) \subseteq \Sigma_2$ and $P_1(N_2) \subseteq \Sigma_2$, are immediate, since $\Sigma_2 = \Sigma_1$.

Define $Y = \{\, \sigma \in \Sigma_1 \mid i(\sigma) = 1 \,\}$ (as in part (iii) in the statement of the theorem). For any $h \in H_1$, we have the condition

$$i(A_1(h)) = j(h) \ + \sum_{e \in F_1{}^{-1}(\{h\})} i(e) \ + \sum_{e \in S_1{}^{-1}(\{h\})} i(e)$$

where $j(h) = 0$; so $h \in A_1{}^{-1}(N_2 \cup Y)$ iff $i(A_1(h)) = 1$ iff there exists $e \in F_1{}^{-1}(\{h\})$ or $e \in S_1{}^{-1}(\{h\})$ (where such an e is unique and belongs to just one of the two sets). This means that

$$E_1 \xrightarrow{F_1} A_1{}^{-1}(N_2 \cup Y) \xleftarrow{S_1} E_1 \text{ is a sum diagram.} \tag{1}$$

This implies $F_1(E_1) \subseteq {A_1}^{-1}(N_2 \cup Y) \subseteq H_2$ and $S_1(E_1) \subseteq {A_1}^{-1}(N_2 \cup Y) \subseteq H_2$, as required.

The functions $W_2, P_2, A_2, F_2, S_2, C_2$ are as required for a subnetwork.

To verify $G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1$ we need the following derivations.

Since $\Sigma_2 = \Sigma_1$, $E_2 = E_1$ and $K_2 = K_1$,

$$\begin{aligned} G_1 \circ id_{\Sigma_2} &= G_{1\Sigma} = id_{\Sigma_2} \circ G_1, \\ G_1 \circ id_{E_2} &= G_{1E} = id_{E_2} \circ G_1, \\ G_1 \circ id_{K_2} &= G_{1K} = id_{K_2} \circ G_1. \end{aligned} \tag{2}$$

Next,

$$\begin{aligned} G_{1H} \circ id_{{A_1}^{-1}(N_2)} &\subseteq G_{1H} \circ id_{{A_1}^{-1}(N_2 \cup Y)} \\ &= G_{1H} \circ \left(\overline{F_1} \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ \overline{S_1}^{-1}\right) && \text{by theorem 7 and (1)} \\ &= G_{1H} \circ \overline{F_1} \circ \overline{F_1}^{-1} \;\cup\; G_{1H} \circ \overline{S_1} \circ \overline{S_1}^{-1} \\ &= G_1 \circ \overline{F_1} \circ \overline{F_1}^{-1} \;\cup\; G_1 \circ \overline{S_1} \circ \overline{S_1}^{-1} \\ &= \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ G_1 \circ \overline{S_1}^{-1} && \text{by theorem 20(ii)} \\ &= \overline{F_1} \circ id_{E_1} \circ G_1 \circ \overline{F_1}^{-1} \;\cup\; \overline{S_1} \circ id_{E_1} \circ G_1 \circ \overline{S_1}^{-1} \\ &\subseteq id_{H_2} \circ \overline{F_1} \circ G_1 \circ \overline{F_1}^{-1} \;\cup\; id_{H_2} \circ \overline{S_1} \circ G_1 \circ \overline{S_1}^{-1} && \text{since } F_1(E_1) \subseteq H_2 \\ &= id_{H_2} \circ G_1 \circ \overline{F_1} \circ \overline{F_1}^{-1} \;\cup\; id_{H_2} \circ G_1 \circ \overline{S_1} \circ \overline{S_1}^{-1} && \text{by theorem 20(ii)} \\ &\subseteq id_{H_2} \circ G_1 \circ id_{H_1} \;\cup\; id_{H_2} \circ G_1 \circ id_{H_1} && \text{by theorem 2(i)} \\ &= id_{H_2} \circ G_{1H}. \end{aligned} \tag{3}$$

Next, from $\overline{A_1} \circ G_1 \subseteq G_1 \circ \overline{A_1}$ we have

$$\begin{aligned} G_{1H} \circ id_{{A_1}^{-1}(\Sigma_2)} &\subseteq \overline{A_1}^{-1} \circ G_1 \circ \overline{A_1} \circ id_{{A_1}^{-1}(\Sigma_2)} && \text{by theorem 2(iv)} \\ &= \overline{A_1}^{-1} \circ G_1 \circ id_{\Sigma_2} \circ \overline{A_1} \\ &= \overline{A_1}^{-1} \circ id_{\Sigma_2} \circ G_1 \circ \overline{A_1} && \text{by (2)} \\ &\subseteq id_{H_2} \circ \overline{A_1}^{-1} \circ G_1 \circ \overline{A_1} && \text{since } {A_1}^{-1}(\Sigma_2) \subseteq H_2 \end{aligned}$$

so by theorem 4(vi)

$$G_{1H} \circ id_{{A_1}^{-1}(\Sigma_2)} \subseteq id_{H_2} \circ G_{1H}. \tag{4}$$

Hence,

$$\begin{aligned} G_{1H} \circ id_{H_2} &= G_{1H} \circ id_{{A_1}^{-1}(N_2 \cup \Sigma_2)} \\ &= G_{1H} \circ id_{{A_1}^{-1}(N_2)} \;\cup\; G_{1H} \circ id_{{A_1}^{-1}(\Sigma_2)} \\ &\subseteq id_{H_2} \circ G_{1H} && \text{by (3) and (4).} \end{aligned} \tag{5}$$

Next,

$$\begin{aligned} G_{1N} \circ id_{N_2} &= id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \circ id_{N_2} && \text{by theorem 20(iii)} \\ &\subseteq id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ id_{H_2} \circ \overline{A_1}^{-1} && \text{since } {A_1}^{-1}(N_2) \subseteq H_2 \\ &\subseteq id_{N_1} \circ \overline{A_1} \circ id_{H_2} \circ G_{1H} \circ \overline{A_1}^{-1} && \text{by (5)} \\ &= id_{N_1} \circ id_{N_2 \cup \Sigma_2} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} && \text{since } H_2 = {A_1}^{-1}(N_2 \cup \Sigma_2) \\ &= id_{N_2} \circ id_{N_1} \circ \overline{A_1} \circ G_{1H} \circ \overline{A_1}^{-1} \\ &= id_{N_2} \circ G_{1N} && \text{by theorem 20(iii).} \end{aligned} \tag{6}$$

From (2), (5) and (6),

$$G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1.$$

Thus $\mathcal{N}_2$ is indeed a subnetwork of $\mathcal{N}_1$. The remaining conditions for a pruning operation,

$$\forall \sigma \in \Sigma_1 \backslash \Sigma_2 \; i(\sigma) = 0, \quad \forall n \in N_1 \backslash N_2 \; i(n) = 0, \quad \forall e \in E_1 \backslash E_2 \; i(e) = 0,$$

hold by definition of Σ_2, N_2, E_2.

Since $\mathcal{N}_1$ is assumed unprunable, given (i,j), it follows that $\mathcal{N}_2 = \mathcal{N}_1$. This means that $N_2 = N_1$ and so $\forall n \in N_1 \; i(n) = 1$.

(iii) follows from (1), bearing in mind that $N_2 = N_1$.

(iv) For any $k \in K_1 \backslash \operatorname{dom}(G_1)$,

$$i(C_1(k)) = j(k) + \sum_{k^* \mid G_1(k,k^*)} i(C_1(k^*))$$

where $i(C_1(k)) = 1$, $j(k) = 0$, and each $i(C_1(k^*)) = 1$, by part (i) and hypothesis (c), so there exists a unique k^* such that $G_1(k,k^*)$. Define $f\colon K_1 \backslash \operatorname{dom}(G_1) \to K_1$ mapping k to this k^*. On the other hand, for any $k \in K_1 \wedge \operatorname{dom}(G_1)$, we have $k \notin \operatorname{ran}(G_1)$, so there exists no k^* such that $G_1(k,k^*)$. Thus $G_{1K}^{-1} = \overline{f}$.

(v) $\forall n \in N_1 \; i(W_1(n)) = 1$ holds since, for any $n \in N_1$,

$$i(W_1(n)) = j(n) + i(n)$$

where $i(n) = 1$ by part (ii), so $i(W_1(n)) = 1$.

The other equation, $\forall n \in N_1 \; i(P_1(n)) = 1$, is proved using theorem 21 and theorem 13. The hypotheses of theorem 21 hold, by parts (iii) and (iv) just proved. Theorem 21 then tells us that G_{1N} is acyclic. The next step is to apply theorem 13 to the sets N_1, Σ_1, the function $P_1\colon N_1 \to \Sigma_1$ and the finite relation G_{1N} on N_1. Let us check the hypotheses of theorem 13.

Hypothesis (a): $G_{1N} \circ G_{1N} \subseteq G_1 \circ G_1 = \perp$ (since $\mathcal{N}_1$ is a network).

Hypothesis (b): G_{1N} is acyclic (as just shown).

Hypothesis (c): $\overline{P_1} \circ G_{1N} = \overline{P_1} \circ G_1 \subseteq \overline{P_1}$ (since $\mathcal{N}_1$ is a network).

Now, for any $\sigma \in \Sigma_1$ such that $i(\sigma) = 0$, taking $X = \{\sigma\}$ in theorem 13(i) gives

$$|{P_1}^{-1}(\{\sigma\})| - |G_{1N} \circ id_{{P_1}^{-1}(\{\sigma\})}| \geq 0.$$

Moreover, (recalling the notation $g_n = \big|\{\, n^* \in N_1 \mid G_1(n^*,n) \,\}\big|$)

$$\begin{aligned} 0 = i(\sigma) = j(\sigma) \; + \sum_{n \in {P_1}^{-1}(\{\sigma\})} (1 - g_n) i(n) &= j(\sigma) \; + \sum_{n \in {P_1}^{-1}(\{\sigma\})} (1 - g_n) \\ &= j(\sigma) + |{P_1}^{-1}(\{\sigma\})| - |\{\, (n^*,n) \in N_1 \times {P_1}^{-1}(\{\sigma\}) \mid G_1(n^*,n) \,\}| \\ &= j(\sigma) + |{P_1}^{-1}(\{\sigma\})| - |G_{1N} \circ id_{{P_1}^{-1}(\{\sigma\})}| \end{aligned}$$

so $j(\sigma) = 0$ and $|{P_1}^{-1}(\{\sigma\})| - |G_{1N} \circ id_{{P_1}^{-1}(\{\sigma\})}| = 0$. Then by theorem 13(ii) ${P_1}^{-1}(\{\sigma\}) = \emptyset$. The contrapositive of this result is that if ${P_1}^{-1}(\{\sigma\}) \neq \emptyset$ then $i(\sigma) = 1$. Thus $\forall n \in N_1 \; i(P_1(n)) = 1$, as required.

