Department of Philosophy - Utrecht University

The Theory of Syntactic Domains

θ π

Marcus Kracht

Logic Group
Preprint Series
No. 75
March 1992

Department of Philosophy
University of Utrecht
Heidelberglaan 8
3584 CS Utrecht
The Netherlands

The Theory of Syntactic Domains

Marcus Kracht*

(Faculteit der Wijsbegeerte
Rijksuniversiteit Utrecht
Heidelberglaan 10
NL-3584 CS Utrecht)

II. Mathematisches Institut
Freie Universität Berlin
Arnimallee 3
D-1000 Berlin 33

February 11, 1992

Abstract

In this essay we develop a mathematical theory of syntactic domains with special attention to the theory of government and binding. Starting from an intrinsic characterization of command relations as defined in [Ba 90] we determine the structure of the distributive lattice of command relations. This allows to introduce implication and negation as constructors, whose logic turns out to be the intuitionistic logic of linear posets. Using what is known about intuitionistic logic we can study how domains can be defined from some basic set of command relations that are naturally supplied by the grammar. Moreover, this can be reversed to see how the requirement that domains can be defined in a particular way constrains the syntax. This general theory will then be applied to GB and we will show that there is great evidence to support our claim that command relations are the basic relations from which all other syntactic domains must be defined in a clear and rigid way.

*This research was supported by the project NF 102/62 - 356 ('Structural and Semantic Parallels in Natural Language and Programming Language') funded by the Netherlands Organization for the advancement of Research (N.W.O.).

Contents

1 Introduction

In a sense, what the entire GB theory is calculated to do is (among other things) precisely to identify the syntactic indicators of scope.
J. HINTIKKA and G. SANDU

A lot of linguistic knowledge consists in knowledge that is phrased not in terms of rules but in terms of *nearness* conditions in trees. To use some simplified examples, a reflexive pronoun must find an antecedent within its own clause, a trace must find an antecedent within the least constituent containing the trace and a barrier (or sometimes two barriers) above the trace NP. All these requirements say in some way or another that if X and Y are some syntactic categories, any construction $\ldots X \ldots Y \ldots$ is banned where X is not close enough to Y. Nearness in grammar is always measured in terms of the syntactic tree not in terms of how many items are in between or how long the intervening material is, although that might occasionally have some influence. This applies not only to syntax but also to other disciplines such as the investigation of discourse; it has been discovered that the availability of a discourse referent is largely scheduled by the internal structure of the discourse and not so much by accidental factors such as time. Nearness conditions are stated in terms of *domains*, that is, 'influence spheres' for elements in a structure.

In contrast to many other devices in syntax, the formal theory of domains is heavily underdeveloped, that is to say, nonexistent. The only exceptions we know of are [Rn 81] and [Ba 90] where the case is argued for some fundamental properties that all domains in grammar must have. The present investigation is intended to attack the theory of domains from a mathematical point of view. Nevertheless we want to make some strong *linguistic* claims namely that the notion of domain is universal in the following sense. Anything that deals with nearness should make reference to domains *as we define them.* This means that not only government or binding should be subjected to regimentation via domains, but also movement and subjacency. This is not as bold a step as it appears to be; formalizing these notions in our way, if possible at all, is rather straightforward. But the hierarchy of domain formation that we will construct will make it apparent which notions are better behaved than others and perhaps also which of the notions of, say, subjacency should be preferred. In addition, we will show that for some domains there is absolutely no way to define them unless one makes particular assumptions about the feature system of the grammar.

This essay starts in § 2 with an intrinsic characterization of command relations (CRs) and gives a rough classfication of them. The relationship between a CR and domains or nearness is that a CR R defines for each node a in the tree a domain R_a. It is useful to single out two special classes of CRs. The smallest class is that of TIGHT CRs. They correspond to the prototypical definitions of C-command, A command relation specifies a set $\{R_a | a \in T\}$ of domains. We refer to a as the **(domain-, R-)head** of R_a. ([Ba 90] call this element the *commander*.) Also, if $\langle a, b\rangle \in R$ (i. e. $b \in R_a$) then a is the **(domain-, R-)head** of b and b the **(domain-, R-)foot** of a. The definition of a domain head goes back to [Rn 81] who was careful not speak of 'the' head of a domain rather than 'a' head. Indeed, if a and b are sisters in a tree then the IDC-domain of a is identical to the IDC-domain of b. Thus, b is a head of IDC_a and a is a head of IDC_b. We say then that a and b are **co-heads** for this relation; formally, a and b are **(R-)co-heads** if $R_a = R_b$, in symbols $a \sim_h b$. $\sim_h$ is an equivalence relation and each maximal set of co-heads is called a **chamber**. Chambers are sets of type $\{b | b \sim_h a\}$ for some a.

On the other side of the fence are the *feet*. Where we called elements co-heads when they generate the same domain, we speak of a and b as **co-feet** if they are contained in the same domains, that is, if for all c $a \in R_c \Leftrightarrow b \in R_c$. In symbols $a \sim_f b$. This again is an equivalence relation; any maximal set of co-feet is a set of type $\{b | b \sim_f a\}$ for some a and is called a **cell**. In contrast to chambers, the cells are rather straightforwardly computed from a given R. The first characterization is that of an atom in the boolean algebra generated by the domains R_a, $a \in T$. That sounds rather horrifying but is in fact not so hard. Just consider the domains R_a as subsets of T; they form a collection of sets closed under intersection but not under union and complementation. But now take as $\mathfrak{B}(R)$ the set of all subsets of T that can be generated from the domains of R by intersection, union and complementation. Then, as is well-known, an atom of $\mathfrak{B}(T)$ is a set of type $A(C) = \bigcap\langle R_a | a \in C\rangle \cap \bigcap\langle -R_b | b \notin C\rangle$ for a C such that $A(C) \neq \emptyset$. Now if $x \in A(C)$ then $x \in R_a$ iff $a \in C$; consequently, any two elements of $A(C)$ are co-feet. On the other hand, given an element x, we let $C = \{a | x \in R_a\}$ and we get $x \in A(C)$ for this C. Moreover, $x \sim_f y$ exactly if $y \in A(C)$ and this shows that cells are indeed such atoms.

Remember also that R-domains are lower cones; so if $R_a = \downarrow x$ and $R_b = \downarrow y$ then $R_a = R_b$ iff $x = y$. This proves first of all that if we take any two different generators x, y of domains they cannot be in the same cell; by which we have at least as many cells as we have different generators of R-domains. But a cell is basically an intersection of cones with a number of cones cut off. As any intersection of cones is a cone, we have that cells are what we call *ragged cones*. A **ragged cone** is any set of type $\downarrow x - \bigcup\langle \downarrow y_i | i \in n\rangle$; ragged cones are *convex sets*. (Recall that C is a convex set if $(\forall xyz)(x, y \in C \wedge x \leq z \leq y. \rightarrow z \in C)$.) It is thus legitimate to speak of the **cell** of R_a whereby we mean the cell of co-feet for the generator of R_a. Moreover, if a and b are co-heads then the cells of R_a and R_b are the same since $R_a = R_b$. Thus there is a one-to-one correspondence between domains, generators of domains, cells and chambers. There is then a way to view R as a function from its chambers to its cells. Given the chambers and cells and the function relating them R is recoverable. But knowing the cells, for example, is not enough. The following theorem shows why this is so and gives us also another characterization of TIGHTNESS.

Theorem 2.2.1 *Let* $\mathbb{T}$ *be a tree and* $\mathfrak{C}$ *a partition of* T *into ragged cones. Then there is a* TIGHT *command relation* R *such that* $\mathfrak{C}$ *is the system of* R*-cells.*

Proof. Take as X the set of all maximal elements of sets in $\mathfrak{C}$. We know that each set $C \in \mathfrak{C}$ has a unique maximal element which is then by definition collected into X. The R-domains now consist of all cones $\downarrow x, x \in X$. If we define R_a to be least cone $\downarrow x$ such that $x \in {}^{\diamond}a$ and $x \in X$, then this relation as defined is indeed tight. This will be proved below. It is not hard to see that R as defined has indeed all $\downarrow x, x \in X$, as its domains and so $\mathfrak{C}$ as its system of cells. ∎

Finally, we mention the notion of a *mate* as defined in [Ba 90]. They define R-**mates** as pairs $\langle a, b\rangle$ such that both $a \in R_b$ and $b \in R_a$. Equivalently, a is a head of b and b a head of a. They incorrectly conclude that if R is TIGHT then R-mates are co-heads. But with the definition given, a and $f_R(a)$ are always R-mates since $f_R(a) \in R_a$ (by definition of f_R) and $a \in R_{f_R(a)}$ because $R_{f_R(a)} \supseteq \downarrow f_R(a) \ni a$ by WELL-INCLUSION. However, a and $f_R(a)$ can never be co-heads unless $f_R(a) = r$ by DOMAIN $^{\diamond}$. There might still be possible uses for mate relations, but they certainly do not characterize the notion of co-headness. However, again we will show that TIGHT means well-behaved.

Proposition 2.2.2 *Let* R *be* TIGHT. *Consider* a *and* b *such that neither* $a = f_R(b)$ *nor* $b = f_R(a)$. *Then* a *and* b *are* R*-mates if and only if they are co-heads.*

Proof. If a and b are co-heads then $a \in R_a = R_b$ and $b \in R_a = R_b$ and so they are R-mates. (This is thus generally valid.) Conversely, if $a \in R_a$ and $a \neq f_R(b)$ then by TIGHTNESS $f_R(a) \leq f_R(b)$; by symmetry, $f_R(b) \leq f_R(a)$ and that had to be shown. ∎

Moreover, if $a \sim_h b$ then a and b are mates. For then $R_a = R_b$ and so since $a \in R_a$ we also have $a \in R_b$ and since $b \in R_b$ also $b \in R_a$. Similarly, if $a \sim_f b$ then $a \in R_c \Leftrightarrow b \in R_c$ for every c and so since $a \in R_a$ we have $a \in R_b$ and by symmetry $b \in R_a$. Thus also co-feet are mates. Both notions are therefore at least partly related with the mate relation.

2.3 Ways of Generating Command Relations

In [Ba 90], it is not functions but relations that are generating CRs and they do that in the following manner. Suppose that G is a binary relation on $\mathbb{T}$. Then the command relation generated by G is defined by $\ulcorner G \urcorner = \{\langle a, b\rangle | (\forall c > a)(aGc \rightarrow$

$b \leq c)\}$. In somewhat more simple words, the relation $\ulcorner G \urcorner$ is determined by the function g which gives for each a the least $c \in\ ^\circ a$ that stands in the G-relation with a. If we denote by $\text{UB}(a, G) =\ ^\circ a \cap G_a$ the set of **upper bounds of** a then $g(a) = \text{MUB}(a, G) = min\ \text{UB}(a, G)$, that is, g picks the least upper bound for a with respect to G. A simple counting argument shows that there are far more generating relations than there are command relations. On the other hand, given R, take $G = \{\langle a, f_R(a)\rangle | a \in T\}$; this is the **graph** of f_R. Then $\ulcorner G \urcorner = R$ since $\text{UB}(a, G) = \{f_R(a)\}$ whence $\text{MUB}(a, G) = f_R(a)$. Thus every CR can be generated via binary relations the easiest and in fact minimal of which is the graph of its associated function. This makes the remarks given in [Ba 90] totally vacuous that government domains can be defined via generating binary relations. This does not disqualify the use of generating relations as such, indeed the formulation of a barrier in [Ch 86] seems to be best approached in these terms; but that seems to have no implications beyond the fact that it qualifies the relation as a CR.

Thus we are left with deciding when two relations G, H generate the same CR. ([Ba 90] write $G \sim H$ for $\ulcorner G \urcorner = \ulcorner H \urcorner$.) This is not hard to do. All that needs to hold is that given any node a, the least upper bound with respect to G should be the same as the least upper bound with respect to H. In particular, from this we deduce that there also is a generating relation for a CR which is maximal with respect to inclusion; given a command relation R we take the set $G = \{\langle a, b\rangle |$ not $a < b$ or not $b < f_R(a)\}$.

