Neighborhood Expansion Grammars[*]

John L. Pfaltz
Dept. of Computer Science
University of Virginia
Charlottesville, VA 22903

March 1, 1999

Dept. of Computer Science Technical Report, TR 99-009

Abstract

Phrase structure grammars, in which non-terminal symbols on the left side of a production can be rewritten by the string on the right side, together with their Chomsky hierarch classification, are familiar to computer scientists. But, these gramamars are most effective only to generate, and parse, strings.

In this report, we introduce a new kind of grmmar in which the right side of the production is simply appended to the intermediate structure in such a way that the left side becomes its "neighborhood" in the new structure.

This permits the grammatical definition of many different kinds of "n-dimensional" discrete structures. Several examples are given.

Moreover, these grammars yield a formal theory grounded in antimatroid closure spaces. For example, we show that restricted neighborhood expansion grammars capture the essence of finite state and context free phrase structure grammars.

[*]Research supported in part by DOE grant DE-FG05-95ER25254.

1 Overview

In this paper we consider two distinct, but intertwined, themes. The first is neighborhood expansion grammars which constitute a considerable change from more traditional phrase structure grammars in that intermediate symbols are not rewritten. Instead, completely new pieces are simply added to the discrete system being generated. Such discrete systems we model by graphs, G_k, in this paper.

The second theme is that of closure spaces and their associated lattices [19]. If the class of discrete system being generated is a closure space, we will be able to define "good" grammars as those which homomorphically preserve the induced lattice structure of the system with each step of the generation.

Finally, we will be able to develop a hierarchy of grammars that includes the Chomsky hierarchy, and by which we can qualitatively measure the complexity of discrete systems.

1.1 Neighborhood Expansion Grammars

In a phrase structure grammar, a substring, or more usually a single non-terminal symbol A, is rewritten by some other longer string α. When phrase structure rewriting is applied to more complex, non-linear systems, a subgraph H, or a single node, of G is rewritten as a larger subgraph H' which must then be embedded into $G' = G - H$. Describing the embedding process is not always easy [20].

Neighborhood expansion grammars do not rewrite any portion of the existing intermediate system. Instead, new portions are simply added to the existing system G. Some subgraph H of G is designated to become the *neighborhood* of this new structure. In production form, the right side H' denotes the new structure, often a single point, and the left side denotes the subgraph H of G which will be the neighborhood of H' in G'. This process is more easily visualized by a small example.

A grammar $\mathcal{G}_{chordal}$ to generate chordal graphs can be described by a single production:

$$K_n := p$$

that is, any clique of order k in G can serve as the neighborhood of a new point p. Every point in K_n will be adjacent to p in G'.[1] Figure 1 illustrates a representative generation sequence. Each expanded neighborhood (in this case clique) has been made bold; and the expansion point circled. The dashed edges indicate those which define the clique as the neighborhood of the expansion point p. It is not hard to see that any graph generated in this fashion must be chordal. Because extreme points are simplicial (neighborhood is a

[1]If we rewrite the production so that the right side is both the neighborhood and the newly created point p within it, the production becomes

$$K_n := K_{n+1}$$

that is, to any clique on n elements can be added a new element to create a clique on $n + 1$ elements.

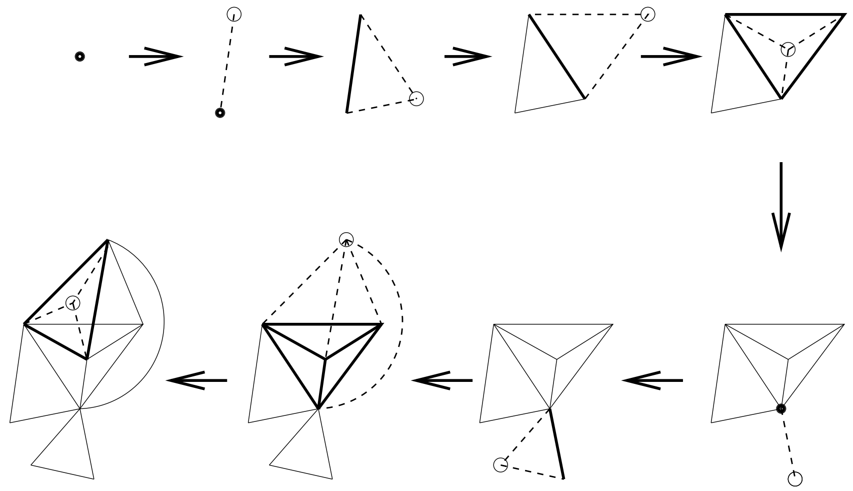


Figure 1: A sequence of neighborhood expansions generating chordal graphs

clique), and because every chordal graph must have at least two extreme points [8, 6], every chordal graph can be so generated.

If the left side of the expansion rule of $\mathcal{G}_{chordal}$ is restricted to be only K_1, *i.e.* a single point, then $\mathcal{G}$ generates all, and only, undirected trees. These two grammars clearly illustrate the essential tree-like structure of chordal graphs. This latter result is well known, since every chordal graph is the intersection graph of the subtrees of a tree [7, 14].

1.2 Neighborhoods

We say $\bar{\eta}$ is a **complete neighborhood** operator if for all $X \subseteq \mathbf{U}$,

$X \subseteq X.\bar{\eta}$,

$X \subseteq Y$ implies $X.\bar{\eta} \subseteq Y.\bar{\eta}$,

$(X \cup Y).\bar{\eta} = X.\bar{\eta} \cup Y.\bar{\eta}$,

$(X \cap Y).\bar{\eta} = X.\bar{\eta} \cap Y.\bar{\eta}$. (possibly unnecessary)

Clearly, a neighborhood operator expands the subset. We are often more interested in the **deleted neighborhood**, η, defined $X.\eta = X.\bar{\eta} - X$, because it represents this incremental addition.

Let us consider a representative neighborhood operator. If G is a graph, we can define a deleted neighborhood by $X.\eta_{adj} = \{y \notin X | \exists x \in X \text{ adjacent to } y\}$. Readily, this is the neighborhood concept we used in $\mathcal{G}_{chordal}$. It appears in many graph-theoretic algorithms.