(vi) For any $\sigma \in \Sigma_1$,

$$i(\sigma) = j(\sigma) + \sum_{n \in P_1{}^{-1}(\{\sigma\})} (1 - g_n) i(n) = j(\sigma) + |P_1{}^{-1}(\{\sigma\})| - |G_{1N} \circ id_{P_1{}^{-1}(\{\sigma\})}|$$

as in part (v), where $i(\sigma) \in \{0,1\}$ and $j(\sigma) \in [0,1]$, so

$$|P_1{}^{-1}(\{\sigma\})| - |G_{1N} \circ id_{P_1{}^{-1}(\{\sigma\})}| \leq 1.$$

For any $X \subseteq \Sigma_1$, summing this inequality over all $\sigma \in X$ gives

$$|P_1{}^{-1}(X)| - |G_{1N} \circ id_{P_1{}^{-1}(X)}| \leq |X|$$

as required. ▮

Theorem 44. If
(a) $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is a network,
(b) $p{:}\mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, where $\mathcal{N}_0$ is a semi-definite network,
(c) (i,j) is a pair of inclusion functions on $\mathcal{N}_1$ satisfying

$\forall x \in \Sigma_1 \cup N_1 \cup E_1 \; i(x) \in \{0,1\}, \quad \forall h \in H_1 \; j(h) = 0, \quad \forall k \in K_1 \setminus \mathrm{dom}(G_1) \; j(k) = 0,$

(d) $\mathcal{N}_1$ is unprunable, given (i,j),
(e) $E_1 \overset{A_1 \circ F_1}{\underset{A_1 \circ S_1}{\rightrightarrows}} N_1 \cup \Sigma_1 \xrightarrow{W_1} \Sigma_1$ is a coequaliser diagram in the category of sets,

then
(i) $\forall x \in \Sigma_1 \cup N_1 \cup E_1 \; i(x) = 1$,
(ii) $\mathcal{N}_1$ is definite.

Proof. The hypotheses of this theorem include those of theorem 43, so we can apply theorem 43.

(i) From theorem 43(i),(ii) we already have $\forall e \in E_1 \; i(e) = 1$ and $\forall n \in N_1 \; i(n) = 1$. Consider the pruning operation that transforms $\mathcal{N}_1$ to $\mathcal{N}_2 = (\Sigma_2, N_2, H_2, E_2, K_2, W_2, P_2, A_2, F_2, S_2, C_2, G_2)$, where

$\Sigma_2 = \{\, \sigma \in \Sigma_1 \mid i(\sigma) = 1 \,\}, \quad N_2 = N_1, \quad H_2 = A_1{}^{-1}(N_2 \cup \Sigma_2), \quad E_2 = E_1, \quad K_2 = K_1,$
$W_2 = W_1|_{N_2 \cup \Sigma_2}, \quad P_2 = P_1, \quad A_2 = A_1|_{H_2}, \quad F_2 = F_1, \quad S_2 = S_1, \quad C_2 = C_1,$
$G_2 = id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2}.$

I shall show that this is indeed a pruning operation, and then since $\mathcal{N}_1$ is assumed unprunable it will follow that $\mathcal{N}_2 = \mathcal{N}_1$.

First we check that $\mathcal{N}_2$ is a subnetwork. The first two conditions, $W_1(N_2) \subseteq \Sigma_2$ and $P_1(N_2) \subseteq \Sigma_2$ follow from theorem 43(v).

Theorem 43(iii) tells us that

$$E_1 \xrightarrow{F_1} H_2 \xleftarrow{S_1} E_1 \text{ is a sum diagram} \tag{1}$$

since $H_2 = A_1{}^{-1}(N_2 \cup \Sigma_2) = A_1{}^{-1}(N_1 \cup Y)$. This implies $F_1(E_2) \subseteq H_2$ and $S_1(E_2) \subseteq H_2$, as required.

The functions $W_2, P_2, A_2, F_2, S_2, C_2$ are as required for a subnetwork.

Next we verify the condition $G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1$.

The hypotheses of theorem 22 are satisfied (with $Y = \Sigma_2$). Theorem 22(iii) gives

$$G_1 \circ id_{\Sigma_2} \subseteq id_{\Sigma_2} \circ G_1. \tag{2}$$

The conditions

$$G_1 \circ id_{N_2} = id_{N_2} \circ G_1 \tag{3}$$
$$G_1 \circ id_{E_2} = id_{E_2} \circ G_1 \tag{4}$$
$$G_1 \circ id_{K_2} = id_{K_2} \circ G_1 \tag{5}$$

hold trivially, since $N_2 = N_1$, $E_2 = E_1$ and $K_2 = K_1$. Also,

$$\begin{aligned}
G_1 \circ id_{H_2} &= id_{H_1} \circ G_1 \circ id_{H_2} && \text{since } H_2 \subseteq H_1 \\
&\subseteq \overline{A_1}^{-1} \circ G_1 \circ \overline{A_1} \circ id_{H_2} && \text{from } \overline{A_1} \circ G_1 \subseteq G_1 \circ \overline{A_1} \text{ by theorem 2(iv)} \\
&= \overline{A_1}^{-1} \circ G_1 \circ id_{N_2 \cup \Sigma_2} \circ \overline{A_1} && \text{since } H_2 = A_1{}^{-1}(N_2 \cup \Sigma_2) \\
&\subseteq \overline{A_1}^{-1} \circ id_{N_2 \cup \Sigma_2} \circ G_1 \circ \overline{A_1} && \text{by (3) and (2)} \\
&= id_{H_2} \circ \overline{A_1}^{-1} \circ G_1 \circ \overline{A_1} && \text{since } H_2 = A_1{}^{-1}(N_2 \cup \Sigma_2)
\end{aligned}$$

and hence by theorem 4(vi)

$$G_1 \circ id_{H_2} \subseteq id_{H_2} \circ G_1. \tag{6}$$

From (2), (3), (4), (5) and (6) we derive $G_1 \circ id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \subseteq id_{\Sigma_2 \cup N_2 \cup H_2 \cup E_2 \cup K_2} \circ G_1$. Hence $\mathcal{N}_2$ is indeed a subnetwork of $\mathcal{N}_1$.

The conditions

$$\forall \sigma \in \Sigma_1 \backslash \Sigma_2 \; i(\sigma) = 0, \quad \forall n \in N_1 \backslash N_2 \; i(n) = 0, \quad \forall e \in E_1 \backslash E_2 \; i(e) = 0$$

follow immediately from the definitions of Σ_2, N_2, E_2. This completes the verification that we have a pruning operation.

Since $\mathcal{N}_1$ is assumed to be unprunable, given (i,j), the pruning operation must be the trivial one, with $\mathcal{N}_2 = \mathcal{N}_1$. This means that $\Sigma_2 = \Sigma_1$ and so $\forall \sigma \in \Sigma_1 \; i(\sigma) = 1$, as required.

(ii) We shall apply theorem 23; let us check its hypotheses.

Hypothesis (a) holds by theorem 43(vi).

Hypothesis (b), that $E_1 \overset{F_1}{\to} H_1 \overset{S_1}{\leftarrow} E_1$ is a sum diagram, follows from (1) and the fact that $\mathcal{N}_2 = \mathcal{N}_1$.

Hypothesis (c) holds by theorem 27(iv).

Hypothesis (d) holds by hypothesis (e) of this theorem.

Thus it follows by theorem 23 that $\mathcal{N}_1$ is definite. ∎

8.4 Extension operations

Given a grammar $\mathcal{N}_0$, a pattern $\mathcal{N}_1$, a homomorphism $p{:}\mathcal{N}_1 \to \mathcal{N}_0$, and an embedding token u for $\mathcal{N}_1$, an *extension* of $(\mathcal{N}_1, p, u)$ is a triple $(\mathcal{N}_1', p', u')$ where $\mathcal{N}_1$ is a subnetwork of $\mathcal{N}_1'$, $p'{:}\mathcal{N}_1' \to \mathcal{N}_0$ is a homomorphism such that $p = p'|_{\mathcal{N}_1}$, and u' is an embedding token for $\mathcal{N}_1'$ such that $u = u'|_{\mathcal{N}_1}$.

$(\mathcal{N}_1', p', u')$ is a *minimal* extension of $(\mathcal{N}_1, p, u)$ satisfying a condition P iff

(i) $(\mathcal{N}_1', p', u')$ is an extension of $(\mathcal{N}_1, p, u)$ satisfying P;

(ii) for any extension $(\mathcal{N}_1'', p'', u'')$ of $(\mathcal{N}_1, p, u)$ satisfying P, $|\mathcal{N}_1'| \leq |\mathcal{N}_1''|$,

where the cardinality $|\mathcal{N}|$ of a network $\mathcal{N}$ is defined by $|(\Sigma, N, H, E, K, W, P, A, F, S, C, G)| = |\Sigma| + |N| + |H| + |E| + |K|$.

An extension is *trivial* iff $\mathcal{N}_1' = \mathcal{N}_1$.