[Ba 90] consider also a special type of command relations, namely command relations which are generated by a set $P \subseteq T$ of nodes rather than a binary relation. Define the set $\text{UB}(a, P)$ of **upper bounds** for a to be $^\circ a \cap P$. Then $\text{MUB}(a, P) = \min \text{UB}(a, P)$ is called the **minimal upper bound** of a with respect to P. We say that P **generates** the command relation $\ulcorner P \urcorner = \{\langle a, b\rangle \mid b \leq \text{MUB}(a, P)\} = \bigcup \langle \{a\} \times \downarrow \text{MUB}(a, P) \mid a \in T\rangle$ and that a **P-commands** b if and only if $\langle a, b\rangle \in \ulcorner P \urcorner$. ([Ba 90] write C_P for this relation.) Relations which are generated by a set will from now on be called FAIR, but

Theorem 2.3.1 *A command relation is* FAIR *iff it is* TIGHT.

Proof. Recall that TIGHTNESS can be stated as $a < f_R(b) \to f_R(a) \leq f_R(b)$. Now let R be FAIR. Then $R = \ulcorner P \urcorner$ for some $P \subseteq T$. If $a < f_R(b)$ then the least P-node in the crown of b is strictly above a. Thus the least P-node in the crown of a is below or equal to the least P-node in the crown of b and that was to be shown for TIGHTNESS. Conversely, suppose that R is TIGHT. First, we define $P = \{f_R(a) | a \in T\}$. It is left to show that $R_a = \downarrow \text{MUB}(a, P)$ for all a. Or, to use the functions, $f_R(a) = \text{MUB}(a, P)$; that means by definition of P that we have to show that there is no b such that $a < f_R(b) < f_R(a)$. But suppose that $a < f_R(b)$; then by TIGHTNESS $f_R(a) \leq f_R(b)$ and hence $f_R(b) < f_R(a)$ cannot hold. ∎

In [Ba 90], two definitions of FAIRNESS are given which turn out to be different at closer look. The first, stated on p. 7, qualifies R as FAIR if "an upper bound for a node a is an upper bound for every node that a dominates." This suggests that R is taken to be generated by a relation G and that R is FAIR in this sense if $(\forall ab)(b \leq a. \rightarrow .\text{UB}(a, G) \rightarrow \text{UB}(b, G))$ from which $(\forall ab)(b \leq a. \rightarrow .\text{MUB}(b, G) \rightarrow \text{MUB}(a, G))$ and thus $(\forall ab)(b \leq a. \rightarrow .f_R(b) \leq f_R(a))$; thus R is MONOTONE. The reverse need not hold: even if R is MONOTONE G need not satisfy this condition, but there certainly is a H that generates R and has this additional property, namely $H = \{\langle a, b\rangle | b \geq f_R(a)\}$. The second definition, given on p. 29, is more accurate and can be proved to do the job it is intended to do, namely to characterize TIGHTNESS. There, R is classified as FAIR if

$$\begin{aligned} & (\forall abcd)[(aRb \wedge bRc \wedge \neg aRc) \rightarrow (aRd \rightarrow b \geq d)] \\ \Leftrightarrow\ & (\forall abcd)[(aRb \wedge bRc \wedge \neg aRc \wedge aRd \wedge \neg b \geq d) \rightarrow \bot] \\ \Leftrightarrow\ & (\forall abcd)[(aRb \wedge aRd \wedge \neg b \geq d) \rightarrow (bRc \rightarrow aRc)] \\ \Leftrightarrow\ & (\forall ab)[(\exists d)(aRb \wedge aRd \wedge \neg b \geq d) \rightarrow (\forall c)(bRc \rightarrow aRc)] \\ \Leftrightarrow\ & (\forall ab)(b \in R_a \wedge \downarrow b \neq R_a. \rightarrow .R_b \subseteq R_a). \end{aligned}$$

For the reasons given we have chosen to revise the terminology; it was also not obvious why the name FAIR was chosen. We think TIGHT is more suggestive.

2.4 Small Print

It is easily seen that relations satisfying CONSTITUENCY correspond to functions on trees. Further requirements on these functions can be expressed by a definition of the type DOMAIN$^\sigma$ where σ is a function assigning a subset of possible generators for R_a for each a. For quasi command relations $\sigma = \uparrow$ and for command relations $\sigma = {}^\diamond$. We call σ a **selector function.** We have tried to keep our definitions and results independent of the selector function, but in the actual wording we will use ${}^\diamond$, the selector function for command relations. All results will remain valid on the condition that $\sigma(a)$ is linear for every $a \in T$. Any particular relation R satisfying DOMAIN$^\sigma$ is represented by another function f_R satisfying $\downarrow f_R(a) = R_a$. Such a function will satisfy $f_R(a) \in \sigma(a)$ and every such function will give rise to a relation satifying DOMAIN$^\sigma$. One may wonder whether it is a good strategy to allow $R_a = \downarrow a$. We think that it this is not the case for the reason that when we will study the logic of domains, such a choice would allow the lexicon to interfere in the definition and behaviour of domains. This is to be avoided since the whole point of a grammar is that it only deals with classes of lexical items i. e. it abstracts away from the special properties of particular lexical items.

3 Representation of Command Relations

3.1 The Lattice of Command Relations

Over a finite set T of cardinality n there exist as many (binary) relations as there are subsets of $T \times T$. Since $T \times T$ has n^2 elements, there are 2^{n^2} relations. It is standard knowledge that the set of (binary) relations over T is closed under unions, intersections and complements and hence forms a boolean algebra which we denote by $\mathfrak{Re}_2(T) = \langle 2^{T\times T}, -, \cap, \cup \rangle$. (Note that we write 2^M for the powerset of M. Sometimes we write $Re_2(M)$ for the set of binary relations over M.) Now if T is the set of nodes of a tree $\mathbb{T} = \langle T, r, < \rangle$ and we want to study the command relations over $\mathbb{T}$ there are immediate questions as to whether the set of command relations over $\mathbb{T}$, denoted by $Cr(\mathbb{T})$, are likewise closed under these operations. It will be shown that closure under negation does not obtain in general, but any union or intersection of command relations is again a command relation. Whence they form a sublattice of $\mathfrak{Re}_2(\mathbb{T})$. Its structure can be fully described.

In order to describe the lattice $\mathfrak{Cr}(\mathbb{T})$ of command relations over $\mathbb{T}$ we use the correspondency between CRs and functions that satisfy DOMAIN $^\diamond$ which we call **strictly increasing functions**. This correspondency can be turned into a homomorphism between the lattice of such functions with operations $\sqcap, \sqcup$ defined below and the lattice of command relations. Let f and g be functions satisfying DOMAIN $^\diamond$. Then define $f \sqcup g$ and $f \sqcap g$ by $f \sqcup g(a) = \max\{f(a), g(a)\}$ and $f \sqcap g(a) = \min\{f(a), g(a)\}$. We then have

$$(\dagger) \qquad \begin{aligned} f_{R\cup S} &= f_R \sqcup f_S \\ f_{R\cap S} &= f_R \sqcap f_S \end{aligned}$$

For a proof suppose $R_a = \downarrow b$ and $S_a = \downarrow c$. Then since $b, c \in {}^\diamond a$ we have either $b < c$, $b = c$ or $b > c$ since ${}^\diamond a$ is linear. Then $(R \cup S)_a = R_a \cup S_a = \downarrow b \cup \downarrow c = \downarrow \max\{b, c\}$, $(R \cap S)_a = R_a \cap S_a = \downarrow b \cap \downarrow c = \downarrow \min\{b, c\}$. The least strictly increasing function is $\bot$ which for every $x < r$ gives $\bot(x) \succ x$ – the unique cover of x. The largest function is $\top : x \mapsto r$. Now denote the set of strictly increasing functions on $\mathbb{T}$ by $Sf(\mathbb{T})$ and the set of command relations on $\mathbb{T}$ by $CR(\mathbb{T})$. The correspondence ($\ddagger$) on page 2.2 yields a one-to-one map conforming with ($\dagger$). This is summarized in the next theorem; before we state it, we must explain the notation. We have $\mathfrak{Cr}(\mathbb{T}) = \langle CR(\mathbb{T}), \cap, \cup, \bot, \top \rangle$ as well as $\mathfrak{Sf}(\mathbb{T}) = \langle Sf(\mathbb{T}), \sqcap, \sqcup, \bot, \top \rangle$. $\mathfrak{Cr}(\mathbb{T})$, being a lattice of sets with intersection and union as operations, is distributive.

Theorem 3.1.1 $\mathfrak{Cr}(\mathbb{T}) \cong \mathfrak{Sf}(\mathbb{T})$. *Both are distributive lattices.* ∎

Now we know that $\mathfrak{Cr}(\mathbb{T})$ is closed under union and intersection we have that it is a **sublattice** of $\mathfrak{Re}_2(\mathbb{T})$; this we write as $\mathfrak{Cr}(\mathbb{T}) \rightarrowtail \mathfrak{Re}_2(\mathbb{T})$, where the arrow

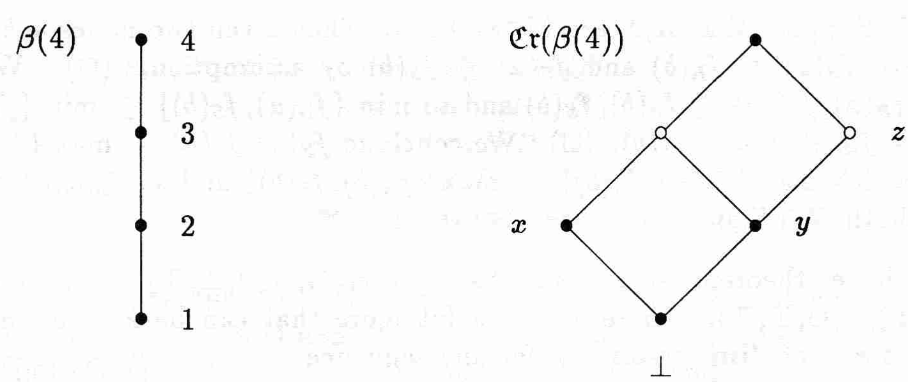
$\rightarrowtail$ tells us that there is an injective lattice homomorphism in the given direction. (Surjective homomorphisms are indicated by $\twoheadrightarrow$.) Moreover, both lattices have a top and a bottom element, but the bottom elements do not coincide. Thus we write $\mathfrak{Cr}(\mathbb{T}) \overset{1}{\rightarrowtail} \mathfrak{Re}(\mathbb{T})$ where the 1 indicates that top is mapped onto top. If bottom were also mapped onto bottom we would write a $\overset{0,1}{\rightarrowtail}$. So much for notation.

The structure of the lattice $\langle Sf(\mathbb{T}), \sqcup, \sqcap, \top, \bot\rangle$ is easily exhibited. Since $f \in Sf(\mathbb{T})$ is determined by the value of the $a \in T$ – which is an element of the crown $^\circ a$ and can be chosen independently of the other values –, $\sharp\, Sf(\mathbb{T}) = \prod_{a\in T} \sharp\, ^\circ a$. Moreover, $R \subseteq S$ if and only if $(\forall a)(R_a \subseteq S_a)$ if and only if $(\forall a)(f_R(a) \leq f_S(a))$. Let $\ell(k) = \langle\{1, 2, \ldots, k\}, \max, \min, \bot, \top\rangle$ be the unique linear lattice of k elements. For each $a \in T$ define a map $\iota_a : \mathfrak{Sf}(\mathbb{T}) \to \ell(\sharp\, ^\circ a)$ by $\iota_a : f \mapsto card(\, ^\circ a \cap \downarrow f(a))$. This map is a lattice homomorphism as can be checked. By principles of universal algebra, two homomorphisms $h_1 : \mathfrak{L} \to \mathfrak{N}_1, h_2 : \mathfrak{L} \to \mathfrak{N}_2$ induce a homomorphism $h_1 \otimes h_2 : \mathfrak{L} \to \mathfrak{N}_1 \otimes \mathfrak{N}_2$ from $\mathfrak{L}$ into the product $\mathfrak{N}_1 \otimes \mathfrak{N}_2$ which consists of pairs $\langle n_1, n_2\rangle$ with $n_i \in N_i$; the operations act independently in each component. One has $h_1 \otimes h_2(\ell) = \langle h_1(\ell), h_2(\ell)\rangle$ for $\ell \in L$. With $\bigotimes_{i\in I} \mathfrak{N}_i$ denoting the product of all $\mathfrak{N}_i$ let $\iota : \mathfrak{Sf}(\mathbb{T}) \to \bigotimes_{a\in T} \ell(\sharp\, ^\circ a)$ be defined by $\iota : f \mapsto \langle \iota_a(f) \mid a \in T\rangle$. It is not hard to see that ι is an isomorphism; it is injective since a function uniquely determines the values for each argument, it is surjective since a function is completely determined by naming the value for each $a \in T$. Thus we have the following result.