2 Other Expansion Grammars

We call G a **block graph** if every maximal 2-connected subgraph is a clique K_n [10]. A simple grammar $\mathcal{G}_{block}$ consisting of only the single rule

$$p := K_n \qquad n \geq 1$$

will generate the block graphs. Figure 2 illustrates one such derivation sequence in $\mathcal{G}_{block}$. Here, each p to be expanded is emboldened and its connection to the new K_n indicated

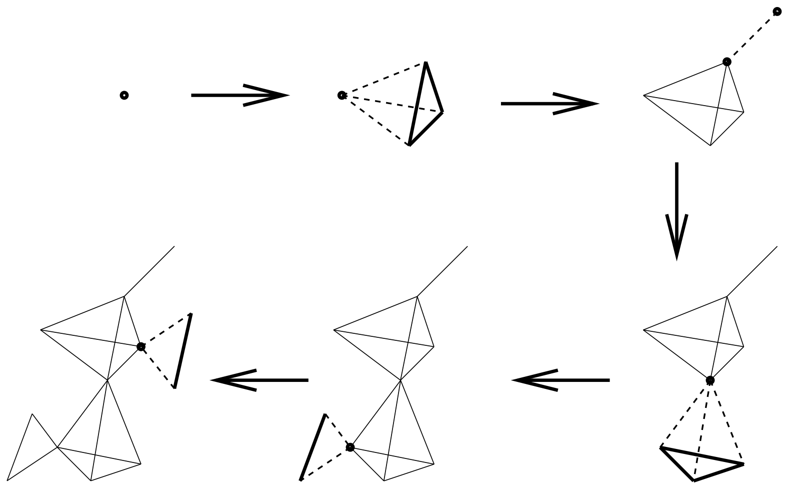
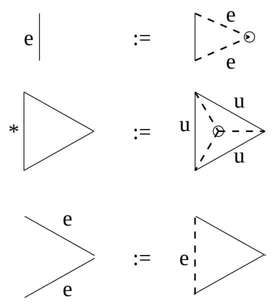


Figure 2: A sequence of neighborhood expansions generating block graphs

by dashed lines. Readily, the neighborhood of each new expansion K_n is the single point. Contrasting $\mathcal{G}_{chordal}$ with $\mathcal{G}_{block}$ is inevitable. The clique K_n on the left side of the production in $\mathcal{G}_{chordal}$ is just exchanged with the new point p on the right side. Both generate tree-like discrete systems.

A grammar to generate triangular planar networks is more complex and introduces the use of labelled edges. The use of edge labels to control grammars is not new, *c.f.* [3, 13].



4

The label e indicates that the edge is exterior. When the first production is applied a new point is created in the exterior in such a way that the edge is its neighborhood. Since, the original edge is no longer exterior, its label is removed and the two connecting edges labelled as "exterior". A new point may be inserted into the interior of any triangle, as in the second production. But, insertion of two such points destroys planarity (except in the special case when two, or more, edges are exterior edges.) Consequently, each of the three edges constituting the neighborhood of the new point are labelled with a u (for *used*). These u labels are never removed. The second production can be applied only if two of the edges are unlabeled ($*$ denotes a don't care condition). The third production is a bit anomolous; it modifies a neighborhood instead of adding a point. Only the indicated edge is added.[2]

The start structure for planar triangulations consists of a single triangle, all of whose edges are exterior. Figure 3 illustrates one such derivation sequence in $\mathcal{G}_{triangle}$.

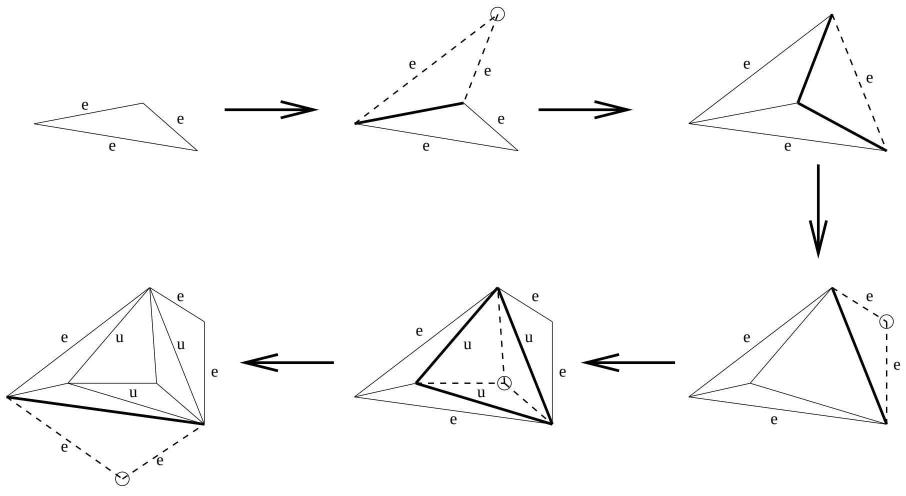


Figure 3: A sequence of neighborhood expansions generating planar triangular grids

If we consider discrete partially ordered systems, there are two distinct neighborhood concepts:

$X.\eta_< = \{z \notin X | z < x \text{ and } z < y \leq x \text{ implies } y \in X\}$ and

$X.\eta_> = \{z \notin X | x < z \text{ and } x \leq y < z \text{ implies } y \in X\}$.

The neighborhood operator $\eta_<$ is called the *covering* operator when the partial order is regarded as a lattice.

[2]This production was added to handle the special cases associated with rule 2. It is not difficult to write two special neighborhood expansion rules to do this; but the rule, as given, is so useful when expanding a triangulation on the "outside", and opposed to subdividing it on the "inside", that it was retained in its current form.