Using these notions we can define the extension operations used in the algorithm. The notation is as usual: we have a grammar $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ and an embedding type $v = (sub, con, rel, symm, tem, in)$ on $\mathcal{N}_0$, a pattern $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ with a parse $p: \mathcal{N}_1 \to \mathcal{N}_0$, an embedding token u, a pair of inclusion functions (i,j), and a bareness function B for $\mathcal{N}_1$; we shall produce an extended network $\mathcal{N}_1' = (\Sigma_1', N_1', H_1', E_1', K_1', W_1', P_1', A_1', F_1', S_1', C_1', G_1')$ with a new parse $p': \mathcal{N}_1' \to \mathcal{N}_0$ and embedding token u'. The threshold θ is the positive constant used in the *DM* function (see §4.5); the thresholds $\theta_0, \theta_1, \theta_2, \theta_3, \theta_4$ are used only in the extension operations and may be made dependent on parameters such as temperature.

In the extension operations the following additional conditions, referred to collectively as *the extension conditions*, will be imposed.

- $\forall \sigma \in \Sigma_1\ (P_1'^{-1}(\{\sigma\}) \not\subseteq N_1 \ \Rightarrow\ j(\sigma) > \theta_0 \ \wedge\ B(\sigma) < \theta)$,
- $\forall h \in H_1\ (F_1'^{-1}(\{h\}) \cup S_1'^{-1}(\{h\}) \not\subseteq E_1 \ \Rightarrow\ j(h) > \theta_0)$,
- $\forall e \in E_1'\ (A_1'(F_1'(e)) \notin N_1 \ \vee\ A_1'(S_1'(e)) \notin N_1 \ \Rightarrow\ W_1'(A_1'(F_1'(e))) = W_1'(A_1'(S_1'(e))))$,
- $\forall n \in N_1' \backslash N_1 \ \ E_{con(p'(n))}(u'(W_1'(n))^{-1} \cdot u'(n)) < \theta_1$,
- the symmetry condition for $\mathcal{N}_1', u', v \circ p'$.

The *extension operations* are as follows (see figure 1 below). For each one, the extension conditions are checked at the end and the operation is cancelled if they do not hold. (Note that no consistent pair of inclusion functions (i',j') can be specified for the extended pattern $\mathcal{N}_1'$, so we recalculate them as in §7.4, using the old inclusion functions (i,j) as a starting point.)

(a) (Joining two symbols.) Given symbols $\sigma_1, \sigma_2 \in \Sigma_1$, an edge $e_0 \in E_0$, and affine transformations $s_1 \in symm(p(\sigma_1))$ and $s_2 \in symm(p(\sigma_2))$ such that

- $P_0(A_0(F_0(e_0))) = p(\sigma_1)$ and $P_0(A_0(S_0(e_0))) = p(\sigma_2)$,
- $E_{rel(e_0)}(s_2^{-1} \cdot u(\sigma_2)^{-1} \cdot u(\sigma_1) \cdot s_1) < \theta_2$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ (if one exists) such that $\mathcal{N}_1'$ contains nodes n_1, n_2 and an edge e for which

- $p'(e) = e_0$, $A_1'(F_1'(e)) = n_1$, $A_1'(S_1'(e)) = n_2$, $P_1'(n_1) = \sigma_1$ and $P_1'(n_2) = \sigma_2$,
- $u'(n_1) = u(\sigma_1) \cdot s_1$ and $u'(n_2) = u(\sigma_2) \cdot s_2$.

(b) (Joining three symbols.) Given symbols $\sigma_1, \sigma_2, \sigma_3 \in \Sigma_1$, edges $e_{01}, e_{02} \in E_0$, $s_1 \in symm(p(\sigma_1))$, $s_2 \in symm(p(\sigma_2))$ and $s_3 \in symm(p(\sigma_3))$ such that

- $A_0(S_0(e_{01})) = A_0(F_0(e_{02}))$, $P_0(A_0(F_0(e_{01}))) = p(\sigma_1)$, $P_0(A_0(S_0(e_{01}))) = p(\sigma_2)$ and $P_0(A_0(S_0(e_{02})))$ $= p(\sigma_3)$,
- $E_{rel(e_{01})}(s_2^{-1} \cdot u(\sigma_2)^{-1} \cdot u(\sigma_1) \cdot s_1) + E_{rel(e_{02})}(s_3^{-1} \cdot u(\sigma_3)^{-1} \cdot u(\sigma_2) \cdot s_2) < \theta_3$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ (if one exists) such that $\mathcal{N}_1'$ contains nodes n_1, n_2, n_3 and edges e_1, e_2 for which

- $p'(e_1) = e_{01}$, $p'(e_2) = e_{02}$, $A_1'(F_1'(e_1)) = n_1$, $A_1'(S_1'(e_1)) = n_2 = A_1'(F_1'(e_2))$, $A_1'(S_1'(e_2)) = n_3$, $P_1'(n_1) = \sigma_1$, $P_1'(n_2) = \sigma_2$ and $P_1'(n_3) = \sigma_3$,
- $u'(n_1) = u(\sigma_1) \cdot s_1$, $u'(n_2) = u(\sigma_2) \cdot s_2$ and $u'(n_3) = u(\sigma_3) \cdot s_3$.

(There are also variations of operation (b), in which the directions of e_{01} and e_1 are reversed, or the directions of e_{02} and e_2 are reversed.)

(c) (Extending from a hook.) Given a hook $h \in H_1$, construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that

- $p'(F_1'^{-1}(\{h\})) = F_0^{-1}(p(\{h\}))$ and $p'(S_1'^{-1}(\{h\})) = S_0^{-1}(p(\{h\}))$,
- $\forall e \in E_1' \backslash E_1 \;\; E_{rel(p'(e))}(u'(A_1'(S_1'(e)))^{-1} \cdot u'(A_1'(F_1'(e)))) = 0$.

(d) (Extending from a facet.) Given a facet $k \in K_1$ such that

- $W_1(A_1(F_1(C_1(k)))) = W_1(A_1(S_1(C_1(k))))$
- $j(k) > \theta_0$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that

- $\overline{p'} \circ G_1'^{-1} \circ id_{\{k\}} = G_0^{-1} \circ \overline{p} \circ id_{\{k\}}$.

(e) (Extending from a part to a whole.) Given a symbol $\sigma \in \Sigma_1$, construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that

- $p'(P_1'^{-1}(\{\sigma\})) = P_0{}^{-1}(p(\{\sigma\}))$,
- $\forall n \in N_1' \backslash N_1 \; E_{con(p'(n))}(u'(W_1'(n))^{-1} \cdot u'(n)) = 0$.

(f) (Filling in a missing part between two parts.) Given nodes $n_1, n_3 \in N_1$, hooks $h_1, h_3 \in H_1$, edges $e_{01}, e_{02} \in E_0$ and an affine transformation f such that

- $W_1(n_1) = W_1(n_3)$, $A_1(h_1) = n_1$, $A_1(h_3) = n_3$, $A_0(S_0(e_{01})) = A_0(F_0(e_{02}))$, $F_0(e_{01}) = p(h_1)$ and $S_0(e_{02}) = p(h_3)$,
- $E_{rel(e_{01})}(f^{-1} \cdot u(n_1)) + E_{rel(e_{02})}(u(n_3)^{-1} \cdot f) < \theta_4$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ (if one exists) such that $\mathcal{N}_1'$ contains edges e_1, e_2 and a node n_2 for which

- $p'(e_1) = e_{01}$, $p'(e_2) = e_{02}$, $F_1'(e_1) = h_1$, $A_1'(S_1'(e_1)) = n_2 = A_1'(F_1'(e_2))$ and $S_1'(e_2) = h_3$,
- if $n_2 \notin N_1$ then $u'(n_2) = f$.

(There are also variations of operation (f), in which the directions of e_{01} and e_1 are reversed, or the directions of e_{02} and e_2 are reversed.)

(g) (Filling in a symbol between part and whole.) Given symbols $\sigma_1, \sigma_3 \in \Sigma_1$, symbols $\sigma_{01}, \sigma_{02}, \sigma_{03} \in \Sigma_0$, nodes $n_{01}, n_{02} \in N_0$, $s \in symm(\sigma_{03})$, and an affine transformation f such that