Theorem 3.1.2 $\mathfrak{Sf}(\mathbb{T}) \cong \bigotimes_{a\in T} \ell(\sharp\, ^\circ a)$. ■

Immediately, we get that $\mathfrak{Sf}(\mathbb{T})$ and $\mathfrak{Cr}(\mathbb{T})$ are isomorphic to a boolean lattice if and only if every node is of depth < 3. We examine this construction in detail with an example. Let $\mathbb{T} = \beta(4)$; then $\mathfrak{Cr}(\mathbb{T}) \cong 3 \times 2 \times 1 \times 1 \cong 3 \times 2$ while there are $2^{4\times 4} = 2^{16} = 65536$ binary relations on $\mathbb{T}$.

Fig. 1



In the picture on the right, the circles denote command relations which are not TIGHT whereas TIGHT relations are denoted by a blob. The lattice can be obtained as follows. Every command relation can be represented by a strictly increasing function on $\beta(4)$. Since such a function must satisfy $f(4) = f(3) = 4$, it is completely determined by its values for 1 and 2. Any combination of $1 \mapsto 2, 1 \mapsto 3, 1 \mapsto 4$ on the one hand and $2 \mapsto 3, 2 \mapsto 4$ on the other is possible. The resulting lattice is a product of the lattices of functions $\langle\{\bot, x\}, \sqcup, \sqcap\rangle$ and $\langle\{\bot, y, z\}, \sqcup, \sqcap\rangle$ with $\sqcup$ and $\sqcap$ as defined in (†) where $\bot : 1 \mapsto 2, 2 \mapsto 3$, $x : 1 \mapsto 2, 2 \mapsto 4$, $y : 1 \mapsto 3, 2 \mapsto 3$ and $z : 1 \mapsto 4, 2 \mapsto 3$. The homomorphism ι associates with each function f the quadruple $\langle f(1) - 1, f(2) - 2, f(3) - 3, 1\rangle$. And for $i < 4$, $\iota_i(f) = f(i) - i$ as well as for $i = 4$ $\iota_i(f) = 1$ is the projection onto the i^{th} linear component.

Incidentally, the functions x, y and z generate $\mathfrak{Sf}(\beta(4))$ since they are the $\sqcup$-irreducible elements. (Recall that an element a is called **$\sqcup$-irreducible** if $x \sqcup y = a$ implies $x = a$ or $y = a$.) This claim is easily verified, since the functions satisfy that $f(x)$ is a mother of x for all $x < r$ except for one. Each element of $\mathfrak{Sf}(\ell(4))$ is a join of a finite set of $\{x, y, z\}$. By standard representation theorems for distributive lattices, $\mathfrak{Sf}(\ell(4))$ is isomorphic to the lattice of downward closed subsets of $\{x, y, z\}$. In this particular case, $Y \subseteq \{x, y, z\}$ is downward closed if and only if $z \in Y$ implies $y \in Y$. The elements x and $x \sqcup y$ are not absolute, since $x(2) = 4 > x(1) = 2$ and $x \sqcup y(2) = 4 > x \sqcup y(1) = 3$.

3.2 The Algebraic Structure of Monotone and Tight Relations

We begin with the MONOTONE relations. The first theorem to note is that the MONOTONE relations $Mcr(\mathbb{T})$ are also closed under intersection and union and $\top, \bot$ are both monotone. Again this is most suitably expressed algebraically.

Theorem 3.2.1 $\mathfrak{Mcr}(\mathbb{T}) \overset{0,1}{\rightarrowtail} \mathfrak{Cr}(\mathbb{T})$.

Proof. Suppose that R, S are MONOTONE. Then given two nodes a, b with $a \leq b$ we have $f_R(a) \leq f_R(b)$ and $f_S(a) \leq f_S(b)$ by assumption. $(\sqcap)$: We conclude $\min\{f_R(a), f_S(a)\} \leq f_R(b), f_S(b)$ and so $\min\{f_R(a), f_S(b)\} \leq \min\{f_R(b), f_S(b)\}$ whence $f_{R\cap S}(a) \leq f_{R\cap S}(b)$. $(\sqcup)$: We conclude $f_R(a), f_S(a) \leq \max\{f_R(b), f_S(b)\}$, from which $\max\{f_R(a), f_S(b)\} \leq \max\{f_R(b), f_S(b)\}$ and so $f_{R\cup S}(a) \leq f_{R\cup S}(b)\}$. Thus both $R \cap S$ and $R \cup S$ are MONOTONE. ■

The above theorem says also that $\langle Mcr(\mathbb{T}), \cap, \cup, \bot, \top\rangle$ is a sub-lattice of $\langle CR(\mathbb{T}), \cap, \cup, \bot, \top\rangle$. There is still a bit more that can be known. Say that an element x is of **dimension** n if for any sequence $\bot \prec y_1 \prec y_2 \prec \ldots \prec y_k = x$ we have $k = n$. In distributive lattices this is well-defined. An **atom** is an element of dimension 1.

Proposition 3.2.2 *Every atom of* $\mathfrak{Cr}(\mathbb{T})$ *is* MONOTONE. *Moreover,* $\mathfrak{Mcr}(\mathbb{T}) \cong \mathfrak{Cr}(\mathbb{T})$ *if and only if* $\mathfrak{Cr}(\mathbb{T})$ *has no irreducible elements of dimension 2 if and only if* $\mathfrak{Cr}(\mathbb{T})$ *is a boolean lattice if and only if* $\mathbb{T}$ *has no nodes of depth 3.*

Proof. Suppose that R is an atom in $\mathfrak{Cr}(\mathbb{T})$. Then the associated function is such that there exists an element w such that for all $x \neq w, r$ $f_R(x) \succ x$ but $f_R(w) \succ v \succ w$ for some v. This function is monotone and that proves the first claim. Now if $\mathfrak{Cr}(\mathbb{T})$ has a non-monotone element then there must be a node of depth 3; but if there is, let us call it v, then the function f such that $f(x) \succ x$ for all $x \neq r, v$ and $f(v) = f(r) = r$ is associated to a non-monotonic relation. ■

Comparing this with the discussion of the last section we see that any $\sqcup$-irreducible of $\mathfrak{Cr}(\mathbb{T})$ of dimension > 2 is not monotone. Now we are turning to TIGHT relations. The first question is to what extent P characterizes $\ulcorner P \urcorner$.

Theorem 3.2.3 $\ulcorner P \urcorner = \ulcorner Q \urcorner$ *if and only if* $P \cap int(\mathbb{T}) = Q \cap int(\mathbb{T})$. *Thus there are* $2^{\sharp\, int(\mathbb{T})}$ FAIR *command relations.* FAIR *command relations are closed under intersection but not under union, i. e. the* Fair *CRs form a meet semilattice.*

Proof. Suppose that $P \cap int(\mathbb{T}) \neq Q \cap int(\mathbb{T})$, say $b \in P \cap int(\mathbb{T})$ but $b \notin Q \cap int(\mathbb{T})$. Then b has a daughter a and $b < r$. Then $\text{MUB}(a, P) = b$ but $\text{MUB}(a, Q) > b$, whence $\ulcorner P \urcorner \neq \ulcorner Q \urcorner$. Now suppose $P \cap int(\mathbb{T}) = Q \cap int(\mathbb{T})$. Then if a P-commands b, $\text{MUB}(a, P) \geq b$. Clearly, $\text{MUB}(a, P) \neq a$ unless $a = r$. So if there is a P-node properly dominating a, there is a Q-node properly dominating a and vice versa and hence $\text{MUB}(a, P) = \text{MUB}(a, Q)$. If there is none, $\text{MUB}(a, P) = r$. But then $\text{MUB}(a, Q) < r$ cannot hold. Hence $\text{MUB}(a, P) = \text{MUB}(a, Q)$ in both cases.

To see that FAIR command relations are closed under intersection we show that a P-commands b and Q-commands b if and only if a $P \cup Q$-commands b. To this end it remains to be shown that $\text{MUB}(a, P \cup Q) = \min\{\text{MUB}(a, P), \text{MUB}(a, Q)\}$. This is not hard to verify.

$$\begin{aligned}
\text{MUB}(a, P \cup Q) &= \min\ \text{UB}(a, P \cup Q) \\
&= \min\ {}^{\diamond}a \cap (P \cup Q) \\
&= \min\ \{\, {}^{\diamond}a \cap P,\ {}^{\diamond}a \cap Q\} \\
&= \min\ \{\text{UB}(a, P), \text{UB}(a, Q)\} \\
&= \min\ \{\text{MUB}(a, P), \text{MUB}(a, Q)\}
\end{aligned}$$

To see a counterexample that FAIR command relations are not closed under union, consider again $\mathbb{T} = \beta(4)$ and $P = \{2\}$, $Q = \{3\}$. In the picture, P corresponds to x and Q to y. $\langle 1, 3\rangle \in \ulcorner P \urcorner \cup \ulcorner Q \urcorner$ since $\langle 1, 3\rangle \in \ulcorner Q \urcorner$ but $\langle 1, 4\rangle \notin \ulcorner P \urcorner \cup \ulcorner Q \urcorner$.

However, $\langle 2,4\rangle \in \ulcorner P\urcorner \cup \ulcorner Q\urcorner$ and so $\ulcorner P\urcorner \cup \ulcorner Q\urcorner$ is shown to lack TIGHTNESS. Hence it is not FAIR. ■

Fig.2

$\mathfrak{Fcr}(\ell(4))$

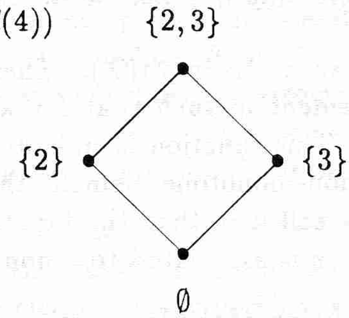


Fig. 2 shows the Hasse diagram of the semilattice of FAIR command relations for $\mathbb{T} = \beta(4)$. In Fig.1 there are four of them as predicted by Theorem 3.2.3. x of Fig.1 corresponds to $\{2\}$ and y to $\{3\}$. $\{2,3\}$ corresponds to $z \sqcup x$. It is easily shown that there are exactly $2^{\sharp T-1}$ FAIR quasi command relations. To sum up, the FAIR command relations form a semilattice and its structure is as follows, where $\mathfrak{Fcr}(\mathbb{T}) = \langle Fcr(\mathbb{T}), \cap, \bot, \top\rangle$.

Theorem 3.2.4 $\mathfrak{Fcr}(\mathbb{T}) \cong \langle 2^{int(\mathbb{T})}, \cup, \emptyset, int(\mathbb{T})\rangle$. ■

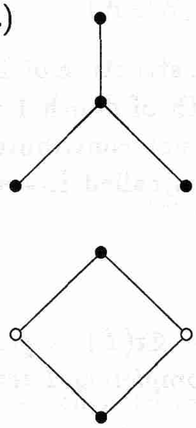
Let us say that a CR is **semi-tight** if it is a union of tight relations. SEMI-TIGHT relations are MONOTONE, but the converse need not hold. The set $STR(\mathbb{T})$ is closed under union and intersection and is therefore a sub-lattice of $\mathfrak{Mcr}(\mathbb{T})$, containing $\top, \bot$.

Theorem 3.2.5 $\mathfrak{Str}(\mathbb{T}) \overset{0,1}{\rightarrowtail} \mathfrak{Mcr}(\mathbb{T})$. ■

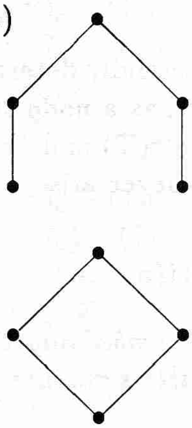
3.3 Recovering the Structure of the Tree

It is quite revealing to see how much information the various lattices contain about $\mathbb{T}$. A first guess – that $\mathfrak{Cr}(\mathbb{T})$ tells us everything about $\mathbb{T}$ – turns out to be false. Here are three simple trees with isomorphic lattice of CRs.