Consequently, we have at least three different grammars to generate partial orders, depending on whether one uses $\eta_<$, $\eta_>$, or $\eta_{<>}$ (which combines them both) as the neighborhood concept. For example, the grammar $\mathcal{G}_{right\_tree}$ with the single production

$$p := q$$

clearly generates the class of rooted trees whose orientation depends on whether p is to be $q.\eta_<$ or $q.\eta_>$. If we let the neighborhood denoted by the left hand of this production be an arbitrary subgraph N of G, as in the production

$$N := q$$

where $q.\eta_<$ is still the neighborhood operator, we get $\mathcal{G}_{left\_root}$ generating all *left rooted* acyclic graphs. Figure 4 illustrates one such generation sequence. If instead of the singleton

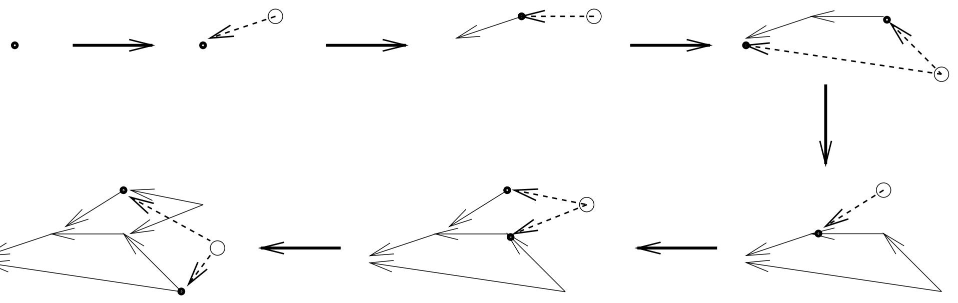


Figure 4: A sequence of neighborhood expansions generating left rooted acyclic graphs

point of Figure 4, we allow an arbitrary antichain, A_n, to be the start structure, then readily $\mathcal{G}_{left\_root}$ becomes $\mathcal{G}_{acyclic}$ which generates all acyclic graphs or partial orders.

If instead, we use the neighborhood operator $\eta_>$, the same production yields all *right rooted* acyclic graphs, which with the antichain start structure generates acyclic graphs as before.

Some interesting subfamilies of the acyclic graphs can be developed if one uses the two-sided neighborhood operator, $X.\eta_{<>} = X.\eta_< \cup X.\eta_>$. The major problem with these two-sided neighborhoods is that for an arbitrary subset of H of G, it may not be evident which elements belong to which side. Such a grammar we would call **ambiguous**. An unambiguous grammar for two terminal, parallel series networks can be specified by requiring the left hand neighborhood to be a single edge. For example, $\mathcal{G}_{ttspn}$ has the two rewrite rules,

$$p_1 \longleftarrow p_2 \quad := \quad q$$

$$p_1 \longleftarrow p_2 \quad := \quad \begin{array}{c} q_1 \\ q_2 \end{array}$$

A representative example of a sequence of structures generating by $\mathcal{G}_{ttspn}$, given a single edge as the start structure is shown in Figure 5. Again, new points of the expansion are denoted

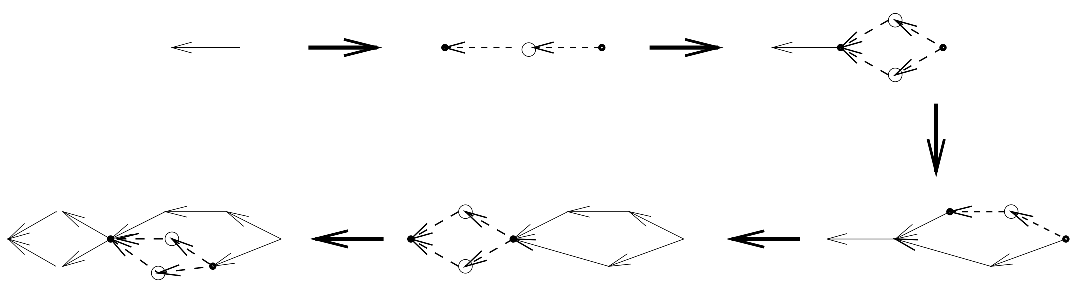


Figure 5: A sequence of two terminal, parallel series networks generated by, $\mathcal{G}_{ttspn}$

by circles with dashed lines indicating the connection to their neighborhoods. Purists may observe that neighborhood expansion grammars are expected to only enlarge the intermediate structures. $\mathcal{G}_{ttspn}$ deletes the edge on the left side after adding the right hand elements for which it is to be the neighborhood. Isn't this really a phrase structure grammar in which edges are rewritten instead of symbols? Although one could adopt this interpretation, it is more accurate to observe that if we retain the left hand edge after the expansion, the resulting structure is simply the transitive closure of a TTSPN, so their erasure is not really changing anything.

3 Closure Spaces

When a phase structure production is applied to a string it preserves the structure of the string. Every intermediate form is a string. We believe that all grammars should preserve the essential structure of their underlying objects. But what is the essential structure of a family of discrete objects?

We believe uniquely generated closure operators represent a basic way of describing the structure of a discrete system. Neighborhood expansions have been defined so as to homomorphically preserve this concept of structure. Any **closure operator** φ must satsify the axioms: $X \subseteq X.\varphi$, $X \subseteq Y$ *implies* $X.\varphi \subseteq Y.\varphi$, and $X.\varphi.\varphi = X.\varphi$, where $X, Y \subseteq \mathbf{U}$ are arbitrary subsets in a universe $\mathbf{U}$ of interest [12]. If in addition, $X.\varphi = Y.\varphi$ implies $(X \cap Y).\varphi = X.\varphi$, we say that φ is **uniquely generated**. For example, monophonic closure [6], in which $X.\varphi$ denotes all points lying on chordless paths between distinct points $p, q \in X$ is uniquely generated over chordal graphs.