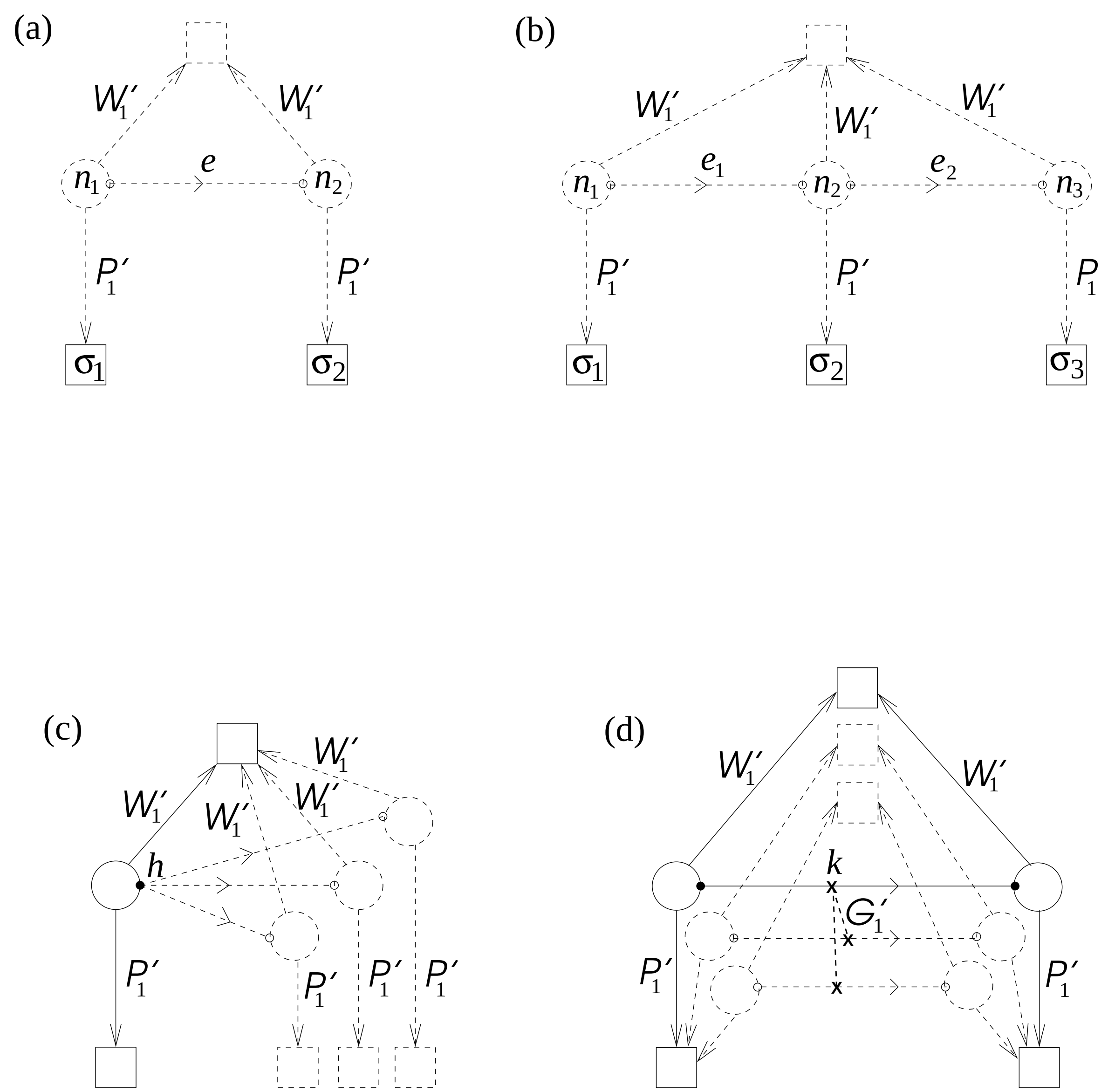
- $p(\sigma_1) = \sigma_{01}$, $p(\sigma_3) = \sigma_{03}$, $W_0(n_{01}) = \sigma_{01}$, $P_0(n_{01}) = \sigma_{02} = W_0(n_{02})$ and $P_0(n_{02}) = \sigma_{03}$,
- $E_{con(n_{01})}(u(\sigma_1)^{-1} \cdot f) + E_{con(n_{02})}(f^{-1} \cdot u(\sigma_3) \cdot s) < \theta_4$,

construct a minimal extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that $\mathcal{N}_1'$ contains a symbol σ_2 and nodes n_1, n_2 for which

- $p'(n_1) = n_{01}$, $p'(n_2) = n_{02}$, $W_1'(n_1) = \sigma_1$, $P_1'(n_1) = \sigma_2 = W_1(n_2)$ and $P_1'(n_2) = \sigma_3$,
- if $n_1 \notin N_1$ then $u'(n_1) = u'(\sigma_2) = f$,
- if $n_2 \notin N_1$ then $u'(n_2) = u(\sigma_3) \cdot s$.

These extension operations are applied concurrently, in a fair order, controlled by probabilities; the probability is low in cases where new symbols would be created (particularly operations (a) and (e)), to avoid the creation of too many new symbols.

These operations are depicted in figure 1. In this figure rectangles represent symbols, circles represent nodes, small filled discs represent hooks, lines with arrowheads halfway along represent edges, and small crosses represent facets (shown only for operation (e)); the W and P functions, which map each node to the whole and part symbols, are represented by lines with arrowheads at the end. For each operation solid lines are used for the symbols, nodes, etc., assumed to be present in the pattern before the extension operation; dashed lines are used for the symbols, nodes, etc., added by the extension operation (if they are not already present in $\mathcal{N}_1$). Thus, for example, operation (g) adds one symbol and two nodes (and their associated hooks), unless a suitable symbol or suitable nodes already exist in $\mathcal{N}_1$. Note, however, that whenever an edge is added the appropriate number of superedges must also be added, in order that p' satisfy the conditions for a homomorphism; and the same applies to symbols, nodes, etc.; these are not shown in the figure.



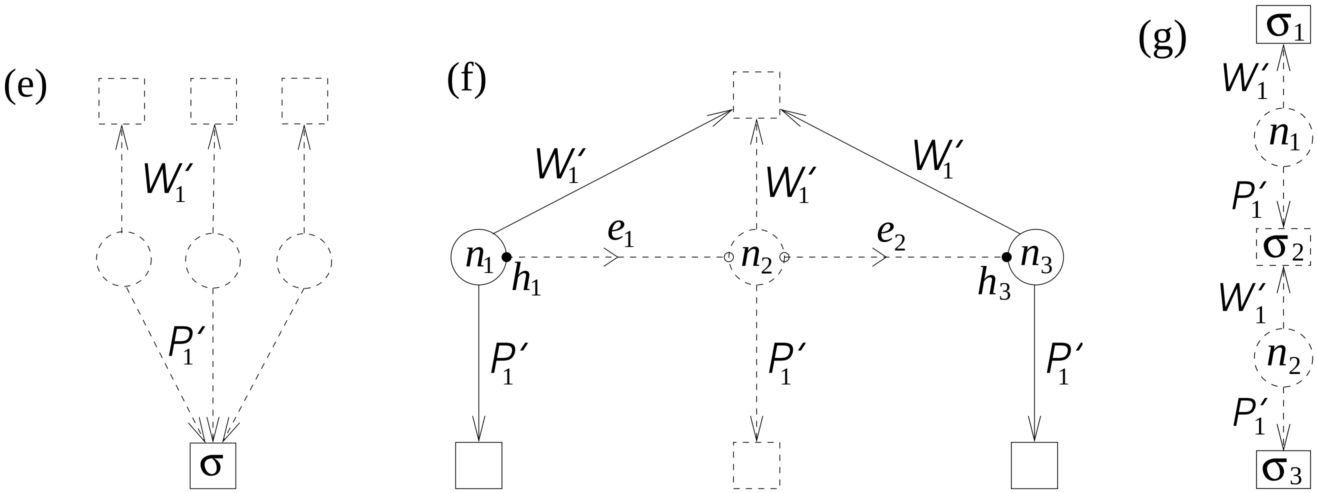


Figure 1. The extension operations. The figure shows the relevant parts of the pattern $\mathcal{N}_1'$ after each extension operation. Solid lines depict what must be present before the operation; dashed lines depict what is added if not already present.

8.5 Inextendability and its consequences

Definition. $(\mathcal{N}_1, p, u)$ is *inextendable*, given i,j,v,B, iff none of the extension operations can be applied to it, other than ones giving a trivial extension.

Theorem 45. If

(a) $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is a network with an embedding token u and a bareness function B,

(b) $p{:}\mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, where $\mathcal{N}_0$ is a semi-definite network with an embedding type v,

(c) (i,j) is a pair of inclusion functions on $\mathcal{N}_1$ satisfying $\forall x \in \Sigma_1 \cup N_1\ i(x) \in \{0,1\}$ and $\forall e \in E_1\ i(e) = 1$,

(d) $(\mathcal{N}_1, p, u)$ is inextendable, given i,j,v,B,

(e) $W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1$,

then $\forall h \in H_1\ j(h) = 0$ and $\forall k \in K_1 \setminus \mathrm{dom}(G_1)\ j(k) = 0$.

Proof. For any $h \in H_1$, we have

$$i(A(h)) = j(h) \ + \sum_{e \in {F_1}^{-1}(\{h\})} i(e) \ + \sum_{e \in {S_1}^{-1}(\{h\})} i(e). \tag{1}$$

This implies that $j(h)$ is an integer, and hence must be 0 or 1. Suppose $j(h) = 1$. Then the operation of extending from h (extension operation (c)) is possible, giving an extension $(\mathcal{N}_1', p', u')$ such that $p'({F_1'}^{-1}(\{h\})) = F_0^{-1}(p(\{h\}))$ and $p'({S_1'}^{-1}(\{h\})) = S_0^{-1}(p(\{h\}))$. However, we are assuming $(\mathcal{N}_1, p, u)$ is inextendable, so $(\mathcal{N}_1', p', u') = (\mathcal{N}_1, p, u)$. Thus $p(F_1^{-1}(\{h\})) = F_0^{-1}(p(\{h\}))$ and $p(S_1^{-1}(\{h\})) = S_0^{-1}(p(\{h\}))$.

Now, $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ is semi-definite and so satisfies $F_0(E_0) \cup S_0(E_0) = H_0$. This means that $F_0^{-1}(p(\{h\})) \neq \emptyset$ or $S_0^{-1}(p(\{h\})) \neq \emptyset$. Hence $p(F_1^{-1}(\{h\})) \neq \emptyset$ or $p(S_1^{-1}(\{h\})) \neq \emptyset$. Hence $F_1^{-1}(\{h\}) \neq \emptyset$ or $S_1^{-1}(\{h\}) \neq \emptyset$. Any edge e in these sets has $i(e) = 1$, by hypothesis (c), so by (1) $i(A(h)) \geq 2$, which is impossible. This contradiction establishes that $j(h) = 0$ as required.