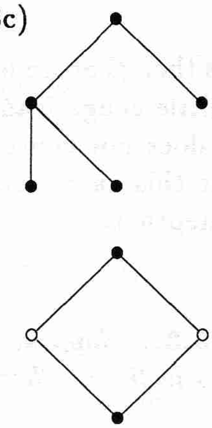
(3a)



(3b)



(3c)



If we analyze the isomorphism type of $\mathfrak{Cr}(\mathbb{T})$ then we see that all it encodes is how many nodes of given depth $n > 1$ the tree contains. For example, if $\mathfrak{Cr}(\mathbb{T}) \cong 2 \times 3 \times 3 \times 4 \times 4$, we have a single node of depth 2 and two nodes of depth 3 and 4 since each linear factor corresponds with a single node. (Note that unlike with numbers, 4 is not the same as 2×2. Thus counting the number of CRs is even less revealing as the isomorphism type of the lattice. In our example we have a lattice with 288 elements and hence, knowing that it is completely decomposable, we are still left with the types $2 \times 2 \times 2 \times 2 \times 2 \times 3 \times 3, 2 \times 2 \times 2 \times 2 \times 3 \times 3 \times 4$ and $2 \times 3 \times 3 \times 4 \times 4$. Note that other decompositions such as $3 \times 3 \times 4 \times 8$ have the same number of elements but there is no tree $\mathbb{T}$ such that $\mathfrak{Cr}(\mathbb{T})$ has this type, as we would then have no nodes of depth 2, but nodes of depth 3.) It is clear then why nodes of depth 1 are not recoverable from this representation; for we have $\mathfrak{L} \cong \mathfrak{L} \otimes 1$ and hence linear factors of type 1 can not be recovered.

Knowing the number of nodes of given depth is obviously not enough; we still need to know how nodes of depth $n+1$ depend on nodes of depth n, that is, we need to know about $\prec$. Here the study of minimal TIGHT relations is useful. A **minimal tight relation** (**MTR**) is a CR that is TIGHT and also minimal as a TIGHT relation; equivalently, it is an atom in the lattice of TIGHT relations. Using our knowledge about TIGHT relations we can see that MTRs are of the form $\ulcorner T - \{x\} \urcorner$ for some interior x. As it turns out, a MTR might not be minimal as a CR as is exemplified in (3a). But this is so precisely because the node x branches as we will see; and so the position of MTRs reveals the branching structure of $\mathbb{T}$. To prove this, let us suppose that we have a MTR $R = \ulcorner T - \{x\} \urcorner$ for some x. Suppose that x covers the nodes $y_1, \dots, y_r$. Then consider the relations Y_i defined by $f_i(y_i) = w$ where $w \succ x$ but $f_i(n) \succ n$ else. The Y_i are then r distinct MONOTONE CRs of dimension 1 and generate a boolean sub-lattice of $\mathfrak{Cr}(\mathbb{T})$; in addition, $R = \bigcup \langle Y_i | i \leq r \rangle$. To see this, note $f_R(y_i) = w$ for all i, hence $R \supseteq Y_i$ for all i. On the other hand, if $z \neq x, y_i, r$ then $f_R(z)$ covers z. This proves equality; as a consequence R is of dimension r in $\mathfrak{Cr}(\mathbb{T})$. Hence the dimension of R tells us about the the number of nodes covered by x.

It seems then that we have successfully determined the structure of $\mathbb{T}$. But there remains a little snag. Namely if $\mathbb{T}$ has a node that is both of depth 1 and interior then (a) it does not contribute to $\mathfrak{Cr}(\mathbb{T})$ and (b) it does not contribute to $\mathfrak{Fcr}(\mathbb{T})$. In grammar this case fortunately never arises. A node is called **invisible** if it is interior of depth 1.

Theorem 3.3.1 *Suppose that the embedding $\mathfrak{Str}(\mathbb{T}) \rightarrowtail \mathfrak{Cr}(\mathbb{T})$ is given together with the two lattices. Then the visible structure of $\mathbb{T}$ is completely determined.* ∎

Strictly speaking we do not need the entire lattice of semi-tight relations, but this way the theorem is better stateable. Observe that this theorem is correct in this formulation because a CR is minimally tight exactly if it is minimally semi-tight; hence, from the embedding of the semi-tight relations into the command relations we can read off the minimal tight relations of $\mathfrak{Cr}(\mathbb{T})$.

3.4 Cores and Mates

A final point to note about command relations is that in many cases there is an additional clause requiring that a commands b only if neither dominates the other. [Ba 90] have argued in the case of Langacker's S-command that the condition that a must precede b takes care of this condition and they intend this argumentation to carry over to C-command and other command relations. There are, however, formal means of dealing with this clause. Say that the **core** R^c of a command relation R is defined by

$$R^c_a = R_a - (\uparrow a \cup \downarrow a)$$

Then a **S-commands** b in the sense of Langacker if and only if $\langle a, b\rangle$ is in the core of $\ulcorner S \urcorner$. A priori there is no reason why the condition excluding domination should be excluded from the definition; even more so, since taking cores is a well-behaved function namely a lattice homomorphism.

Theorem 3.4.1 $(-)^c$ *is a lattice homomorphism.*

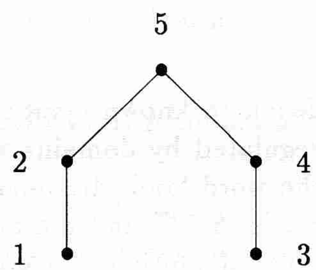
Proof. For a proof let $a \in T$.

$$
\begin{aligned}
(R \cup S)^c_a &= (R \cup S)_a - (\uparrow a \cup \downarrow a) \\
&= (R_a \cup S_a) - (\uparrow a \cup \downarrow a) \\
&= (R_a - (\uparrow a \cup \downarrow a)) \cup (S_a - (\uparrow a \cup \downarrow a)) \\
&= R^c_a \cup S^c_a
\end{aligned}
$$

$$
\begin{aligned}
(R \cap S)^c_a &= (R \cap S)_a - (\uparrow a \cup \downarrow a) \\
&= (R_a \cap S_a) - (\uparrow a \cup \downarrow a) \\
&= (R_a - (\uparrow a \cup \downarrow a)) \cap (S_a - (\uparrow a \cup \downarrow a)) \\
&= R^c_a \cap S^c_a. \blacksquare
\end{aligned}
$$

Finally we turn to mate-relations. Although we have cast some doubts as to whether mate relations are good relations, they nevertheless behaves better from an algebraic point of view than co-headness or co-footness, for which we have not been able to establish any algebraic properties. It is not hard to see that mate relations are generally not command relations. For example, the mate relation corresponding to IDC-command holds in a binary branching tree only between sister nodes. But if x, y are sisters and $z < y$, then x IDC-commands z but z does not IDC-command x. Hence, the two are not mates. Since $(R \cap S)^{-1} = R^{-1} \cap S^{-1}$ we get $M_R \cap M_S = M_{R \cap S}$; so, mate relations are closed under intersections. Moreover, each command relation defines a different mate relation. For a command relation R is characterized uniquely by the pairs $\langle a, c\rangle \in R$ where $a < c$. For such pairs, $\langle a, c\rangle \in M_R$ if and only if $\langle a, c\rangle \in R$ since $\langle c, a\rangle \in R$ is always satisfied. From the mate relation M R can be recovered, for letting $R_a = \bigcup\langle \downarrow c \mid \langle a, c\rangle \in M, c \in {}^\circ a\rangle$ we get $M_R = M$. To use algebraic terminology again, if $\langle M(\mathbb{T}), \cap\rangle$ is the semilattice of mate relations on $\mathbb{T}$, the map $R \mapsto M_R = R \cap R^{-1}$ is an isomorphism from $\langle Cr(\mathbb{T}), \cap\rangle$ to $\langle M(\mathbb{T}), \cap\rangle$. To see that the union of mate relations need not be a mate relation it suffices to show that $R \mapsto M_R$ does not preserve set theoretic unions, since if $M_R \cup M_S = M_T$ for some T, $T = R \cup S$. We will however give a counterexample to both claims. For $\mathbb{T}$ take

Fig.4



and $R = R_f, S = R_g$ where $f : 1 \mapsto 2, 3 \mapsto 5$, $g : 1 \mapsto 5, 3 \mapsto 4$. Then $\langle 1,3\rangle \notin R$, $\langle 3,1\rangle \in R$ whence $\langle 1,3\rangle \notin M_R$. Likewise $\langle 1,3\rangle \in S$, $\langle 3,1\rangle \notin S$ and so $\langle 1,3\rangle \notin M_S$.

Consequently, $\langle 1,3\rangle \notin M_R \cup M_S$. But since $\langle 1,3\rangle, \langle 3,1\rangle \in R \cup S$, we get $\langle 1,3\rangle \in M_{R\cup S}$. This shows $M_R \cup M_S \neq M_{R\cup S}$. In order to prove that $M_R \cup M_S$ is not a mate relation, observe that $\langle 1,5\rangle \in M_S \cup M_R$ and $\langle 3,5\rangle \in M_R \cup M_S$. If $M_R \cup M_S = M_T$ for some T, we had $\langle 1,5\rangle, \langle 3,5\rangle \in T$, whence $\langle 1,3\rangle \in T$ and $\langle 3,1\rangle \in T$ showing $\langle 1,3\rangle \in M_T$, a contradiction.

3.5 Small Print

There is an interesting connection between cores and C-command. Consider a node a. Then there is a command relation R such that $R^c_a \neq \emptyset$ exactly if there is a branching node strictly above a. Moreover, if B denotes the relation C-command then $B_a = \downarrow b$ for the lowest branching node b dominating a. Now in the lattice of cores of quasi command relations, a relation A^c is an atom, exactly if $A^c_a \neq \emptyset$ for one and only one $a < r$. Consequently, B^c is the union of all atoms in the lattice of cores of quasi command relations. Equivalently, it is the least command relation assigning a nonzero core domain to each $a < r$, if there is one at all. (This is the lattice dual of a definition often encountered in general algebra; there one considers the intersection of all maximal elements. This yields the set of *non-generators* of an algebra.)

4 The Logic of Command Relations

4.1 Internalizing Definitions

The study of a *logic* of command relations seems far fetched; yet, there is a natural way to make this meaningful. Consider the case when a CR R is defined in the following way from two CRs S and T.

$$aRb \text{ iff } aSb \text{ and } aTb$$

This definition involves in addition to known symbols also the logical word 'and'. If grammatical processes are regulated by domains we may ask ourselves whether or not it is possible to make the word 'and' disappear. In the present case this is easy. Just let R be the intersection $S \cap T$ and the above is automatically fulfilled. This strategy to replace a complex definition of a domain by a single domain that encodes this definition we call **internalization**. As it stands, we have been able to internalize 'and' by '$\cap$'; the reason for this is that CRs are closed under intersection and thus R – defined in this way – is again a CR. By similar arguments we see that a definition of R via

$$aRb \text{ iff } aSb \text{ or } aTb$$

yields $R = S \cup T$ and thus 'or' is internalized by '$\cup$'. There are, however, definitions that resist internalization. To take a real example, consider the conditions [A] and [B] of the binding theory.

CONDITION [A] An anaphor must be bound inside its governing category.

CONDITION [B] A pronoun must be free in its governing category.

If we assume that we know how to define the set GC of governing categories the two conditions can be accurately rephrased as follows.

CONDITION [A] If x is an anaphor, then x GC-commands its antecedent.

CONDITION [B] If x is a pronoun, then x does not GC-command any of its antecedents.

Condition [B] is exactly the opposite or negation of Condition [A]. Thus the 'domain' in which a pronoun may find an antecedent according to this definition is exactly the complement of the domain for [A], namely GC-command. However, it is easily seen that such a relation can never qualify as a CR in our sense since it violates WELL-INCLUSION by excluding x from its own domain. Maybe this is the reason for the rather ill-fated history of Condition [B]. It exemplifies an instance of a definition of the type

$$aRb \text{ iff not } aSb.$$

Such a definition is not good as it does not define a CR; equivalently, we may classify the notion of boolean negation by saying that 'not' is not internalizable. Similarly, a definition

$$aRb \text{ iff } aSb \text{ implies } aTb$$

fails to yield a CR given that both S and T are CRs. Yet there are definitions that do correspond in some way to negation and implication, which can also be internalized. For example, let R be the largest CR such that $S \cap R \subseteq T$; then this can be checked to be a CR and internalizes some notion of 'implication'.

The following section is meant to give some formal credibility to this. It has a speculative status since we are not intending to view the new types of negations or implications as intrinsically useful for grammar; in fact, that they lack MONOTONY disqualifies them immediately. Nevertheless, we believe that the detour through logic will pay off in providing us with a good epistructure in which formal results of significance can be proved among which Theorem 4.2.1 which will play a central role in decidability proofs.