A vector space, or matroid $\mathcal{M}$, is the closure (or span σ) of a set of basis vectors [21]. $\mathcal{M}$ must satisfy the **exchange axiom**: $p, q \notin X.\sigma$, *and* $q \in (X \cup p).\sigma$ *imply that*

$p \in (X \cup q).\sigma$.[3] It can be shown [19] that uniquely generated closure spaces must satisfy the **anti-exchange axiom**: *$p, q \notin X.\varphi$, and $q \in (X \cup p).\varphi$ imply that $p \notin (X \cup q).\varphi$.* From this comes the adjective **antimatroid** closure space. Closure spaces form a kind of dual to vector spaces.

If the sets of a closure space $(\mathbf{U}, \varphi)$ are partially ordered by

$$X \ \leq_\varphi \ Y \text{ if and only if } Y \cap X.\varphi \subseteq X \subseteq Y.\varphi$$

we obtain a **closure lattice** [19], in which the sublattice of closed subsets is lower semi-modular.[4] We have asserted that this closure lattice $\mathcal{L}_{(U,\varphi)}$ describes the structure of the underlying discrete system. Figure 6(b) illustrates the closure lattice of the sixth chordal graph generated in Figure 1. This is a complex diagram; but there are significant regularities

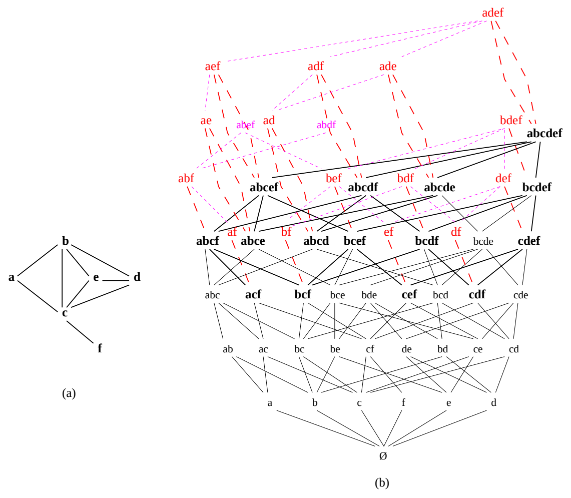


Figure 6: A chordal graph and its closure lattice

that make it amenable to analysis. For example, the sublattice of closed subsets (those which are connected by solid lines) can be shown to be a lower semi-modular lattice [15]. Consider the closed subset $\{abcf\}$ which covers the closed subsets $\{abc\}, \{acf\}, \{bcf\}$. They, in turn, cover their pairwise inf's, $\{ac\}, \{bc\}, \{cf\}$, which in turn cover their common inf, $\{c\}$, to form a distributive sublattice. This is a general property of lower semi-modularity, as shown in [4]. The closed subset $\{abcf\}$ is generated by the subset $\{abf\}$, which is above, to the left, and connected by a dashed line. Such generators, denoted $X.\kappa$, are the unique minimal subsets with a given closure.[5] The subsets contained between any closed set $X.\varphi$

[3]This permits the familiar change of basis of a vector space.

[4]The lattice of vector subspaces partially ordered by inclusion is upper semi-modular.

[5]Many terms are found in the literature for these generating sets depending on ones approach. With convex closure in discrete geometry one speaks of *extreme points* [5]. With respect to transitive closures in

and its generator $X.\kappa$ comprise a boolean algebra, $[X.\varphi, X.\kappa]$, which we sketch in with a dashed diamond. A few of the ordering relationships between non-closed subsets have been indicated by dotted lines.

Although these closure lattices grow exponentially, for example, the final chordal graph generated in Figure 1 on 9 points has 146 closed sets, one can still reason about them. We will see this presently. But, they become too large to illustrate easily.

Although there are similarities, a neighborhood operator is not in general a closure operator because (1) a neighborhood operator need not be idempotent, $X.\bar{\eta} \subseteq X.\bar{\eta}.\bar{\eta}$, and (2) a closure operator need not be union preserving, $X.\varphi \cup Y.\varphi \subseteq (X \cup Y).\varphi$. There can be many different closure operators, φ. Similarly, there can be many neighborhood operators, η. We say that η and φ are **compatible** if X closed implies $X.\bar{\eta}$ is closed.

It is not hard to show that if G is chordal then η_{adj} and $\varphi_{monophonic}$ are compatible on chordal graphs. Similarly, η_{adj} and $\varphi_{geodesic}$ are compatible on block graphs. We know of no uniquely generated, closure operator on triangulated planar grids. But one may exist. In [17] it is shown that at least n^n distinct operators can be defined over any n element system with $n \geq 10$. However, such a closure operator cannot be compatible with the neighborhood operator. Figure 7 indicates why. Assume that every point is closed, so $p.\bar{\eta}_{adj}$ is closed for

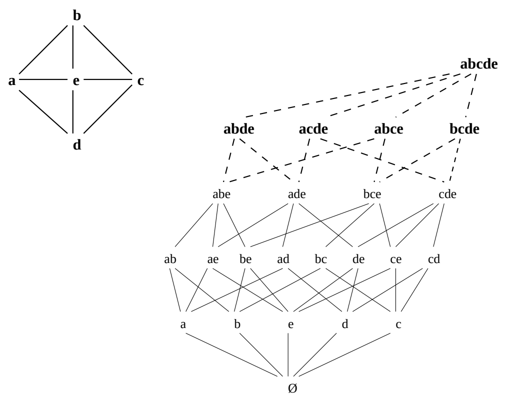


Figure 7: Closed sets of a non-chordal graph

all p. Consequently, the four quadrilaterals $abce, abde, bcde$, and $acde$ are closed. Because closed sets are closed under intersection, the triangles abe, ade, bce, and cde must be closed as well. This determines the upper portion of the lattice [6] which cannot be a closure lattice.

relational algebras one calls them the *irreducible kernel* [1]. In [19], we called them *generators*, but denoted them with the symbol β, suggestive of *basis* the generators of a matroid space.

[6]In Figure 7, all edges are also closed. Because closed sets are closed under intersection, the edges ae, be, ce, and de must be closed. One could assume that the remaining 4 edges are not closed, in which case ab becomes the generator of abe, *etc.* But, this will not change the essence of this counter example.