For any $k \in K_1 \setminus \mathrm{dom}(G_1)$, we have

$$i(C(k)) = j(k) + \sum_{k^* | G_1(k,k^*)} i(C(k^*)).$$

This implies that $j(k)$ is an integer, and hence must be 0 or 1. Suppose $j(k) = 1$. Then $\sum_{k^* | G_1(k,k^*)} i(C(k^*)) = 0$, and hence by hypothesis (c) there are no k^* such that $G_1(k,k^*)$ holds. This can be expressed as

$$G_1^{-1} \circ id_{\{k\}} = \perp. \tag{2}$$

Also, since $k \notin \mathrm{dom}(G_1)$ we have

$$G_1 \circ id_{\{k\}} = \perp. \tag{3}$$

Now, by hypothesis (e) and since $j(k) = 1$, the operation of extending from k (extension operation (d)) is possible, giving an extension $(\mathcal{N}_1', p', u')$ such that $\overline{p'} \circ {G_1'}^{-1} \circ id_{\{k\}} = G_0^{-1} \circ \overline{p} \circ id_{\{k\}}$. However, we are assuming $(\mathcal{N}_1, p, u)$ is inextendable, hence $(\mathcal{N}_1', p', u') = (\mathcal{N}_1, p, u)$. Thus

$$\overline{p} \circ G_1^{-1} \circ id_{\{k\}} = G_0^{-1} \circ \overline{p} \circ id_{\{k\}}. \tag{4}$$

Since $\mathcal{N}_0$ is semi-definite, G_0 is minimal relative to $\mathcal{N}_0$ and hence $id_{K_0} \subseteq G_{0K} \circ G_{0K}^{-1} \cup G_{0K}^{-1} \circ G_{0K}$. Then

$$\begin{aligned}
\overline{p} \circ id_{\{k\}} &\subseteq G_{0K} \circ G_{0K}^{-1} \circ \overline{p} \circ id_{\{k\}} \ \cup\ G_{0K}^{-1} \circ G_{0K} \circ \overline{p} \circ id_{\{k\}} \\
&\subseteq G_0 \circ G_0^{-1} \circ \overline{p} \circ id_{\{k\}} \ \cup\ G_0^{-1} \circ G_0 \circ \overline{p} \circ id_{\{k\}} \\
&= G_0 \circ \overline{p} \circ G_1^{-1} \circ id_{\{k\}} \ \cup\ G_0^{-1} \circ \overline{p} \circ G_1 \circ id_{\{k\}} && \text{by (4) and since } p \text{ is a homomorphism} \\
&= \perp \cup \perp = \perp && \text{by (2) and (3).}
\end{aligned}$$

But this is absurd, since $(\overline{p} \circ id_{\{k\}})(p(k), k)$ holds. This contradiction establishes that $j(k) = 0$, as required. ∎

8.6 Merging two symbol tokens

We may merge two symbol tokens of the same type that have similar embedding transformations (up to a symmetry transformation). Two symbol tokens $\sigma_1, \sigma_2 \in \Sigma_1$ of type $\sigma_0 \in \Sigma_0$ are considered to have similar embeddings up to a symmetry transformation iff there exists $s \in symm(\sigma_0)$ such that $E_{in(\sigma_0)}(s^{-1} \cdot u(\sigma_2)^{-1} \cdot u(\sigma_1))$ is below a threshold. If this condition holds then the symmetry s is applied to σ_2 (this is a local symmetry operation; the other symbols are unchanged); σ_1 and σ_2 are replaced by a single symbol; and the nodes and edges of σ_1 and σ_2 are pooled. Local symmetry operations have already been defined in §4.3. The merging itself is formally defined as follows.

Definition. Given a network $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$, and a homomorphism $p: \mathcal{N}_1 \to \mathcal{N}_0$, and two symbols $\sigma_1, \sigma_2 \in \Sigma_1$ such that $p(\sigma_1) = p(\sigma_2)$, the operation of *merging* σ_1 *and* σ_2 produces a new network $\mathcal{N}_1' = (\Sigma_1', N_1', H_1', E_1', K_1', W_1', P_1', A_1', F_1', S_1', C_1', G_1')$ and a homomorphism $\mu: \mathcal{N}_1 \to \mathcal{N}_1'$ defined as follows. First note that there are bijections $p|_{A_1^{-1}(\{\sigma_1\})}: {A_1}^{-1}(\{\sigma_1\}) \to {A_0}^{-1}(\{p(\sigma_1)\})$ and and $p|_{A_1^{-1}(\{\sigma_2\})}: {A_1}^{-1}(\{\sigma_2\}) \to {A_0}^{-1}(\{p(\sigma_1)\})$ and hence a bijection $\gamma = p|^{-1}_{A_1^{-1}(\{\sigma_1\})} \circ p|_{A_1^{-1}(\{\sigma_2\})}: {A_1}^{-1}(\{\sigma_2\}) \to {A_1}^{-1}(\{\sigma_1\})$. Now define

$\Sigma_1' = \Sigma_1 \setminus \{\sigma_2\}$, $N_1' = N_1$, $H_1' = H_1 \setminus A_1{}^{-1}(\{\sigma_2\})$, $E_1' = E_1$, $K_1' = K_1$,
$\forall \sigma \in \Sigma_1\ \mu(\sigma) = \begin{cases} \sigma_1 & \text{if } \sigma = \sigma_2 \\ \sigma & \text{otherwise} \end{cases}$, $\forall h \in H_1\ \mu(h) = \begin{cases} \gamma(h) & \text{if } A_1(h) = \sigma_2 \\ h & \text{otherwise} \end{cases}$,
$\forall x \in N_1 \cup E_1 \cup K_1\ \mu(x) = x$
$\forall \sigma \in \Sigma_1'\ W_1'(\sigma) = \sigma$, $\forall n \in N_1'\ W_1'(n) = \mu(W_1(n))$,
$P_1' = \mu \circ P_1$, $A_1' = A_1|_{H_1'}$, $F_1' = \mu \circ F_1$, $S_1' = \mu \circ S_1$, $C_1' = C_1$,
$G_1' = \overline{\mu} \circ G_1 \circ \overline{\mu}^{-1}$.

This gives the network $\mathcal{N}_1'$ and homomorphism $\mu: \mathcal{N}_1 \to \mathcal{N}_1'$.

Note that μ is surjective and satisfies $\forall x, y \in \Sigma_1 \cup N_1 \cup H_1 \cup E_1 \cup K_1\ (\mu(x) = \mu(y) \Rightarrow p(x) = p(y))$, so there exists a unique function p' such that $p' \circ \mu = p$; this p' is a homomorphism from $\mathcal{N}_1'$ to $\mathcal{N}_0$. This is the new parse.

If u is the old embedding token on $\mathcal{N}_1$, we can define a new embedding token u' on $\mathcal{N}_1'$ by

$$\forall \sigma \in \Sigma_1'\ u'(\sigma) = \begin{cases} G & \text{if } \sigma = \sigma_1 \\ u(\sigma) & \text{otherwise} \end{cases}, \quad \forall n \in N_1'\ u'(n) = u(n)$$

where G is an affine transformation intermediate between $u(\sigma_1)$ and $u(\sigma_2) \cdot s$. This is calculated as follows. Find $A \in \mathcal{A}$ such that $u(\sigma_2) \cdot s \cdot u(\sigma_1)^{-1} = \exp(A)$; then choose $G = \exp(A/2) \cdot u(\sigma_1) = \exp(-A/2) \cdot u(\sigma_2) \cdot s$.

No consistent pair of inclusion functions (i', j') can be specified for the new network $\mathcal{N}_1'$, so we recalculate them as in §7.4, using the restriction of the old inclusion functions (i, j) to $\mathcal{N}_1'$ as a starting point.

8.7 Partitioning a symbol token into two

Consider any pattern $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$. For any $\sigma_0 \in \Sigma_1$, if the nodes and symbol in $W_1{}^{-1}(\{\sigma_0\})$ can be partitioned into two disjoint non-empty subsets T_1, T_2, such that there is no edge between any element T_1 and any element of T_2, then σ_0 may be replaced by two symbols, $(\sigma_0, 1)$ and $(\sigma_0, 2)$, with $(\sigma_0, 1)$ getting the nodes of T_1 and $(\sigma_0, 2)$ getting the nodes of T_2. The subsymbols of σ_0 are glued to $(\sigma_0, 1)$ or $(\sigma_0, 2)$ as appropriate (or duplicated if necessary). The nodes above σ_0 must be duplicated; i.e., n is replaced by (n, i) and (n, j), for two indices i, j, and likewise for their hooks, edges and facets. This operation is called *partitioning* σ_0. It is roughly the inverse of the merging operation.

Definition. In a network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, a symbol $\sigma_0 \in \Sigma$ is *partitionable* iff there exist sets T_1, T_2 such that $T_1 \cup T_2 = W^{-1}(\{\sigma_0\})$ and $T_1 \cap T_2 = \emptyset$ and $T_1, T_2 \neq \emptyset$ and $F^{-1}(A^{-1}(T_1)) \cap S^{-1}(A^{-1}(T_2)) = \emptyset$ and $S^{-1}(A^{-1}(T_1)) \cap F^{-1}(A^{-1}(T_2)) = \emptyset$.

The network $(\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is *partitionable* iff at least one symbol in Σ is partitionable; otherwise it is *unpartitionable*.

Definition. If a symbol σ_0 is partitionable in $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$, with sets T_1, T_2, then *partitioning* σ_0 is the construction of a new network $\mathcal{N}'$ as follows.

First define a function I on $\Sigma \cup N \cup E$ by

- $I(\sigma_0) = \{1, 2\}$,
- $\forall n \in N \cap T_1\ (I(n) = \{1\} \times I(P(n)) \wedge \forall n^*\ (G(n, n^*) \Rightarrow I(n^*) = I(n)))$
- $\forall n \in N \cap T_2\ (I(n) = \{2\} \times I(P(n)) \wedge \forall n^*\ (G(n, n^*) \Rightarrow I(n^*) = I(n)))$
- $\forall \sigma^*\ (G(\sigma_0, \sigma^*) \Rightarrow I(\sigma^*) = \bigcup_{n \in N \cap W^{-1}(\{\sigma^*\})} proj_1(I(n)))$

- for all other $\sigma \in \Sigma$, $I(\sigma) = \{0\}$,
- for all other $n \in N$, $I(n) = I(W(n)) \times I(P(n))$,
- $\forall e \in E\ I(e) = I(A(F(e))) \times I(A(S(e)))$,

where the function $proj_1$ is defined by $\forall (i,j)\ proj_1(i,j) = i$.