4.2 The Intuitionistic Nature of Command Relations

We have described the lattice of command relations and have seen that it is distributive. By standard knowledge of lattice theory, a finite distributive lattice is a heyting algebra, that is, for every a, b there exists a largest element c such that (‡) $a \sqcap c \leq b$; and this element is denoted by $a \rightarrow b$. For if we take c to be the union of all elements satisfying (‡), we can prove that c itself satisfies (‡) using distributivity.

$$a \sqcap d \leq b, a \sqcap e \leq b \Rightarrow a \sqcap (d \sqcup e) = a \sqcap d. \sqcup .a \sqcap e \leq b$$

For infinite lattices this fails unless we have full distributivity. In our case, since it is checked that an infinite disjunction or conjunction of command relations is again a command relation we have indeed a heyting algebra of CRs no matter whether the tree is infinite or not. (Note that if the tree is infinite, the set $^{\diamond}a$ is a well-order and that accounts for this fact here.) Let us denote the heyting algebra of CRs over a tree $\mathbb{T}$ by $\mathfrak{H}(\mathbb{T}) = \langle Cr(\mathbb{T}), \sqcap, \sqcup, \rightarrow, \bot \rangle$. Note that $\bot$ is by definition the least element of the lattice and is nothing but IDC-command. It is not difficult to see that if a distributive lattice $\mathfrak{D}$ is a direct product $\mathfrak{C}_1 \otimes \mathfrak{C}_2$ then seen as a heyting algebra it decomposes in the same way. In our context we can then use our structure theorem for the lattice of CRs. Let $\mathfrak{h}(n) = \langle \{1, 2, \ldots, n\}, min, max, imp, 1 \rangle$ denote the linear heyting algebra of n elements then

$$\mathfrak{H}(\mathbb{T}) = \bigotimes_{a \in T} \mathfrak{h}(\sharp \, ^{\diamond}a))$$

The table for imp is easily exhibited using (‡). Since $\mathfrak{h}(n)$ is linear, we have either $a \leq b$ or $b \leq a$. If $a \leq b$ then $imp(a, b) = n$, the top element, which we might denote by $\top$. However, if $a \geq b$ then $imp(a, b) = b$ since b is really the largest element that can intersect with itself and give b.

A lot of things are known about linear heyting algebras, see for example [Ra 79]. For example, the logic of linear heyting algebras, which is denoted by **Liℓ**, is stronger than intuitionistic logic but weaker than classical logic. Such logics are called **intermediate** or sometimes **superintuitionistic**. **Liℓ** can be characterized over the intuitionistic calculus **Li** by the axiom $p \rightarrow q. \vee .q \rightarrow p$. **Li$\ell$** has the **interpolation property** which says that if $P \rightarrow R \in$ **Liℓ** then there exists a Q with

$var(Q) \subseteq var(P) \cap var(R)$ such that $P \to Q \in$ **Liℓ** and $Q \to R \in$ **Liℓ**. (Q is called an **interpolant**.) **Liℓ** is **pretabular**, which means that every logic that is stronger is determined by a finite algebra i.e. one of the $\mathfrak{h}(n)$, while **Liℓ** itself obviously is not. (For example, classical logic is characterized by $\mathfrak{h}(1)$.) But the most interesting fact for our investigation is the fact that the variety of linear heyting algebras is **locally finite**; that is, every finitely generated linear HA must be finite, and moreover, we can give upper bounds for the n-generated linear HAs. What this means here is that from a given finite set of command relations there are only finitely many command relation that can be composed using the connectives $\bot, \wedge, \vee, \to$. In addition, we will be able to see what sort of relations are definable from others and which ones are not.

Recall that the intuitionistic calculus can be modelled in **partially ordered sets, posets** for short, which are sets ordered by a relation $\leq$ satisfying the axioms

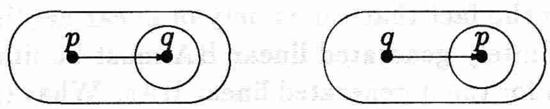
(rf) $x \leq x$
(tr) $x \leq y, y \leq z \Rightarrow x \leq z$
(as) $x \leq y, y \leq x \Rightarrow x = y$.

Given a poset $\mathbb{P} = \langle P, \leq \rangle$ and a set of variables V we define a **valuation** to be a function $\gamma : V \to P$ such that $\gamma(p) = \uparrow \gamma(p)$. This has for consequence for each variable p there is a $x \in P$ such that $\gamma(p) = \uparrow x$. A **model** is a triple $\langle \mathbb{P}, \gamma, x \rangle$ with $x \in P$ being a point. Truth of a formula in a model $\langle \mathbb{P}, \gamma, x \rangle$ is defined as follows:

(v)	$\langle \mathbb{P}, \gamma, x \rangle \models p$	if $x \in \gamma(p)$	
$(\wedge)$	$\langle \mathbb{P}, \gamma, x \rangle \models P \wedge Q$	if	$\langle \mathbb{P}, \gamma, x \rangle \models P$ and $\langle \mathbb{P}, \gamma, x \rangle \models Q$
$(\vee)$	$\langle \mathbb{P}, \gamma, x \rangle \models P \vee Q$	if	$\langle \mathbb{P}, \gamma, x \rangle \models P$ or $\langle \mathbb{P}, \gamma, x \rangle \models Q$
$(\to)$	$\langle \mathbb{P}, \gamma, x \rangle \models P \to Q$	if for all $y \geq x$	$\langle \mathbb{P}, \gamma, y \rangle \models P \Rightarrow \langle \mathbb{P}, \gamma, y \rangle \models Q$
$(\bot)$	$\langle \mathbb{P}, \gamma, x \rangle \not\models \bot$		

There is an alternative formulation. On a finite poset $\mathbb{P}$ we have the set of upward cones $uco(\mathbb{P})$ on which we can base a heyting algebra, since it is a distributive lattice. These naturally defined operations (intersection, union, implication) allow us to lift the map γ into a homomorphism denoted by $\overline{\gamma}$ from the term algebra of propositions into the algebra $\mathfrak{Co}(\mathbb{P})$ of upward cones. Thus, for every formula P the set $\overline{\gamma}(P) = \{x | \langle \mathbb{P}, \gamma, x \rangle \models P\}$ is upward closed. But we can prove an even better result in presence of the fact that our models are not only posets but **linear** posets, which are the same as totally ordered sets. In that case we can prove by induction that for every P there exists a $\ell(P) \in var(P) \cup \{\top, \bot\}$ such that $\overline{\gamma}(P) = \gamma(\ell(P))$. This goes as follows. If P is a variable or a constant, we are done. If $P = Q_1 \wedge Q_2$, we take $\ell(P)$ to be $\ell(Q_1)$ or $\ell(Q_2)$ depending on whether $\gamma(\ell(Q_1))$ or $\gamma(\ell(Q_2))$ is maximal, if $P = Q_1 \vee Q_2$ then we let $\ell(P)$ be such that $\gamma(\ell(P)) = min\{\gamma(\ell(Q_1)), \gamma(\ell(Q_2))\}$. If $P = Q_1 \to Q_2$ then if $\gamma(\ell(Q_1)) \leq \gamma(\ell(Q_2))$ we let $\ell(P) = \top$ and in the other case $\ell(P) = \ell(Q_2)$. The claim that $\overline{\gamma}(P) = \gamma(\ell(P))$ is now verified by induction. We refer to this strategy as *literal reduction.*

Thus, using the connectives has not added too much strength to generate new command relations. But it is of course not true that for every P there exists a c such that $P \leftrightarrow c$ is a theorem.



On the two models pictured above we have $p \wedge q$ once equivalent to q and once equivalent to p. But nevertheless the above result allows us to show that each linear model $\langle \mathbb{P}, \gamma, x\rangle$ is basically equivalent to a model $\langle \mathbb{Q}, \delta, y\rangle$ where $\sharp Q \leq 1 + \sharp var(P)$. For suppose that x immediately precedes y, that is, suppose that $x \leq y$ but for all z with $x \leq z \leq y$ we have $z = x$ or $z = y$. Then if for all $p \in V$, $x \in \gamma(p) \Leftrightarrow y \in \gamma(p)$, then we can drop x from the model because for every formula in the variables from V we will have $x \in \overline{\gamma}(P) \Leftrightarrow y \in \overline{\gamma}(P)$ just because $x \in \overline{\gamma}(P) \Leftrightarrow x \in \gamma(\ell(P)) \Leftrightarrow y \in \gamma(\ell(P)) \Leftrightarrow y \in \overline{\gamma}(P)$. Let then Q be be the set of all y such that there is a p with $y \notin \gamma(p)$ but for all $z > y$, $z \in \gamma(p)$. Such elements we call **critical**. And let $\leq$ be the same order as in $\mathbb{P}$, just reduced to the critical elements. Finally, put $\delta(p) = \gamma(p) \cap Q$. Then if for some x $\langle \mathbb{P}, \gamma, x\rangle \models P$ then there is a critical $y \geq x$ such that $\langle \mathbb{P}, \gamma, y\rangle \models P$ (since $\langle \mathbb{P}, \gamma, x\rangle \models \ell(P)$, and so take y to be the least critical point above x). Then $\langle \mathbb{P}, \delta, y\rangle \models P$. And it is easily checked that $\sharp Q \leq 1 + \sharp var(P)$.

Theorem 4.2.1 *The freely n-generated* Liℓ*-algebra contains at most* $2^{(n+1)^{n+1}}$ *elements.*

Proof. We have seen that each model for a set of n variables is equivalent to one of length at most $n+1$. (For modal logicians: the refined model contains at most chains of length $n+1$.) Each variable gets associated with a point in that chain, namely that point x for which $\gamma(p) = \uparrow x$. We have n variables, so we have at most $(n+1)^n$ assignments at thus as many linear components. Thus the underlying frame consists of $(n+1) \times (n+1)^n = (n+1)^{(n+1)}$ elements. An element of the algebra is an upward set and therefore determines on each chain a unique cone. There are thus at most $2^{(n+1)^{(n+1)}}$ upward sets. ■

5 Command Relations in Labelled Trees

5.1 Labelled Trees

We assume now that we have a finite or countably infinite stock LB of labels. A **labelled tree** is nothing but a pair $\langle \mathbb{T}, \ell\rangle$ where $\mathbb{T}$ is a tree and $\ell : T \to LB$

a function assigning to each node its (unique) label. Given such a labelled tree, the **extension** $[\![L]\!]$ of a label $L \in LB$ is the set of nodes having the label L. We write $\langle \mathbb{T}, \ell, x\rangle \models L$ if indeed $\ell(x) = L$. The set $[\![L]\!]$ naturally defines the relation $\ulcorner[\![L]\!]\urcorner$ which we call L-command. We will also write L for the relation L-command. In this manner, every label can be associated with a TIGHT command relation in $\langle \mathbb{T}, \ell\rangle$. Using union, intersection and the other intuitionistic connectives we can then construct new command relations out of them. To this end, we first define the language $\mathbb{I}(LB)$.

(0) $LB \subseteq \mathbb{I}(LB)$

(+) If $\mathfrak{a}, \mathfrak{b} \in \mathbb{I}(LB)$ then also $\bot, \top, \neg\mathfrak{a}, \mathfrak{a} \wedge \mathfrak{b}, \mathfrak{a} \vee \mathfrak{b}, \mathfrak{a} \to \mathfrak{b} \in \mathbb{I}(LB)$

(μ) Nothing else is in $\mathbb{I}(LB)$.

Any labelled tree $\langle \mathbb{T}, \ell\rangle$ defines a homomorphism $\overline{\ell} : \mathbb{I}(LB) \to \mathfrak{H}(\mathbb{T})$ in the following way.