A fundamental property of uniquely generated closure lattices can be found in [19]

Lemma 3.1 *If φ is uniquely generated, and if $Z \neq \emptyset$ is closed, $p \in Z.\kappa$ if and only if $Z - \{p\}$ is closed, in which case $Z.\kappa - \{p\} \subseteq (Z - \{p\}).\kappa$.*

Using this property one can deduce that $\{abde\}.\kappa = \{bd\}$, $\{acde\}.\kappa = \{ac\}$, $\{abce\}.\kappa = \{ac\}$, and $\{bcde\}.\kappa = \{bd\}$. But, this is impossible. The set $\{ac\}$ cannot generate two different closed sets! Neither can $\{bd\}$.

Because Figure 7 is the simplest non-chordal graph, we have

Theorem 3.2 *Let the individual elements of an undirected structure G be closed. A closure operator φ can be compatible with the adjacency neighborhood operator, η_{adj} if and only if G is chordal.*

Proof: If G is chordal, then the monophonic closure operator $\varphi_{monophonic}$ satisfies the theorem. Only if follows for the example above. □

Conjecture 3.3 *Let $\mathcal{F}$ be a family of discrete undirected systems G. If there exists a neighborhood operator η and a compatible closure operator φ in which every singleton is closed, then G is essentially tree-like.*

For "tree-like", we expect a definition based on intersection graphs [14].

In [19], three different closure operators are defined over partially ordered systems. They are

$$Y.\varphi_L = \{x | \exists y \in Y, x \leq y\},$$
$$Y.\varphi_R = \{z | \exists y \in Y, y \leq z\},$$
$$Y.\varphi_C = \{x | \exists y_1, y_2 \in Y, y_1 \leq x \leq y_2\}.$$

On partially ordered systems, φ_L, and φ_R are ideal closure operators, φ_C is an interval, or convex, closure operator.[7] It is not difficult to show that $\eta_<$ is compatible with both φ_L and φ_R, as is $\eta_>$.

The induced closure lattice resulting from the closure operator φ_L on the 7 point poset of Figure 4 is shown in Figure 8.

As in Figure 6, the sublattice of closed subsets is denoted by solid lines; and the boolean intervals bounded by the closed subsets $X.\varphi$ and their generating sets $X.\kappa$ are denoted by dashed lines which have been oriented toward the upper left. This example is less cluttered, so the characteristic closure lattice structure is more evident. For instance, the subset $\{ef\}$ is the unique generator for the closed set $\{abcdef\}$. All 2^4 subsets in the boolean lattice

[7]Partial orders may be structured with respect to at least these three different closure operators, φ_L, φ_R or φ_C. The ability to interpret discrete systems with respect to different closure concepts can be quite powerful. See [18].

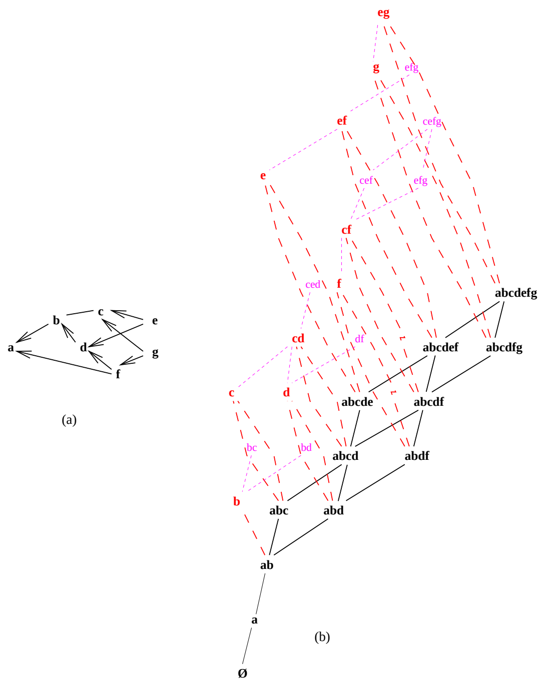


Figure 8: A closure lattice $\mathcal{L}_{G,\varphi_L}$

they delimit (including $\{cef\}$), have $\{abcdef\}$ as their closure. The ordering between these boolean generator sub-lattices, which have been suggested by a few dotted lines, mirror the ordering of the closed sets below them.

4 Structure Preservation

Why should one adopt a process that generates a lfamily of discrete structures by adding some completely new portion within a neighborhood rather than rewriting an existing element, which of course must have a neighborhood? A partial answer was given in sections 1 and 2. The neighborhood expansion approach can be used to generate several families of interesting discrete systems. In Section 5, we will demonstrate that the classes of regular and context free languages in the Chomsky hierarchy of the phrase structure grammars can be subsumed by neighborhood expansion classes. In this section we examine a more subtle argument based on the homomorphic preservation of discrete system structure.

Homomorphic (structure preserving) *generation* is most easily defined in terms of its inverse, homomorphic *parsing*. In this case parsing consists of simply deleting the point and all its incident edges. Figure 9(b) illustrates the deletion χ_d of the element d from the chordal graph of Figure 2(a). We let χ, which is suggestive of "striking out", denote the deletion operator. The subscript denotes the element, or set, being deleted. In the