Define the new network $\mathcal{N}' = (\Sigma', N', H', E', K', W', P', A', F', S', C', G')$, by

- $\Sigma' = \{ (\sigma, i) \mid \sigma \in \Sigma\ \wedge\ i \in I(\sigma) \}$, $N' = \{ (n, i) \mid n \in N\ \wedge\ i \in I(n) \}$,
- $H' = \{ (h, i) \mid h \in H\ \wedge\ i \in I(A(h)) \}$, $E' = \{ (e, i) \mid e \in E\ \wedge\ i \in I(e) \}$,
- $K' = \{ (k, i) \mid k \in K\ \wedge\ i \in I(C(k)) \}$,
- $\forall (\sigma, i) \in \Sigma'\ W'(\sigma, i) = (\sigma, i)$, $\forall (n, (i,j)) \in N'\ W'(n, (i,j)) = (W(n), i)$,
- $\forall (n, (i,j)) \in N'\ P'(n, (i,j)) = (P(n), j)$, $\forall (h, i) \in H'\ A'(h, i) = (A(h), i)$,
- $\forall (e, (i,j)) \in E'\ F'(e, (i,j)) = (F(e), i)$, $\forall (e, (i,j)) \in E'\ S'(e, (i,j)) = (S(e), j)$,
- $\forall (k, i) \in K'\ C'(k, i) = (C(k), i)$,
- $\forall (\sigma, i), (\sigma^*, j) \in \Sigma'\ (G'((\sigma, i), (\sigma^*, j))\ \text{ iff }\ G(\sigma, \sigma^*)\ \wedge\ c(i,j))$,
- $\forall (n, i), (n^*, j) \in N'\ (G'((n, i), (n^*, j))\ \text{ iff }\ G(n, n^*)\ \wedge\ c_2(i,j))$,
- $\forall (h, i), (h^*, j) \in H'\ (G'((h, i), (h^*, j))\ \text{ iff }\ G(h, h^*)\ \wedge\ c_2(i,j))$,
- $\forall (e, i), (e^*, j) \in E'\ (G'((e, i), (e^*, j))\ \text{ iff }\ G(e, e^*)\ \wedge\ c_4(i,j))$,
- $\forall (k, i), (k^*, j) \in K'\ (G'((k, i), (k^*, j))\ \text{ iff }\ G(k, k^*)\ \wedge\ c_4(i,j))$,

where

$$\begin{array}{rcl} c(i,j) & \text{iff} & i = j\ \vee\ i = 0 \\ c_2((i,j),(k,l)) & \text{iff} & c(i,k)\ \wedge\ c(j,l) \\ c_4((i,j),(k,l)) & \text{iff} & c_2(i,k)\ \wedge\ c_2(j,l). \end{array}$$

Define a homomorphism $\pi: \mathcal{N}' \to \mathcal{N}$ by $\forall (x, i) \in \Sigma' \cup N' \cup H' \cup E' \cup K'\ \pi(x, i) = x$.

During the recognition process any partitionable symbol in the pattern $\mathcal{N}_1$ may be partitioned, producing a new network $\mathcal{N}_1'$, with a homomorphism $\pi: \mathcal{N}_1' \to \mathcal{N}_1$. The parse $p: \mathcal{N}_1 \to \mathcal{N}_0$ becomes $p \circ \pi: \mathcal{N}_1' \to \mathcal{N}_0$ and the embedding token u becomes $u \circ \pi$. The new pair of inclusion functions is $(i \circ \pi, j \circ \pi)$.

Theorem 46. If $\mathcal{N} = (\Sigma, N, H, E, K, W, P, A, F, S, C, G)$ is an unpartitionable network satisfying the condition $W \circ A \circ F = W \circ A \circ S$ then $E \overset{A \circ F}{\underset{A \circ S}{\rightrightarrows}} N \cup \Sigma \xrightarrow{W} \Sigma$ is a coequaliser diagram in the category of sets.

Proof. Consider any set X and function $\alpha: N \cup \Sigma \to X$ such that $\alpha \circ A \circ F = \alpha \circ A \circ S$. We must show that there exists a unique function $\mu: \Sigma \to X$ such that $\alpha = \mu \circ W$.

Define $\mu = \alpha|_\Sigma$. Consider any symbol $\sigma \in \Sigma$.

Define $T_1 = \{ x \in W^{-1}(\{\sigma\}) \mid \alpha(x) = \alpha(\sigma) \}$ and $T_2 = \{ x \in W^{-1}(\{\sigma\}) \mid \alpha(x) \neq \alpha(\sigma) \}$. Then $T_1 \cup T_2 = W^{-1}(\{\sigma\})$ and $T_1 \cap T_2 = \emptyset$. Note also that $\sigma \in T_1$, so $T_1 \neq \emptyset$.

Since $W \circ A \circ F = W \circ A \circ S$ and $\alpha \circ A \circ F = \alpha \circ A \circ S$ we have, for any $e \in E$,

$$\begin{array}{ll} e \in F^{-1}(A^{-1}(T_1)) & \text{iff}\quad A(F(e)) \in T_1 \quad \text{iff}\quad W(A(F(e))) = \sigma\ \wedge\ \alpha(A(F(e))) = \alpha(\sigma) \\ & \text{iff}\quad W(A(S(e))) = \sigma\ \wedge\ \alpha(A(S(e))) = \alpha(\sigma) \quad \text{iff}\quad A(S(e)) \in T_1 \\ & \text{iff}\quad e \in S^{-1}(A^{-1}(T_1)) \end{array}$$

so $F^{-1}(A^{-1}(T_1)) = S^{-1}(A^{-1}(T_1))$, and similarly $F^{-1}(A^{-1}(T_2)) = S^{-1}(A^{-1}(T_2))$. Then

$$\begin{aligned} F^{-1}(A^{-1}(T_1)) \cap S^{-1}(A^{-1}(T_2)) &= F^{-1}(A^{-1}(T_1)) \cap F^{-1}(A^{-1}(T_2)) = F^{-1}(A^{-1}(T_1 \cap T_2)) \\ &= F^{-1}(A^{-1}(\emptyset)) = \emptyset \end{aligned}$$

and similarly $S^{-1}(A^{-1}(T_1)) \cap F^{-1}(A^{-1}(T_2)) = \emptyset$.

So all the conditions for σ to be partitionable have been verified, except $T_2 \neq \emptyset$. But σ is not partitionable, by hypothesis, so $T_2 = \emptyset$. This means that $\forall n \in W^{-1}(\{\sigma\})\ \alpha(n) = \alpha(\sigma) = \mu(W(n))$.

Since we have shown this for arbitrary $\sigma \in \Sigma$ we have $\forall n \in N \cup \Sigma\ \alpha(n) = \mu(W(n))$, i.e., $\alpha = \mu \circ W$, as required.

Conversely, consider any function $\mu : \Sigma \to X$ such that $\alpha = \mu \circ W$. Then $\forall \sigma \in \Sigma\ \alpha(\sigma) = \mu(W(\sigma)) = \mu(\sigma)$, so $\mu = \alpha|_\Sigma$. This shows that μ is unique.

This completes the verification of the coequaliser condition. ∎

8.8 The outcome of the recognition process

The recognition process must ensure that, at the end, the pattern $\mathcal{N}_1$, the homomorphism $p : \mathcal{N}_1 \to \mathcal{N}_0$, the inclusion functions i,j, and the embedding token u satisfy the conditions

- $\forall x \in \Sigma_1 \cup N_1 \cup E_1\ i(x) \in \{0, 1\}$,
- $\mathcal{N}_1$ is unprunable, given (i,j),
- $(\mathcal{N}_1, p, u)$ is inextendable, given i,j,v,B,
- $\mathcal{N}_1$ is unpartitionable,
- $W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1$,
- $\forall \sigma \in \Sigma_1 \backslash P_1(N_1)\ B(\sigma) = \theta$.

It will be shown in §9.6 that these conditions imply that $\forall x \in \Sigma_1 \cup N_1 \cup E_1\ i(x) = 1$ and $\mathcal{N}_1$ is definite; hence the recognition process is finished.

9. The Whole Recognition Process

9.1 Introduction

This section completes the account of recognition by filling in some missing details: the line operations, the determination of temperature, the determination of the bareness function, and the halting condition.

At the end of recognition the pattern should be definite and all the symbols', nodes' and edges' inclusion values should be 1. Previous sections have stated various sufficient conditions for this to be the case. Now I shall show that (under some mild assumptions) these conditions are indeed satisfied at the end of recognition, and that the definite match function *DM* is maximised.

9.2 Line operations

I shall assume the grammar contains a symbol type called *line*; symbol tokens of this type are called lines. Geometrically they are infinitely thin line segments, and their embeddings are always similarities. Parsing starts with a set of randomly arranged lines, symbols of other types being built up from there. There is no necessary reason to start with lines, but it is convenient for a wide range of examples, so I shall provide some special operations for dealing with lines.

(a) Create a line. A line symbol token σ is created; its initial embedding $u(\sigma)$ is chosen by a random search aiming to maximise $\rho_{I,T,sat}(u(\sigma))$. This operation is applied throughout recognition (though mostly at the start when most of the image is not covered by symbol tokens).

(b) Randomise a line. If a line's inclusion value falls too low, and it is bare (not part of another symbol token), a new random embedding transformation is chosen for it by the same method as in (a).

(c) Remove a bare line. If a line is not part of another symbol and its temperature falls below a threshold, it has a small probability of being removed. The purpose of this is to tidy up the pattern by removing lines that have not been incorporated into higher-level symbols.

(d) Glue two lines together, end to end. If two lines are nearly collinear, with an end of one line close to an end of the other, as measured by a fleximap, the two lines are replaced by a single large line. Their bars are connected together, end to end.

9.3 How temperature is determined

Every symbol token, node token, hook token, edge token and facet token has a temperature. These change continually by the following three processes. (There are two parameters controlling the spread of temperature across the network, $\beta = 0.3$ and $\gamma = 0.2$. The notation $X \Rightarrow Y$ means the assignment statement $Y := \max(X, Y)$, and $X \Leftrightarrow Y$ means $X \Rightarrow Y$ and $Y \Rightarrow X$.)

1. Every symbol, node, hook, edge or facet token created by a line operation or an extension operation is given a high initial temperature.