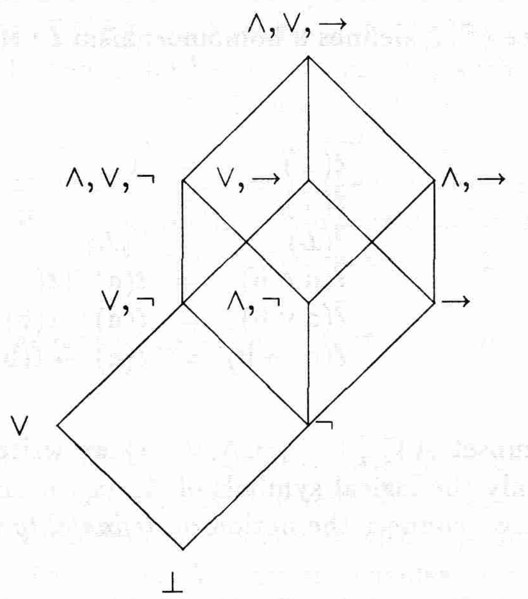
$$\begin{aligned}
\overline{\ell}(\bot) &= \bot \\
\overline{\ell}(\top) &= \top \\
\overline{\ell}(L) &= \ulcorner[\![L]\!]\urcorner \\
\overline{\ell}(\mathfrak{a} \wedge \mathfrak{b}) &= \overline{\ell}(\mathfrak{a}) \sqcap \overline{\ell}(\mathfrak{b}) \\
\overline{\ell}(\mathfrak{a} \vee \mathfrak{b}) &= \overline{\ell}(\mathfrak{a}) \sqcup \overline{\ell}(\mathfrak{b}) \\
\overline{\ell}(\mathfrak{a} \to \mathfrak{b}) &= \overline{\ell}(\mathfrak{a}) \to \overline{\ell}(\mathfrak{b})
\end{aligned}$$

For a selected subset $A \subseteq \{\bot, \top, \neg, \wedge, \vee, \to\}$ we write $\mathbb{I}_A(LB)$ for the sublanguage of CRs where only the logical symbols of A may occur other than the labels. With these languages we connect the notion of *definability* in the following way.

Definition 5.1.1 *Given a class $\mathfrak{T}$ of labelled trees and an abstract relation R defined over every individual member of $\mathfrak{T}$, that is, given an indexed collection $\langle R_{\langle \mathbb{T},\ell\rangle} | \langle \mathbb{T}, \ell\rangle \in \mathfrak{T}\rangle$, we say that R is* **intuitionistically definable** *if there is a $\mathfrak{b} \in \mathbb{I}(LB)$ such that for all $\langle \mathbb{T}, \ell\rangle$ from $\mathfrak{T}$ $\langle \mathbb{T}, \ell\rangle \models R = \mathfrak{b}$. If $\mathfrak{b} \in \mathbb{I}_A(LB)$, R is said to be (intuitionistically) A-definable.*

5.2 Degrees of Definability

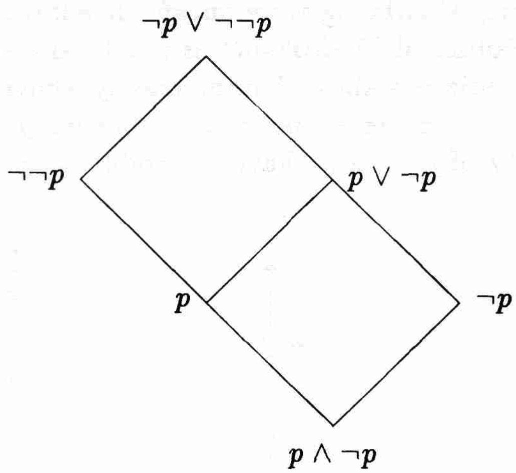
The first question is that of relative strength of the notions of definability. Let us for the moment assume that we have just **Li** and not the stronger **Liℓ**. Then we know that the connectives $\wedge, \vee, \neg, \to$ are independent, that is, any subset of $\{\wedge, \vee, \neg, \to\}$ defines a different class of relations. $\top, \bot$ introduce some complications here since for example $\top = p \rightarrow p$ and $\neg p = p \rightarrow \bot$ so that there is not a complete independence between all six connectives. For reasons that will become clear later, the picture we have to draw here looks quite different from the intuitionistic case. Namely, we will always have that there are $0, 1 \in LB$ such that $\bot = \overline{\ell}(1)$ and $\top = \overline{\ell}(0)$ and moreover for each $L, M \in LB$ there will be a $L \cup M \in LB$ such that $\overline{\ell}(L \cup M) = \overline{\ell}(L) \sqcap \overline{\ell}(M)$. Thus, under these circumstances $\mathbb{I}_\emptyset(LB) = \mathbb{I}_{\top,\bot,\wedge}(LB)$. Thus, as $\mathbb{I}_\bot = \mathbb{I}_\emptyset$, we have $\mathbb{I}_\rightarrow = \mathbb{I}_{\rightarrow,\neg}$. As concerns the other connectives matters are more complex. We expect that the relative notions of definability are also independent given the caveat that $\mathbb{I}_\wedge = \mathbb{I}_\emptyset$ and that $\mathbb{I}_\neg \subseteq \mathbb{I}_\rightarrow$. We have $\mathbb{I}_{\vee,\wedge}(LB) = \mathbb{I}_\vee(\mathbb{I}_\wedge(LB)) = \mathbb{I}_\vee(LB)$ by distributivity; but $\mathbb{I}_{\rightarrow,\wedge} = \mathbb{I}_\rightarrow$ need not hold. In general, $\mathbb{I}_{A,\wedge} = \mathbb{I}_A$ if in every formula in $\mathbb{I}_{A,\wedge}(\mathfrak{Lb})$ conjunction can be pushed inside i.e. if $\mathbb{I}_{A,\wedge} = \mathbb{I}_A \circ \mathbb{I}_\wedge$. This leaves us with the following picture consisting of 10 languages rather than 16 in the case of complete independence.



By earlier results, among the intuitionistically definable CRs the $\{\bot, \top, \wedge\}$-definable relations are TIGHT and the $\{\bot, \top, \wedge, \vee\}$-definable relations are MONOTONE. But does the converse hold as well? The answer is not known to us. Finally, let us look at definable in presence of linearity. There the picture does not change significantly. Although we do now have De Morgan's laws $\neg(p \vee q). \leftrightarrow .\neg p \wedge \neg q, \neg(p \wedge q). \leftrightarrow .\neg p \vee \neg q$ the absence of double negation $\neg\neg p \rightarrow p$ (which we cannot have since it immediately implies Peirce's law and so we have classical logic) means that neither of $\wedge$ and $\vee$ can be reduced to the other with $\neg$. However, it turns out that $\vee$ is definable with the help of $\rightarrow$ and $\wedge$ since $p \vee q. \leftrightarrow .(p \rightarrow q. \rightarrow .q) \wedge (q \rightarrow p. \rightarrow .p)$ and thus $\mathbb{I}_{\rightarrow,\wedge,\neg} = \mathbb{I}$. Thus the total diagram is therefore reduced by one point.

5.3 Formulae with one Variable

The intuitionistic formulae of one variable are well understood. Ordered by implication they form a lattice which is called the **Nishimura-Rieger lattice**. This lattice is infinite in the intuitionistic case but in presence of the linearity axiom this lattice reduces to a six element lattice.



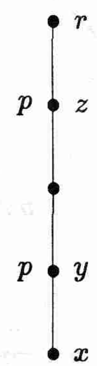
Thus $\top = \neg p \vee \neg\neg p$ and $\neg\neg p \to p = p \vee \neg p$. This can be checked using models, but we omit that here. Instead we want to see what sort of domains can be defined using these formulae. A good description can be obtained from their corresponding functions.

$\mathfrak{d}$	$f_{\mathfrak{d}}$
$\neg p \wedge p$	mother
p	next p strictly above
$\neg p$	r if mother is p; else mother
$\neg\neg p$	mother if mother is p; else r
$\neg p \vee \neg\neg p$	r if mother is p; else next p strictly above
$p \vee \neg p$	r

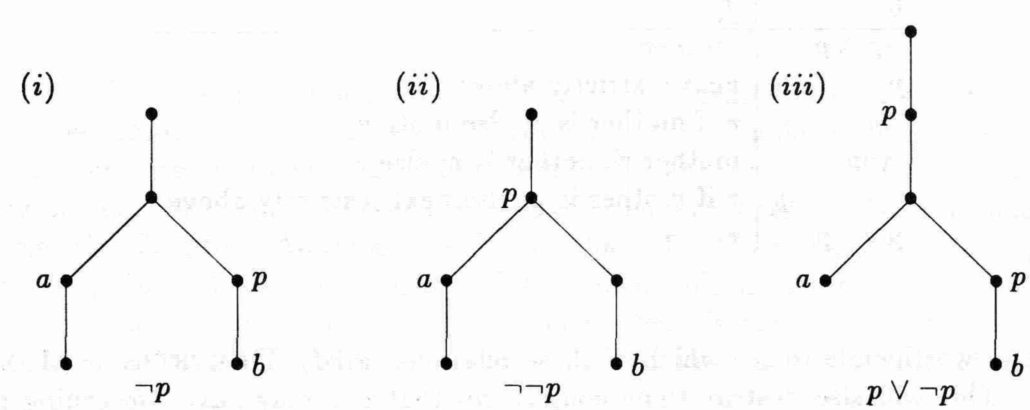
It is also worthwhile to see which of these relations satisfy TIGHTNESS or MONOTONY. This will also destroy some conjectures that one may have concerning the relation between MONOTONY and monotony in logic. This time we omit the trivial relations.

	MON	TIGHT
p	yes	yes
$\neg p$	no	no
$\neg\neg p$	no	no
$p \vee \neg p$	no	no

Although in this case, TIGHTNESS is the same as MONOTONY, this is not true in general. The relation $p \vee q$, being a union of TIGHT relations, is MONOTONE but not itself TIGHT. Failure of MONOTONY is not hard to demonstrate for these relations. Let c be immediately above b immediately above a. Let $c \not\models p, b \models p$. Then $f_{\neg p}(a) = r, f_{\neg p}(b) = c$. However, if $c \models p, b \not\models p$ then $f_{\neg\neg p}(a) = r, f_{\neg\neg p}(b) = c$. For the non-monotonicity of $p \vee \neg p$ we have to produce a more complex model:



Then $f_{p\vee\neg p}(x) = r$ but $f_{p\vee\neg p}(y) = z$. Non TIGHTNESS is harder to demonstrate. To show that we must glue some different branches together:



In each of these cases we have $b \in R_a, b \neq f_R(a)$ but $R_b \not\subseteq R_a$.

5.4 Formulae with Two or More Variables

The number of formulae over a finite set of n variables which are inequivalent over linear models grows quite fast as n increases. But to get a complete overview of the possibilities it is enough to look at those formulae which are conjunctively irreducible, that is, formulae P which cannot be written as $Q_1 \wedge Q_2$ over the same variables such that neither Q_1 nor Q_2 is equivalent to P. The number of irreducible formulae can be estimated by $f(0) = 2$, $f(n+1) \leq 2f(n) + 1$, a rather harmless upper bound. Moreover, it is irrelevant how the variables are named so that $p \vee \neg q$ is essentially the same as $q \vee \neg q$ modulo renaming variables. To see how this reduces the problem we look at the formulae over two variables. There are according to calculations by L. Hendriks, 342 different formulae. But there are only $f(2) = 11(!)$ different irreducible formulae.

(1) $\neg(p \wedge q)$
(2) $\neg q \vee (p \to q)$
(3) $\neg p \vee (q \to p)$
(4) $(q \to p) \vee (q \vee \neg p)$
(5) $(p \to q) \vee (p \vee \neg q)$
(6) $\neg p \to \neg q$
(7) $\neg q \to \neg p$
(8) $\neg\neg(p \vee q)$
(9) $\neg q \to (p \vee \neg p)$
(10) $\neg p \to (q \vee \neg q)$
(11) $((p \vee q) \to (p \wedge q)) \to (p \vee \neg p)$

Among the listed formulae, (2) and (3), (4) and (5), (6) and (7) as well as (9) and (10) are pairs where one results from the other by swapping the variables.

6 Grammars with Attributes versus Grammars with Labels

6.1 Boolean Grammars

We now want to enter a realistic account of domains in natural language. To this end we need to consider first some definitions and facts about context free grammars. Recall that a quadruple $\mathbb{G} = \langle START, NT, TE, R\rangle$ is called a **context free grammar (CFG)** if $START \in NT$, NT and TE are finite and and disjoint sets and R is a finite subset of $NT \times (NT \cup TE)^+$. $START$ is called the **start symbol** and NT the set of **nonterminal symbols**; TE is the set of **terminal symbols** and finally R the set of **rules** of $\mathbb{G}$. A rule is normally written as $A \to \Gamma$

where $A \in NT$ and Γ is a non empty sequence of terminals and nonterminals. The importance of grammars (and context free grammars in particular) is that although $\mathbb{G}$ is finite it may generate an infinite number of syntactic trees. Formally, we first define a **labelled tree** to be a structure $\langle \mathbb{T}, \ell \rangle$ where $\ell : T \to NT \cup TE$ is a function from the nodes of T into the set of symbols of $\mathbb{G}$. We say that $\langle \mathbb{T}, \ell \rangle$ is **generated by** $\mathbb{G}$ or that it **satisfies** $\mathbb{G}$ if the following holds. (We write $t^{\downarrow}$ for the string of nodes immediately dominated by t. Note that this requires that they are linearly ordered. If this is not so, any ordering may be chosen to satisfy the clause (nt) below.)