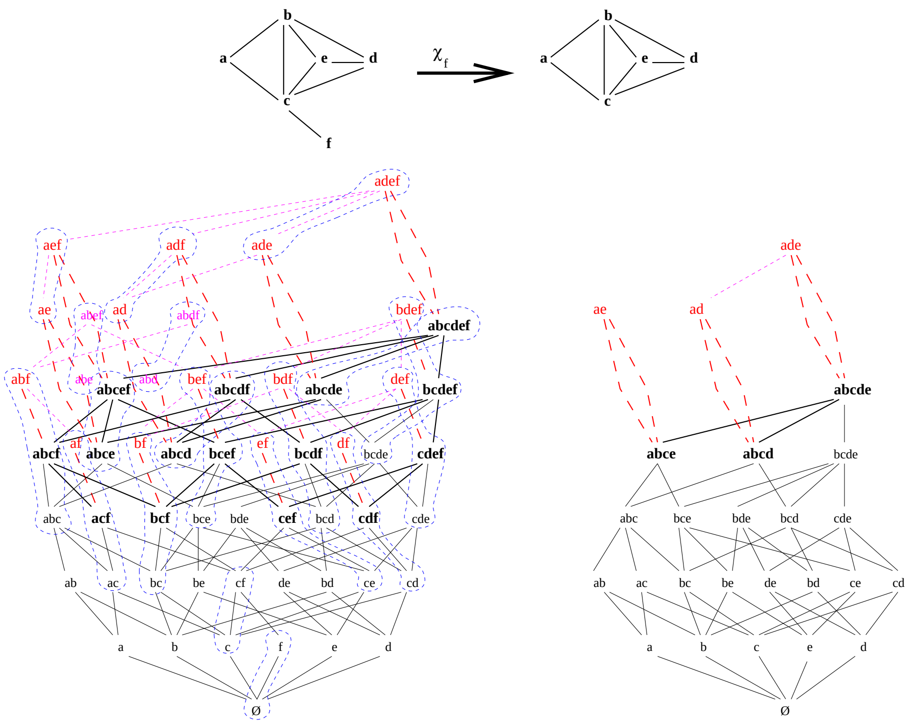


Figure 9: A "deletion" transformation, χ_d

source lattice $\mathcal{L}_G$, pre-image equivalence sets are indicated by dotted lines. The element d is mapped onto $\emptyset$ in the target lattice, and many sets containing d, such as $\{bcd\}$, are mapped onto the set obtained by removing d. But, as we can see from the moderately complex pre-image partition in $\mathcal{L}_G$ of Figure 9, deletion is *not* just a simple set extension of a point function.[8] For example, the set $\{ad\}$ maps onto the set $\{abc\}$. This is necessary if the the transformation is to preserve the structure of *all* the subsets, that is, if the lattice of $G.\chi_d$, denoted $\mathcal{L}_{G.\chi_d}$, is to be a semi-homomorphic contraction of $\mathcal{L}_G$. For a discrete system with n elements, its closure lattice will have 2^n elements. Consequently, reducing the size of a system greatly reduces its stuctural complexity, as is clear in Figure 9.

[8]Deletion of elements make little sense within the usual definition of discrete functions because a deleted element has no image in the codomain. However, in the context of a lattice over all subsets of a discrete space, a deleted element or subspace can be simply mapped onto the empty set, $\emptyset$, in the codomain.

We now have a more formal way of defining what a homomorphic neighborhood expansion, ϵ_N, really is. Let $G^{(n)} \xrightarrow{\chi_p} G^{(n-1)}$ be any deletion in a discrete system with n points. By an expansion ϵ_N with respect to the neighborhood N we mean an inverse operator to χ_p, where N is the image in $G^{(n-1)}$ of the neighborhood N of p denoted $p.N$ in $G^{(n)}$. Thus, we always have $G^{(n)} \xrightarrow{\chi_p} G^{(n-1)} \xrightarrow{\epsilon_N} G^{(n)}$ And, since deletions always induce homomorphic mappings between the corresponding closure lattices, neighborhood expansion can be regarded as homomorphic as well.

Lemma 4.1 *If there exists a closure operator, φ that is compatible with the neighborhood operator, η, used by a neighborhood expansion grammar $\mathcal{G}$, then any sequence of deletions or expansions homomorphically preserves the internal structure of each intermediate system.*

The sense of this lemma is illustrated by Figure 10. As we have seen with $\mathcal{G}_{triangle}$, not all

$$\begin{tikzcd}
G^{(1)} \arrow[r, shift left, "\varepsilon_{N_1}"] \arrow[d, "\varphi"] & G^{(2)} \arrow[l, shift left, "\chi_{p_1}"] \arrow[r, shift left, "\varepsilon_{N_2}"] \arrow[d, "\varphi"] & G^{(3)} \arrow[l, shift left, "\chi_{p_2}"] \arrow[r, shift left, "\varepsilon_{N_3}"] \arrow[d, "\varphi"] & G^{(4)} \arrow[l, shift left, "\chi_{p_3}"] \arrow[r, shift left, "\varepsilon_{N_4}"] \arrow[d, "\varphi"] & \cdots \arrow[l, shift left, "\chi_{p_4}"] \arrow[d, "\varphi"] \\
L_{G^{(1)}} & L_{G^{(2)}} \arrow[l] & L_{G^{(3)}} \arrow[l] & L_{G^{(4)}} \arrow[l] & \cdots \arrow[l]
\end{tikzcd}$$

Figure 10: A sequence of deletions/expansions

expansion grammars have compatible closure operators. But, if one exists it enforces a kind of regularity in the grammar's derivations.

5 Expansion Grammars and the Chomsky Hierarchy

How different are neighborhood expansion grammars from more customary phrase structure grammars? We will claim they are very similar, except that neighborhood expansion seems more flexible.

Theorem 5.1 *Let $\mathcal{L}$ be any regular language. There exists a neighborhood expansion grammar $\mathcal{G}$ in which the left side (neighborhood) of every production is a closed singleton element, such that $L(\mathcal{G}) = \mathcal{L}$.*

Proof: Let $\mathcal{G}_{regular}$ be any regular, or finite state, can be cast into either a left, or right recursive form. We assume $\mathcal{G}_{regular}$ is right recursive, that is all productions are of the form $V := aV'|a$, so only the rightmost element of the string is a variable.
We use $\eta_<$ and φ_R as our neighborhood and closure operators respectively. The rightmost element

is closed, as are all rightmost intervals. This rightmost element provides the neighborhood, $\eta_<$, for the expansion. If $\eta_< D$ is labeled with V, it is either relabeled with a and the new expansion point labeled with V', or else $\eta_< D$ is just relabeled with the terminal symbol a, thereby ending the generation. If $\mathcal{G}_{regular}$ is left recursive, we would use $\eta_>$ and φ_L as the neisghborhood and closure operators to obtain the same result. $\Box$

The only distinction is that expansion grammars with these properties can also generate languages of rooted trees as well.