2. Temperature is spread through the pattern by applying the following operations periodically.

- For every hook $h \in H_1$, do $T_{A_1(h)} \Rightarrow T_h$; $\beta T_h \Rightarrow T_{A_1(h)}$.
- For every edge $e \in E_1$, do $T_{F_1(e)} \Leftrightarrow T_e$; $T_{S_1(e)} \Leftrightarrow T_e$.
- For every node $n \in N_1$, do
  $T_{P_1(n)} \Rightarrow T_n$; $\gamma T_n \Rightarrow T_{W_1(n)}$;
  if $n \in \mathrm{dom}(G_1)$ then $\gamma T_{W_1(n)} \Rightarrow T_n$ else $\gamma T_n \Rightarrow T_{P_1(n)}$.
- For every pair $k, k^* \in K_1$ such that $G_1(k, k^*)$, do $T_k \Leftrightarrow T_{k^*}$.
- For each $k \in K_1$, do $T_{C_1(k)} \Leftrightarrow T_k$.

3. Periodically each temperature T declines by the formula

$$T := a + \eta T$$

   where the constant η is slightly below 1 and the constant a is small and positive. In the absence of other temperature changes all temperatures will converge to $T_{min} = a/(1-\eta)$. The value of a is chosen so as to give a desired minimum temperature, T_{min}.

The temperatures perform a simulated annealing function. Wherever the structure of the pattern changes, temperature is increased, by process 1. These increases in temperatures are spread to neighbouring parts of the pattern, by process 2. In the absence of changes in the structure the temperature will decline, by process 3. The general effect of this is that rapidly changing areas of the pattern will be hot, and areas that have settled down will become cold.

Where the pattern is hot the inclusion values will take mid-range values; this allows rival grammatical possibilities to co-exist. Where the pattern is cold the inclusion values tend to be driven towards 0 or 1, and this forces a choice to be made between the rival grammatical possibilities, leading to a definite pattern.

As recognition finishes, structural and geometric changes cease and the temperature declines to T_{min} throughout the pattern. All the inclusion values are likely to approach 0 or 1. However, this is not guaranteed; it is possible to construct a network in which there is no consistent final assignment of 0 or 1 inclusion values, except for the trivial solution in which all inclusion values are 0 (consider, for example, the case of an edge that is connected to the same hook at both ends where the hook has no other incident edges). Nevertheless, in realistic cases the inclusion values do go to 0 or 1, and this will be assumed in what follows.

9.4 How the bareness function B is determined

For each $h \in H_1$, $B(h)$ is initially 0 and increases by a fixed amount every time an extension operation is applied that adds edges to $F_1{}^{-1}(\{h\}) \cup S_1{}^{-1}(\{h\})$.

For each $k \in K_1$, $B(k)$ is initially 0 and increases by a fixed amount every time an extension operation is applied that adds facets to $\{\, k^* \in K_1 \mid G_1(k,k^*) \,\}$.

For each $\sigma \in \Sigma_1$, if $P_0{}^{-1}(p(\{\sigma\})) = \emptyset$ then $B(\sigma)$ is initially set to θ (where θ is the constant used in the *DM* function), and never changes thereafter. If $P_0{}^{-1}(p(\{\sigma\})) \neq \emptyset$ then $B(\sigma)$ is initially set to a positive value $\theta_0 < \theta$; $B(\sigma)$ is increased by a fixed amount every time extension operation (a), (b), (e) or (g) is applied that adds nodes to $P_1{}^{-1}(\{\sigma\})$. The increment is chosen so that $\theta - \theta_0$ is a multiple of the increment.

The purpose of all this is to prevent the recognition algorithm from getting stuck in an infinite loop by trying the same thing over and over again. Suppose a hook h is bare, and then edges are added to it by the operation of extending from h, then the edges are removed by a pruning operation; and this is repeated indefinitely. The value of $B(h)$ will increase each time edges are added, making the algorithm more and more unwilling to remove all the edges and leave the hook bare. Eventually the algorithm will either keep an edge or remove the node $A(h)$ altogether; either way, the infinite loop is averted. Likewise, the monotonic increase in $B(k)$ prevents the algorithm from repeatedly adding and removing sub-facets to k forever.

The monotonic increase in $B(\sigma)$ prevents the algorithm from making any further attempts to add nodes above σ using extension operations (a), (b), (e) or (g), once $B(\sigma)$ has increased to θ. It follows that $B(\sigma)$ can never increase above θ. Throughout recognition we have

$$\forall \sigma \in \Sigma_1 \; B(\sigma) \leq \theta, \text{ with equality if } P_0{}^{-1}(p(\{\sigma\})) = \emptyset. \tag{1}$$

At the end of recognition all the $B(\sigma)$ values for bare symbol tokens σ should equal θ. This is ensured by the following theorem.

Theorem 47. If

(a) $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is a definite network satisfying $W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1$,

(b) $p\colon \mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, where $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ is a semi-definite network,

(c) (i,j) is a pair of inclusion functions on $\mathcal{N}_1$ satisfying $\forall x \in \Sigma_1 \cup N_1 \cup E_1 \; i(x) = 1$,

(d) u is an embedding token for $\mathcal{N}_1$ and v is an embedding type for $\mathcal{N}_0$,

(e) B is a bareness function for $\mathcal{N}_1$ satisfying $\forall \sigma \in \Sigma_1 \; B(\sigma) \leq \theta,$ with equality if $P_0{}^{-1}(p(\{\sigma\})) = \emptyset$,

(f) $(\mathcal{N}_1, p, u)$ is inextendable, given i, j, v, B,

then

$$\forall \sigma \in \Sigma_1 \backslash P_1(N_1) \; B(\sigma) = \theta.$$

Proof. Consider any $\sigma \in \Sigma_1 \backslash P_1(N_1)$. In view of hypotheses (a) and (c), and the fact that $\sigma \notin P_1(N_1)$, theorem 32 tells us that $j(\sigma) = 1$. Now, if $P_0{}^{-1}(p(\{\sigma\})) = \emptyset$ then $B(\sigma) = \theta$ by hypothesis (e). Suppose, on the other hand, $P_0{}^{-1}(p(\{\sigma\})) \neq \emptyset$. Then hypothesis (e) gives $B(\sigma) \leq \theta$. Suppose that $B(\sigma) < \theta$. Then, since $j(\sigma) = 1$, extension operation (e) is possible, giving an extension $(\mathcal{N}_1', p', u')$ of $(\mathcal{N}_1, p, u)$ such that $p'(P_1'{}^{-1}(\{\sigma\})) = P_0{}^{-1}(p(\{\sigma\}))$. However hypothesis (f) tells us that this is a trivial extension, so $(\mathcal{N}_1', p', u') = (\mathcal{N}_1, p, u)$. This means

$p({P_1}^{-1}(\{\sigma\})) = {P_0}^{-1}(p(\{\sigma\}))$. But this is impossible: the left-hand side is empty, since $\sigma \notin P_1(N_1)$, but the right-hand side is non-empty. This contradiction establishes that $B(\sigma) = \theta$, as required. ▮

9.5 Summary of the entire recognition process

The input to the parser is an image I, a semi-definite network $\mathcal{N}_0$ (representing a grammar), and an embedding type v for $\mathcal{N}_0$. During recognition, a network $\mathcal{N}_1$ (representing a pattern), a homomorphism $p\colon\mathcal{N}_1 \to \mathcal{N}_0$, a pair of inclusion functions (i,j), an embedding token u, and a bareness function B for $\mathcal{N}_1$ are constructed.

The recognition process is a sequence of steps, called *cycles*. In each cycle,

- i and j are recalculated (§7.4);
- u is adjusted (subject to the symmetry condition) to increase $IM(I,\mathcal{N}_1,u,v\circ p,i,j,B)$ (§6.5);
- all temperatures spread and decline a little (§9.3);
- structural operations are applied to $\mathcal{N}_1$ if the conditions are satisfied (elementary pruning operations, extension operations, merging two symbol tokens, and partitioning a symbol token into two); every new symbol, node, hook, edge or facet is given a high temperature (§9.3), and some bareness values are increased after an extension operation (§9.4);
- line operations are applied to $\mathcal{N}_1$ (§9.2).

The algorithm halts when

- no further structural operations are possible (except for trivial extensions);
- the temperatures have declined very close to the minimum T_{min}.

9.6 The outcome of recognition

At the end of recognition, condition (1) will hold (since it holds throughout recognition); also, the pattern $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ and the inclusion functions i,j are likely to satisfy the condition

$$\forall x \in \Sigma_1 \cup N_1 \cup E_1 \; i(x) \in \{0,1\}. \tag{2}$$

This is likely because the lowering of temperature at the end of recognition pushes all inclusion values to 0 or 1.

Also, the condition

$$W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1. \tag{3}$$

is likely to hold. This is because the term $E_{in(p(W_1(A_1(F_1(e)))))}(u(W_1(A_1(S_1(e))))^{-1} \cdot u(W_1(A_1(F_1(e)))))$ in $IM(I,\mathcal{N}_1,u,v\circ p,i,j,B)$ penalises any edge $e \in E_1$ for which $W_1(A_1(F_1(e))) \neq W_1(A_1(S_1(e)))$. The penalty is large and increases in effect as temperature falls. This term generates a force that pulls the two symbols $W_1(A_1(F_1(e)))$ and $W_1(A_1(S_1(e)))$ closer together; eventually, either they will merge or $i(e)$ will fall so low that e is pruned.

Conditions (2) and (3) are not guaranteed to hold, but they can be expected to hold in practice. Assuming they do hold we can apply the following theorem.