(r) $\ell(r) = START$
(nt) $\ell(t) \to \ell(t^{\downarrow}) \in R$ for all $t \in T$ which are not leaves
(t) $\ell(t) \in TE$ for all leaves.

(nt) implies that if t is not a leaf then $\ell(t) \in NT$; otherwise no rule can apply to $\ell(t)$. If $\langle \mathbb{T}, \ell \rangle$ satisfies $\mathbb{G}$ we write $\mathbb{G} \gg \langle \mathbb{T}, \ell \rangle$. For what is to follow we will simplify the notion of a context free grammar a bit. The terminal labels are, as far as grammar is concerned, only decoration. The way context free grammars are normally written for natural language the lexical items are classified into certain syntactic categories such that each terminal symbol can be substituted for another symbol of the same category salva acceptabilitate. Each category is represented by a prelexical symbol X and there is a rule $X \to x$ for each lexical item x of category X. There are then no rules expanding X to anything other than a single occurrence of a lexical item of its category. This effectively allows to ignore the particular choice of lexical items for X. We thus replace all terminal symbols uniformly by $STOP$. In addition, we want to treat $START$ and $STOP$ on an equal basis. Both are considered as symbols not belonging to the 'real grammar'. We want that $START$ occurs strictly to the left of a rule just as $STOP$ occurs strictly to the right. This can always be made true in an ordinary CFG by introducing a symbol $START^*$ which satisfies $START^* \notin NT$ and by adding to R a rule $START^* \to \Gamma$ for every rule $START \to \Gamma$ already in R. As there are no real nonterminal symbols we want to speak of the former nonterminals as **labels**. Therefore, in future a CFG is taken to be a quadruple $\langle START, STOP, LB, R \rangle$ where $START, STOP \notin LB$ and R is a finite subset of $(\{START\} \cup LB) \times (\{STOP\} \cup LB)^+$. Thus, labels are assigned only to interior nodes; START is reserved for r and STOP for the leaves.

Modern syntactic theories do, however, not take the syntactic labels of context free grammars as unanalyzable. Indeed, in tandem with the inflation of rules due to the unravelling of conditions on derivations or well-formedness (once believed to be non-context free) into context free rules, the attention has focussed more on the internal structure of category labels than on the formulation of the rules themselves. In GPSG and in its extreme form in HPSG a rather rich theory of these labels better known as *attribute value structures* has been developed and employed. In order not to introduce too much structure we have chosen to admit into our discussion only the boolean part of attribute value structures, which certainly is the most unproblematic and the least controversial part since any syntactic theory will sooner or later accept boolean constructions albeit implicitly as in most GB work.

Definition 6.1.1 *A* **boolean context free grammar** *is a quadruple* $\mathbb{BG} = \langle START, STOP, \mathfrak{Lb}, R\rangle$ *where* $\mathfrak{Lb} = \langle LB, 0, -, \cap, \cup\rangle$ *is a finite boolean algebra,* $START, STOP \notin LB$ *and* R *a finite subset of* $(\{START\} \cup LB) \times (\{STOP\} \cup LB)^+$.

Together with the notion of grammar we also have to adapt our notion of a labelled tree. A **partially labelled tree** is a pair $\langle \mathbb{T}, \ell\rangle$ with $\ell : int(T) \to LB$. If $x \in int(T)$ we write $\langle \mathbb{T}, \ell, x\rangle \models a$ if $\mathfrak{Lb} \models \ell(x) \leq a$. The reason we call this labelling *partial* is that it is no longer true that either $\langle \mathbb{T}, \ell, x\rangle \models a$ or $\langle \mathbb{T}, \ell, x\rangle \models -a$; thus we have lost *bivalence*, a hall-mark of classical logic. Nevertheless, we use classical logic, but the labels now express only part of what is or may be true of a node. A **(fully) labelled tree** is a partially labelled tree in which for every x the label $\ell(x)$ is an atom of $\mathfrak{Lb}$. A partially labelled tree is **generated** by $\mathbb{BG}$ for all t either t is a leaf or there is a rule $A \to \Gamma \in R$ such that $\ell(t) \leq A$ and $\ell(t^{\downarrow}) \leq \Gamma$ (if taken componentwise); or $t = r$, in which case we require $\ell(t^{\downarrow}) \leq \Gamma$ for some rule $START \to \Gamma$. Intuitively, this makes sense as follows. Given a partially labelled tree $\langle \mathbb{T}, \ell\rangle$ and a node $t \in T$ we write $\langle \mathbb{T}, \ell, t\rangle \models B$ to say that in the labelled tree x is a node of **type** B. This is formally the case if $\mathfrak{Lb} \models \ell(t) \leq B$. Now whenever we have a rule $A \to \Gamma$ it is understood to mean that any node of type A immediately dominates a sequence $t^{\downarrow}$ of nodes of type Γ.

The first result to be proved here is that a boolean CFG is effectively a CFG. To this end one should note that a boolean rule $\rho = A \to \Gamma$ effectively abbreviates a set of rules, namely all *precisifications* of ρ. To also be precise, $\sigma = B \to \Delta$ is a **precisification** of ρ, $\sigma \leq \rho$, if $\mathfrak{Lb} \models B \leq A, \Delta \leq \Gamma$, sequences being compared item-by-item in their respective order. (Of course, Δ must be of equal length with Γ.) Each tree in which a rule ρ is used at a point t instantiates a precisification of ρ in the sense that $\ell(t) \to \ell(t^{\downarrow})$ precisifies ρ. Since in a CFG all labels are mutually exclusive, we must take as labels of the grammar all maximal precisifications of $\mathfrak{Lb}$, in other words the **atoms** of $\mathfrak{Lb}$. Each element $b \in LB$ is uniquely determined by the set b^* of all atoms below it. Thus if $\langle START, STOP, \mathfrak{Lb}, R\rangle$ is a boolean CFG, let LB^* be the set of atoms of $\mathfrak{Lb}$. Now replace a rule $\rho = A \to \Gamma$ by the set of atomic precisifications of ρ $\rho^* = \{B \to \Delta | B \leq A, \Delta \leq \Gamma, B \in LB^*, \Delta \subseteq LB^*\}$; and define $R^* = \bigcup\langle \rho^* | \rho \in R\rangle$. Then $\mathbb{G}^* = \langle START, STOP, LB^*, R^*\rangle$ is a CFG. The two grammars are equivalent in the sense that they admit the same trees with labels from LB^*.

6.2 Presentations of Boolean Grammars

Boolean grammars might be well defined and suitable for abstract purposes but in practise one never has the boolean algebra $\mathfrak{Lb}$ as such; rather there exists a description of $\mathfrak{Lb}$ in terms of generators and equations. The generators are primitive properties of nodes such as CAT : N or BAR : 2; things that can be true or false (or perhaps undefined) on nodes and are unanalyzed. But we want to neglect the internal structure of these descriptions in terms of attributes and values and we assume two-valuedness. Even with these simplifications there is a lot to be done. First, assume that we have a stock $F = \{f_i | i \in m\}$ of elementary properties of nodes. Then one can define the **algebra of boolean terms** $\mathfrak{Tm}_{\mathbb{B}}(F) = \langle Tm_{\mathbb{B}}(\mathbb{T}), 0, -, \cap, \cup \rangle$ which consists of all terms that can be written down using the symbols of F and the connectives $0, -, \cap, \cup$. Then define an equivalence $\equiv$ by $P \equiv Q \leftrightharpoons \vdash P \Delta Q$ where $P \Delta Q = P \supset Q. \cap .Q \supset P$, $P \supset Q := -P \cup Q$, and put $[P] = \{Q | P \equiv Q\}$. $[P]$ is the **equivalence class of P** modulo $\equiv$. It is standard from universal algebra that this equivalence relation is also a congruence which in this context means that in propositional logic we can always neglect the difference between equivalent formulae. Thus, we may define boolean operations on the classes instead of formulae via $-[P] = [-P], 0 = [0], [P] \cap [Q] = [P \cap Q]$ and $[P] \cup [Q] = [P \cup Q]$. The algebra of equivalence classes with operations as just defined is written $\mathfrak{Tm}_{\mathbb{B}}(F)/\equiv$. This algebra is the **free boolean algebra over F**. We denote it by $\mathfrak{Fr}_{\mathbb{B}}(F)$. Given that F has m elements, the free algebra has 2^{2^m} elements and is isomorphic to the boolean algebra of subsets of 2^m as can be proved using normal forms. For every formula using symbols of F can be written as a disjunction $\bigcup \langle Q_{C(i)} | i \in n \rangle$ where all $C(i) \subseteq F$ and $Q_D = \bigcap \langle f | f \in D \rangle \cap \bigcap \langle -f | f \notin D \rangle$ for any $D \subseteq F$. (This is called the *disjunctive normal form.*) As there are 2^m choices for subsets $C(i)$, there are 2^{2^m} sets of such sets.

Now as it turns out, the labels F are mostly not independent; in most cases there is some interdependence, for example, a relation $f_0 = f_1 \cup f_2$ or $f_3 \leq f_4$. These relations can always be given the form $P = 1$ for some $P \in \mathfrak{Tm}_{\mathbb{B}}(F)$. For any in-equation $P \leq Q$ is equivalent to $P \supset Q = 1$ and any equation $P = Q$ is equivalent to $P \Delta Q = 1$. We are not actually interested in the completely free algebra on these generators but in a boolean algebra in which some relations and no more hold; such an algebra comes in the definition of a **factor algebra**. Given a boolean algebra $\mathfrak{A}$ and some set $\Delta = \{a_i = 1 | i \in \ell\}$ of relations we can form the algebra $\mathfrak{A}/\Delta$ which makes exactly the a_i equal to 1 plus whatever necessarily follows from that. The conclusions of type $P = 1$ from a set of equations of the same type can be computed as follows. (The symbol $\cong$ denotes as usual the fact that two structures are *isomorphic.*)

($f\wedge$) $a = 1$ and $b = 1$ iff $a \cap b = 1$
($f \leq$) $a = 1$ and $a \leq b$ implies $b = 1$

A set of elements of $\mathfrak{A}$ satisfying $(f\wedge)$ and $(f\ \leq)$ is called a **filter**. It can be shown that if F is a filter then $\mathfrak{A}/F \models a = b$ iff $\mathfrak{A} \models a\Delta b \in F$.

All this motivates the following definition, which is standard in universal algebra.

Definition 6.2.1 *Let X be a finite set and T a finite set of terms over X. The pair $\langle X, T\rangle$ is called a* **finite presentation of** $\mathfrak{A}$, *if* $\mathfrak{A} \cong \mathfrak{Fr}_{\mathbb{B}}(X)/T$.

Finally, we are in a position to define the following notion of a grammar that will be relevant in the subsequent chapters of this essay. It consists simply in replacing the boolean algebra $\mathfrak{Lb}$ by a finite representation of it.

Definition 6.2.2 *A quintuple* $\mathbb{R} = \langle START, STOP, F, EQ, R\rangle$ *is called a* **presented boolean CFG** *if $START, STOP \notin F$, EQ is a finite set of boolean terms over F and R a finite subset of* $(\{START\} \cup Tm_{\mathbb{B}}(F)) \times (\{STOP\} \cup Tm_{\mathbb{B}}(F))^{+}$.

The notions of labelled trees etc. remain intact with the modification that $\mathfrak{Lb}$ is replaced by $\mathfrak{Tm}_{\mathbb{B}}(F)/EQ$. It is not difficult to pass from the presented grammar $\mathbb{R}$ to an equivalent boolean grammar. Namely, with $\mathbb{R}$ as above, define $\mathbb{R}^{\beta} = \langle START, STOP, \mathfrak{Lb}, R^{\beta}\rangle$ as follows. Put $\mathfrak{Lb} = \mathfrak{Fr}_{\mathbb{B}}(F)/EQ$ and let $\kappa : \mathfrak{Fr}_{\mathbb{B}}(F) \twoheadrightarrow \mathfrak{Lb}$ be the canonical homomorphism. (This homomorphism is surjective, whence the double arrow.) Then $R^{\beta} = \{\kappa(\rho) | \rho \in R\}$. Thus a presented grammar may be compiled into a context free grammar but we do not want to think of a presented grammar as coding a somewhat more complex CFG, but rather as a grammar in its own right. But this is not a question pertaining to formal details in any way.