Theorem 5.2 *Let $\mathcal{L}$ be any context free language. There exists a neighborhood expansion grammar $\mathcal{G}$ in which the left side of every production is closed, such that $L(\mathcal{G}) = \mathcal{L}$.*

Proof: Because a context free grammar G_{cf} permits non-terminal symbols to be rewritten anywhere in the string, we require **convex closure** φ_C where $X.\varphi_C = \{q | p \leq q \leq r, \quad p, r \in X\}$, to emulate it. We also assume Chomsky normal form [2, 9], in which every production is of the form $V := V_1 V_2$ or $V := a$. Again, it is apparent that the neighborhood labeled with V can be relabeled with V_1 and the expansion element labeled V_2. $\Box$

By their very nature, we would assume that neighborhood expansion grammars can emulate any context-sensitive grammar, which must be non-decreasing. Neighborhood expansion grammars too must be non-decreasing, and the neighborhood dependent productions would seem to exactly capture the context sensitive quality. We believe this to be true. But, as yet, we have found no convincing proof.

It is possible to catalog neighborhood expansion grammars according to whether the left side (neighborhood) of every production is:

a simple closed set,
a closed set,
a singleton element, or
has some other well defined property,

which we call **neighborhood criteria**; and whether the right side (neighborhood) of every production is:

a singleton element, or
has some other well defined property,

which we call **replacement criteria**.

We observe that the left side of the production of $\mathcal{G}_{chordal}$ is a closed neighborhood in the intermediate structure, as in $\mathcal{G}_{block}$ as well. But, in the latter grammar every left side is also a singleton element. The language $\mathcal{L}_{block}$ is a subset of $\mathcal{L}_{chordal}$ [11]. If this production is further restricted to have only singleton elements on the right side, as in $\mathcal{G}_{tree}$, it language is $\mathcal{L}_{tree}$ which is a subset of $\mathcal{L}_{block}$.

The grammar $\mathcal{G}_{left\_root}$ with a characteristic generation as illustrated in Figure 4, satisfies none of the *neighborhood criteria* described above. If in addition to the neighborhood operator $\bar{\eta}_<$, we choose φ_L (as illustrated in Figure 8) to be its compatible closure operator, and we require each left side neighborhood to be closed, we obtain derivations such as Figure 11. Readily, requiring the left side neighborhood to be closed generates a language

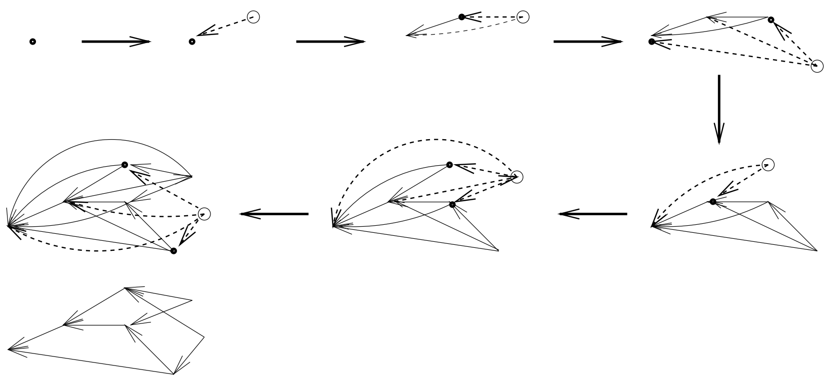


Figure 11: Another sequence of neighborhood expansions generating left rooted acyclic graphs

of transitively closed acyclic graphs.

Simply switching the compatible closure operator to be φ_R includes the set of all **basic** representations of the partial order, that is if $x < y$ and $y < z$ then the edge (x, z) is not in G [16]. If further, we require the left side neighborhood to be a closed antichain, $\mathcal{G}_{left\_root}$ generates only these basic representations of a partial order.

In any case, partially ordered systems are more complex than directed, or undirected, trees with respect to the neighborhood expansion hierarchy. This accords with our algorithmic intuition. Moreover, the set of all trees is comparable to a regular language while the set of all partial orders is comparable to a context free language. Again, this seems intuitively correct. Thus we begin to see a linguistic hierarchy emerging by which we can qualitatively compare the complexity of directed and undirected discrete systems.

6 Neighborhood Operators

So far we have only considered two kinds of neighborhood operator — simple node adjacency, η_{adj}, and the comparative neighborhoods, $\eta_<, \eta_>, \eta_{<>}$ in partial orders. These are familiar, easy to work with, and make good illustrative examples. However, there exist many more ways of defining neighborhood operators and their mechanisms for binding new elements to an expanding configuration. The examples we develop is this section are suggestive of the way atoms may bond to form larger molecules, the way small procedures with some well-defined interfaces may be incorporated into a larger task, or how individual parts may be combined to create larger assemblies.

Our example is absurdly simple. The elements are equilateral triangles with edges labeled (in clockwise order) a, b, c or d, e, f as show in Figure 12. We have superimposed

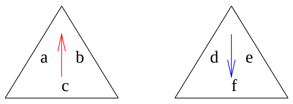


Figure 12: Two spatial elements

an arrow to make the orientation evident in our figures; but they are not actually part of the grammar or language. We suppose that these elements are subject to the affine transformations, translation and rotation, in the plane.

Now suppose that neighborhood operator η_1 specifies that

a can abut d,
b can abut e, and
c can abut f.

We let the grammar $\mathcal{G}_1$ consist of the single rule

$$p := q$$

Now, depending on the initial orientation of the initial element a derivation must yield a uniform pattern such as Figure 13.