Theorem 48. If

(a) $\mathcal{N}_1 = (\Sigma_1, N_1, H_1, E_1, K_1, W_1, P_1, A_1, F_1, S_1, C_1, G_1)$ is a network satisfying $W_1 \circ A_1 \circ F_1 = W_1 \circ A_1 \circ S_1$,
(b) $p{:}\mathcal{N}_1 \to \mathcal{N}_0$ is a homomorphism, where $\mathcal{N}_0 = (\Sigma_0, N_0, H_0, E_0, K_0, W_0, P_0, A_0, F_0, S_0, C_0, G_0)$ is a semi-definite network,
(c) (i,j) is a pair of inclusion functions on $\mathcal{N}_1$ satisfying $\forall x \in \Sigma_1 \cup N_1 \cup E_1\ i(x) \in \{0,1\}$,
(d) u is an embedding token for $\mathcal{N}_1$ and v is an embedding type for $\mathcal{N}_0$,
(e) B is a bareness function for $\mathcal{N}_1$ satisfying $\forall \sigma \in \Sigma_1\ B(\sigma) \leq \theta,$ with equality if $P_0{}^{-1}(p(\{\sigma\})) = \emptyset$,
(f) $\mathcal{N}_1$ is unprunable, given (i,j),
(g) $(\mathcal{N}_1, p, u)$ is inextendable, given i,j,v,B,
(h) $\mathcal{N}_1$ is unpartitionable,

then

(1) $\mathcal{N}_1$ is definite,
(2) $\forall x \in \Sigma_1 \cup N_1 \cup E_1\ i(x) = 1$,
(3) $IM(I, \mathcal{N}, u, v, i, j, B) = DM(I, \mathcal{N}, u, v)$.

Proof. Theorem 46, using hypotheses (a,h), tells us that
(i) $E_1 \overset{A_1 \circ F_1}{\underset{A_1 \circ S_1}{\rightrightarrows}} N_1 \cup \Sigma_1 \overset{W_1}{\to} \Sigma_1$ is a coequaliser diagram in the category of sets.

Theorem 42, using hypotheses (c,f), gives
(j) $\forall e \in E_1\ i(e) = 1$.

So theorem 45, using hypotheses (a,c,g) and (j), gives
(k) $\forall h \in H_1\ j(h) = 0$ and $\forall k \in K_1 \setminus \mathrm{dom}(G_1)\ j(k) = 0$.

Then theorem 44, using hypotheses (c,f) and (i,k), gives conclusions (1,2).

Theorem 47, using hypotheses (a,e,g) and (1,2), then gives
(l) $\forall \sigma \in \Sigma_1 \setminus P_1(N_1)\ B(\sigma) = \theta$.

Conclusion (3) then follows by theorem 33, using (l) and (1,2). ∎

As explained in §6.6 and §7.5, the algorithm has attempted throughout to choose i,j to maximise E and to choose u to maximise $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$; as recognition ends this amounts to choosing i,j,u to maximise $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$, and this amounts to maximising $DM(I, \mathcal{N}_1, u, v \circ p)$ at the end of recognition. Thus the recognition problem is solved.

It should be noted that there is a number of ways in which recognition can fail. The core of the theory is provably correct, but the more peripheral parts of the algorithm are supported only by heuristic arguments (which I have indicated throughout by use of the word 'likely').

(i) The algorithm for determining the inclusion functions (§7.4) is not guaranteed to halt. It finds only a local maximum of E, not a global maximum.
(ii) Even for low temperature, maximising E is not precisely the same as maximising $IM(I, \mathcal{N}_1, u, v \circ p, i, j, B)$.
(iii) The monotonic raising of the bareness function may cut off possible structural extensions prematurely.
(iv) Symbol tokens may be merged or not merged, or partitioned or not partitioned, wrongly. (Errors in merging can be corrected by partitioning and vice versa.)
(v) The whole algorithm is not guaranteed to halt.
(vi) It is not guaranteed that very low temperature will force all inclusion values to 0 or 1.

(vii) Incoherent edges may survive to the end.

(viii) Some portions of the image may have been overlooked.

In practice it is only (iii), (iv), (v) and (viii) that matter.

Summary of Notation and Terminology

Logical notation used

$\wedge$, $\vee$, $\Rightarrow$, iff , $\neg$, $\forall$, $\exists$ – ‘and’, ‘or’, ‘implies’, ‘iff’, ‘not’, ‘for all’ and ‘there exists’ (note that $\forall$, $\exists$ and $\neg$ have highest precedence, followed by $\wedge$ and $\vee$, followed by $\Rightarrow$ and iff)

Concepts from category theory presupposed

the category of sets – its objects are sets and its morphisms are functions

sum, pullback, coequaliser – limits and colimits in the category of sets

§2. Functions and Relations

function – §2.2

dom(f), ran(f) – the domain and range of function f – §2.2

$f(A)$, $f^{-1}(B)$ – the image and preimage of a set under a function – §2.2

$f|_A$ – the restriction of a function f to a set A – §2.2

$f \circ g$ – the composition of functions f and g – §2.2

(binary) relation – §2.2

dom(R), ran(R) – the domain and range of a relation R – §2.2

id_A – the identity relation on a set A – §2.2

$\perp$ – the empty relation – §2.2

$R \subseteq S$ – R is a sub-relation of S (the same symbol is used for the subset relation) – §2.2

$R \subset S$ – R is a proper sub-relation of S (the same symbol is used for the proper subset relation) – §2.2

$R \cap S$, $R \cup S$, $R \backslash S$ – the intersection, union and difference of two relations (or sets) – §2.2

$R \circ S$ – the composition of two relations – §2.2

R^{-1} – the inverse of a relation R – §2.2

$\overline{f}$ – the graph of a function f – §2.2

finite relation – a relation satisfied by finitely many pairs – §2.3

NE – a relation such that $NE(x,y)$ iff $x \neq y$ – §2.3

acyclic relation – §2.3

connected relation relative to a function – §2.3

minimal relation relative to a function – §2.4

§3. Networks and Homomorphisms

network (symbols, nodes, hooks, edges, facets, gluing relation, subsymbol, subnode, subhook, subedge, subfacet, supersymbol, supernode, superhook, superedge, superfacet) – §3.2

homomorphism (from a network to another) – §3.2

f^{-1} – inverse homomorphism of f – §3.2

isomorphism – an invertible homomorphism – §3.2

automorphism – an isomorphism from a network to itself – §3.2

minimal relation relative to a network – §3.3

semi-definite network – the grammar is a semi-definite network – §3.4

definite network – the pattern is definite at the end of recognition – §3.4

subnetwork (and proper subnetwork) – §3.7

the inclusion homomorphism (from a subnetwork to the whole network) – §3.7

$p|_{\mathcal{N}'}$ – the restriction of a homomorphism p to a subnetwork $\mathcal{N}'$ – §3.7

§4. Embeddings and the Definite Match Function

$\mathcal{G}$ – the group of all affine transformations on the plane – (2004, §3)

$g \cdot g'$ – the composition of two affine transformations – (2004, §3)

E_τ – the 'energy' or penalty function for a fleximap τ – (2004, §5.3)

u – an embedding token – §4.2

$u_1 \cdot u_2$ – the product of two embedding tokens – §4.2

$u \circ f$ – the embedding token induced by a homomorphism f from an embedding token u – §4.2

v – an embedding type – §4.2

the symmetry condition for a network, an embedding token, and an embedding type – §4.2

$v \cdot u$ – an embedding token defined by an embedding type v and an embedding token u – §4.2

$v \circ f$ – the embedding type induced by a homomorphism f from an embedding type v – §4.2

$u|_{\mathcal{N}'}$ – the restriction of an embedding token u to a subnetwork $\mathcal{N}'$ – §4.2

$v|_{\mathcal{N}'}$ – the restriction of an embedding type v to a subnetwork $\mathcal{N}'$ – §4.2

(a,s) – a symmetry of a network $\mathcal{N}$ with respect to v – §4.3

π – a local symmetry of a network $\mathcal{N}_1$ with respect $\mathcal{N}_0, p, v$ – §4.3

the application of a local symmetry to p, u – §4.3

T – a template – (2004, §4)

I – an image, $I\colon \mathbb{R}^2 \to \mathbb{R}$ – (2004, §4)

$sat(x)$ – (definite) saturation at a point x (used in $\rho_{I,T,sat}$) – §4.4

w – the weighting function (used in $\rho_{I,T,sat}$) – §4.4

$\rho_{I,T,sat}$ – the correlation function for an image I and a template T – §4.4

k – a constant in the definition of $\rho_{I,T,sat}$ – §4.4

DM – the definite match function – §4.5

θ – the penalty for a bare symbol (a positive real constant) – §4.5

the recognition problem (final statement) – §4.7

§5. Inclusion Functions

i and j – inclusion functions – §5.2

§6. The Indefinite Match Functions

$sat(x)$ – (indefinite) saturation at a point x – §6.2

IM – the indefinite match function – §6.3

B – the bareness function – §6.3

§7. How the Inclusion Functions are Determined

$\boldsymbol{i}$ – the inclusion vector, with typical component $\boldsymbol{i}_x$ (for $x \in X$) – §7.2

X – the index set for the components of $\boldsymbol{i}$ – §7.2

Y – the index set for the constraints on $\boldsymbol{i}$ – §7.2

$\boldsymbol{c}^y$ – one constraint vector (for one $y \in Y$), with typical component $\boldsymbol{c}^y_x$ (for $x \in X$) – §7.2

$\boldsymbol{m}$ – the vector of coefficients in IM, with typical component $\boldsymbol{m}_x$ (for $x \in X$) – §7.2

T_x – the temperature of x (for $x \in X$) – §7.3

E – an expression that is maximised to determine $\boldsymbol{i}$ – §7.3

T_{min} – the minimum allowed temperature – §7.3

§8. The Extension Operations

pruning operation (and trivial pruning operation) – transforming a network into a subnetwork, given a pair of inclusion functions – §8.2

the restriction of (i,j) to a subnetwork, $\mathcal{N}'$ – §8.2

elementary pruning operation – pruning a symbol, a node or an edge – §8.2

unprunable network – i.e., no non-trivial pruning operation is possible on it – §8.3

extension of a triple $(\mathcal{N}_1, p, u)$ – §8.4

minimal and trivial extensions – §8.4

the extension conditions – §8.4

the extension operations – §8.4

$\theta_0, \theta_1, \theta_2, \theta_3, \theta_4$ – constant thresholds used in the extension operations – §8.2

inextendable triple $(\mathcal{N}_1, p, u)$ – §8.5

merging two symbols – §8.6

partitioning a symbol token – §8.7

partitionable and unpartitionable symbols and networks – §8.7

§9. The Whole Recognition Process

line operations – §9.2

the recognition process (full summary) – §9.5

the outcome of recognition – §9.6