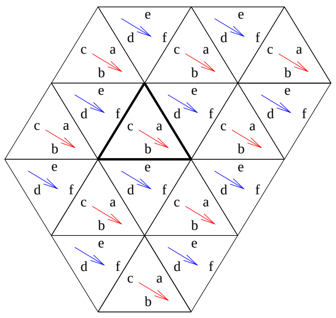


Figure 13: A uniform field in the language $\mathcal{L}(\mathcal{G}_1)$

The rigidity of the neighborhood operator allows no flexibility in $\mathcal{G}_1$. Suppose we adjoin the possibility that

b can also abut d

to the operator η_1 to obtain η_2 and a grammar $\mathcal{G}_2$. Readily $\mathcal{L}(\mathcal{G}_1) \subseteq \mathcal{L}(\mathcal{G}_2)$, but $\mathcal{L}(\mathcal{G}_2)$ has many additional configurations.

If, using the new neighborhood rule, a central hexagonal "rotator" as shown in the center of Figure 14 is initially constructed, and if this is expanded in a "breadth-first" fashion, then a rigidly rotating pattern must be generated. This follows because the only edge labels that

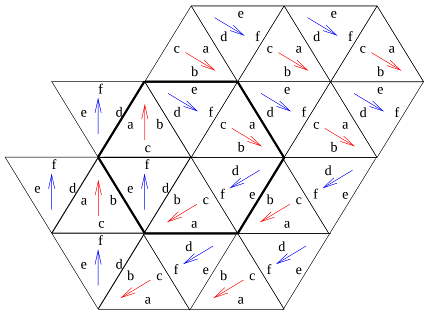


Figure 14: A "rotating" field in the language $\mathcal{L}(\mathcal{G}_2)$ obtained by expanding the central hexagonal configuration

can appear on any convex exterior boundary are a, c, e, and f which uniquely determine the corresponding orientation of the expansion elements as they expand out from the central core.

Of course, there exist more irregular configurations in the language $\mathcal{L}(\mathcal{G}_2)$ which can be generated by alternately letting e and d edges be neighbors of a b edge. Figure 15 illustrates one such configuration. Note that neither of the triangles denoted by x and y

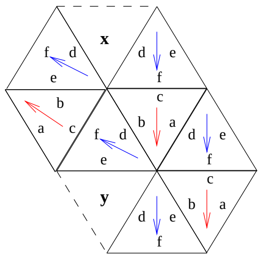


Figure 15: A irregular derivation in the language $\mathcal{L}(\mathcal{G}_2)$

can be occupied in a final configuration.

It would appear that $\mathcal{L}(\mathcal{G}_2)$ can be arbitrarily irregular; but this is not true. We observe that, of the 6 possible orientations of the two elements, only 2 appear in Figure 15, and only 1 more could appear. Similarly, we observe that only 3 orientations appear in Figure 14; and they are not those of Figure 15. The neighborhood expansion grammar concept has captured a notion of symmetric parity in this very simple hexagonal space. We are exploring what other properties it can describe.

References

[1] Chris Brink, Wolfram Kahl, and Gunther Schmidt. *Relational Methods in Computer Science.* Springer Verlag, Wien, 1997.

[2] Noam Chomsky. Formal Properties of Grammars. In R.D. Luce, R.R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology.* Wiley, New York, 1963.

[3] M.J.M. de Boer and A. Lindenmayer. Map 0L-systems with edge label control, Comparison of marker and cyclic system. In Hartmut Ehrig, Manfred Nagl, Grzegorz Rozenberg, and A. Rosenfeld, editors, *Graph-Grammars and Their Application to Computer Science*, Lecture Notes in Computer Science 291, pages 378–392. Springer-Verlag, Dec. 1986.

[4] Paul H. Edelman. Meet-distributive lattices and the anti-exchange closure. *Algebra Universalis*, 10(3):290–299, 1980.

[5] Paul H. Edelman and Robert E. Jamison. The Theory of Convex Geometries. *Geometriae Dedicata*, 19(3):247–270, Dec. 1985.

[6] Martin Farber and Robert E. Jamison. Convexity in Graphs and Hypergraphs. *SIAM J. Algebra and Discrete Methods*, 7(3):433–444, July 1986.

[7] Fanica Gavril. The Intersection Graphs of Subtrees in Trees are exactly the Chordal Graphs. *J. Combinatorial Theory B*, 16:47–56, 1974.

[8] Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[9] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, MA, 1979.

[10] Robert E. Jamison-Waldner. Convexity and Block Graphs. *Congressus Numerantium*, 33:129–142, Dec. 1981.

[11] Robert E. Jamison-Waldner. Partition Numbers for Trees and Ordered Sets. *Pacific J. of Math.*, 96(1):115–140, Sept. 1981.

[12] Kazimierz Kuratowski. *Introduction to Set Theory and Topology.* Pergamon Press, 1972.

[13] J. Luck and H.B. Luck. From 0L and 1L map systems to indeterminate and determinate growth in plant morphogenesis. In Hartmut Ehrig, Manfred Nagl, Grzegorz Rozenberg, and A. Rosenfeld, editors, *Graph-Grammars and Their Application to Computer Science*, Lecture Notes in Computer Science 291, pages 392–410. Springer-Verlag, Dec. 1986.

[14] Terry A. McKee and Fred R. McMorris. *Topics in Intersection Graph Theory.* SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Math., Philadelphia, PA, 1999.

[15] B. Monjardet. A use for frequently rediscovering a concept. *Order*, 1:415–416, 1985.

[16] John L. Pfaltz. *Computer Data Structures.* McGraw-Hill, Feb. 1977.

[17] John L. Pfaltz. Evaluating the binary partition function when $N = 2^n$. *Congress Numerantium*, 109:3–12, 1995.

[18] John L. Pfaltz. Partition coefficients of acyclic graphs. In Manfred Nagl, editor, *Proc. WG 95, Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Comp. Sci., #1017, pages 318–332. Springer-Verlag, 1995.

[19] John L. Pfaltz. Closure Lattices. *Discrete Mathematics*, 154:217–236, 1996.

[20] John L. Pfaltz and Azriel Rosenfeld. Web Grammars. In *Proc. Intn'l Joint Conf on AI*, pages 609–619, Washington, DC, May 1969.

[21] W. T. Tutte. *Introduction to the Theory of Matroids.* Amer. Elsevier, 1971.