6.3 Small Print

A boolean grammar differs from an ordinary grammar in that it allows for generality or perhaps imprecision. Although the rule schema $\overline{\mathsf{N}} \rightarrow \mathsf{N}\ \overline{\overline{Y}}$ was successfully reduced to a single rule in a boolean grammar, this does not mean that it effectively allows for the use of variables as do the so-called definite clause grammars; these are grammars which are characterized as boolean grammars in which the rules may contain boolean variables. For what boolean grammars cannot do is share information across the symbols occurring in a rule. The schematic rule $\overline{X} \rightarrow X\ \overline{\overline{Y}}$ in $\overline{X}$-syntax, which covers the case of our rule concerning $\overline{\mathsf{N}}$ cannot likewise be reduced to a single rule – although it certainly can if we allow variables. Under the assumption that we have only finitely many nonterminal symbols, an assumption which is true for GPSG for example, such a rule schema anyway represents only finitely many rules, which can of course simply be listed. Thus these devices do not exceed the power of CFGs under this assumption. Thus from our standpoint boolean grammars are sufficient, because we are not interested in questions of expressing generalizations. Boolean grammars are here not for the purpose of expressing generalizations but because they are suitable in this context. This should be born in mind also for with respect to the internal structure of category symbols. We are in no mood to face detailed questions about the exact make-up of categorial symbols or attribute value structures and it does not seem to matter. All that matters for our purposes is whether attribute value structures are *feasible* i. e. whether the logic of the admissible attribute value structures is decidable. This question is answered affirmatively in [Kr 89] for the system of [Ga 88] and by [Ka 90] for the system of Kasper and Rounds as well as [Bl 91] for the system of Kasper and Rounds with negation.

7 Domains in Boolean Grammars

7.1 The Language of Domain Specification

We will now return to the subject of Section 5 where we considered CRs constructed from tight command relations. There the labels were considered unanalyzable. However, this is not an adequate way to formalize the domains that arise in linguistics. We should consider the labels to be members of $Tm_{\mathbb{B}}(F)$, the set of boolean terms over some set F of features. Thus the language of domains $\mathbb{D}(F)$ over F consists of:

- variables $p_i, i \in \omega$
- constants $f_i, i \in n$
- connectives for labels $0, -, \cap, \cup$
- connectives for domains $\bot, \top, \neg, \wedge, \vee, \rightarrow$

With this language we first define the set $\mathbb{L}(F)$ of labels by induction:

(0) $p_i \in \mathbb{L}(F), f_j \in \mathbb{L}(F)$.

(+) If $p, q \in \mathbb{L}(F)$ then $0, -p, p \cap q, p \cup q \in \mathbb{L}(F)$.

(μ) Nothing else is in $\mathbb{L}(F)$.

On top of $\mathbb{L}(F)$ we define $\mathbb{D}(F) = \mathbb{I}(\mathbb{L}(F))$ as in Section 5. Here it pays off to have made a careful distinction between the symbols for labels and the symbols for domains. Note that in the same way that $\mathbb{I}$ can be parametrized with respect to the symbols we can parametrize $\mathbb{L}$ and $\mathbb{D}$ and thus create a number of sublanguages of $\mathbb{D}$ depending on the connectives we allow to occur. In $\mathbb{D}$ we refer to the connectives $0, -, \sqcap, \sqcup$ as the boolean or the *inside* connectives and to the others as the intuitionistic or *outside* connectives. The former characterization refers to their logical behaviour whereas the latter refers to their intrinsic character; 'inside' means operating inside a node whereas 'outside' means operating across nodes.

Given a labelled tree $\langle \mathbb{T}, \ell \rangle$ and a $\mathfrak{d} \in \mathbb{D}(F)$ we define the interpretation $\bar{\ell}(\mathfrak{d})$ by induction in the obvious way. First we let $[\![-]\!] : \mathbb{L} \to 2^T$ be the function that assigns to each $b \in \mathbb{L}(F)$ the set of nodes satisfying b, i.e. $[\![b]\!] = \{x | \mathfrak{L}b \models \ell(x) \leq b\}$.

Next we do induction over $\mathbb{I}$ as defined in Section 5. There is, however, a problem in that partial labellings do not give rise to unambiguously defined CRs. Just consider the notion of S-command in a tree where no node is positively or negatively S. Then in this tree S-command can for example be $\bot$ or $\top$ depending on the chosen precisification. In order to continue at this point we therefore assume to have labellings that are not partial at least for the occurring symbols; what happens in the purely partial case is then up to ones personal choice. We might, for example, let $\mathfrak{d}$ be undefined if it is not unique for all precisifications. Given $x \in T$ we write $\langle \mathbb{T}, \ell, x \rangle \models \mathfrak{d}$ if $\bar{\ell}(\mathfrak{d})_x = \bar{\ell}(\top)_x$ and we write $\langle \mathbb{T}, \ell \rangle \models \mathfrak{d}$ if $\bar{\ell}(\mathfrak{d}) = \bar{\ell}(\top)$. Moreover, we write $\langle \mathbb{T}, \ell, x \rangle \models \mathfrak{d} = \mathfrak{e}$ if $\bar{\ell}(\mathfrak{d})_x = \bar{\ell}(\mathfrak{e})_x$ and again $\langle \mathbb{T}, \ell \rangle \models \mathfrak{d} = \mathfrak{e}$ if $\bar{\ell}(\mathfrak{d}) = \bar{\ell}(\mathfrak{e})$. The **domain theory** of $\langle \mathbb{T}, \ell \rangle$ is $Th^{\mathbb{D}}(\langle \mathbb{T}, \ell \rangle) = \{\mathfrak{d} | \langle \mathbb{T}, \ell \rangle \models \mathfrak{d}\}$. For a set $\mathfrak{T}$ of labelled trees the domain theory of $\mathfrak{T}$ is the common theory of all members of $\mathfrak{T}$ and that is $Th^{\mathbb{D}}(\mathfrak{T}) = \bigcap \langle Th^{\mathbb{D}}(\langle \mathbb{T}, \ell \rangle) | \langle \mathbb{T}, \ell \rangle \in \mathfrak{T} \rangle$. Notice that $\langle \mathbb{T}, \ell \rangle \models \mathfrak{d} = \mathfrak{e}$ iff $\langle \mathbb{T}, \ell \rangle \models \mathfrak{d} \leftrightarrow \mathfrak{e}$ iff $\mathfrak{d} \leftrightarrow \mathfrak{e} \in Th^{\mathbb{D}}(\langle \mathbb{T}, \ell \rangle)$ and thus no additional benefit is gained by introducing the *equational theory* of domains as long as $\leftrightarrow$ is a definable symbol. Finally, given a boolean grammar $\mathbb{G}$ over the same set F of symbols we write $\mathbb{G} \models \mathfrak{d}$ ($\mathbb{G} \models \mathfrak{d} = \mathfrak{e}$) if for all labelled trees $\mathbb{G} \gg \langle \mathbb{T}, \ell \rangle$ we have $\langle \mathbb{T}, \ell \rangle \models \mathfrak{d}$ ($\langle \mathbb{T}, \ell \rangle \models \mathfrak{d} = \mathfrak{e}$). Thus the theory $Th^{\mathbb{D}}(\mathbb{G})$ is simply the common theory of its trees.

7.2 The Formal Context of Grammars, Trees and Domains

This is a digression into a less well-known topic of lattice theory that may shed some more light on the formal properties of the things that we are dealing with. The general sources for the mathematical details are [Wi 90] and [Da 91]. Following [Wi 90], a triple $\langle O, A, R \rangle$ is called a **formal context** if $R \subseteq O \times A$ is a binary relation. Nothing more is required; this seems rather weak but the interest lies in the things that will be defined in due course. The members of O are called **objects** and the members of A **attributes**. R simply specifies which attribute holds of which object. To let matters be concrete, we take $O = \mathbb{BG}(F)$, the set of boolean CFGs over F, $A = \mathbb{D}(F)$ and $R = \models$, the satisfaction relation. Then $\langle \mathbb{BG}(F), \mathbb{D}(F), \models \rangle$ is indeed a formal context. With each object $o \in O$ we can associate the set $o^\wedge = \{a \in A | oRa\}$ of its attributes and with each attribute $a \in A$ we can associate a set $a^\vee = \{o \in O | oRa\}$. Similar definitions can be made for sets; we let $X^\wedge = \{a \in A | (\forall o \in X)(oRa)\}$ for a set $X \subseteq O$ and $Y^\vee = \{o \in O | (\forall a \in Y)(oRa)\}$. The most important notion in the theory of formal contexts is the notion of a *concept*. A **concept** is a pair $C = \langle X, Y \rangle$ of sets $X \subseteq O, Y \subseteq A$ such that $X^\wedge = Y$ and $Y^\vee = X$. $X^\wedge$ is called the **intent** of C and $Y^\vee$ the **content** of C. Thus in C X and Y form an ideal pair of sets of objects and sets of attributes that is married in such a way that Y is the set of common attributes of X and X the set of common objects of Y. In our specific case we ask for sets of boolean CFGs and sets of definable CRs that define each other in this way. From the following list of facts we will show that such concepts are quite easily constructed.

Proposition 7.2.1
(1) $X_1 \subseteq X_2$ *implies* $X_1^\wedge \supseteq X_2^\wedge$ *for* $X_1, X_2 \subseteq O$.
(1') $Y_1 \subseteq Y_2$ *implies* $Y_1^\vee \supseteq Y_2^\vee$ *for* $Y_1, Y_2 \subseteq A$.
(2) $X \subseteq X^{\wedge\vee}$ *and* $X^\wedge = X^{\wedge\vee\wedge}$ *for* $X \subseteq O$.
(2') $Y \subseteq Y^{\vee\wedge}$ *and* $Y^\vee = Y^{\vee\wedge\vee}$ *for* $Y \subseteq A$.
(3) $(\bigcup\langle X_i | i \in I\rangle)^\wedge = \bigcap\langle X_i^\wedge | i \in I\rangle$, $X_i \subseteq O, i \in I$.
(3') $(\bigcup\langle Y_i | i \in I\rangle)^\vee = \bigcap\langle Y_i^\vee | i \in I\rangle$, $Y_i \subseteq A, i \in I$. ■

The reader will find a proof in [Wi 90] if he cannot be bothered to do it by himself. Especially (2) and (2') give us a hint of how to find concepts.

Proposition 7.2.2 *Concepts are of the form* $\langle X^{\wedge\vee}, X^\wedge\rangle$ *and* $\langle Y^\vee, Y^{\vee\wedge}\rangle$ *for some* $X \subseteq O, Y \subseteq A$.

Proof. If $\langle X, Y\rangle$ is a concept then $Y = X^\wedge$ and $X = Y^\vee = X^{\wedge\vee}$. Similarly $X = Y^\vee$ and $Y = X^\wedge = Y^{\vee\wedge}$, as required. ■

A concept $\langle A, B\rangle$ is completely determined by either A or B and if we let $\langle A, B\rangle \leq \langle A', B'\rangle$ iff $A \subseteq A'$ then concepts are ordered and the order is a complete lattice according to (3), (3').

For our particular example there are two things to look at; (a) for a set B of boolean grammars study the domains $\mathfrak{d}$ such that $B \models \mathfrak{d}$; in other words, $B^\wedge$ is nothing but the domain theory $Th^{\mathbb{D}}(B)$ of B. (b) For a set D of domain axioms study the set of grammars $\mathbb{B}$ such that $\mathbb{B} \models D$. For want of a better name we slightly abuse the terminology and call this set the **extent** of a set of domain axioms. We write $Ext(\mathfrak{d})$.

7.3 The Domain Theory of Grammars

There are a number of easy results that we can collect about $Th^{\mathbb{D}}$.

Proposition 7.3.1 *$Th^{\mathbb{D}}(\mathbb{B})$ is closed under modus ponens and substitution. Moreover, every* **Lil** *theorem is in $Th^{\mathbb{D}}(\mathbb{B})$.*

Proof. Suppose $\mathbb{B} \models P$ and $\mathbb{B} \models P \to Q$. Then if $\mathbb{B} \gg \langle \mathbb{T}, \ell \rangle$ we have $\overline{\ell}(P) = \overline{\ell}(\top)$ and $\overline{\ell}(P \to Q) = \overline{\ell}(\top)$ by which $\overline{\ell}(Q) = \overline{\ell}(\top)$ follows since for every x $f_P(x) = r, f_{P\to Q}(x) = r$ so that by definition of $\to$ $f_Q(x) = r$. Hence $\mathbb{B} \models Q$, which had to be proved. For closure under substitution observe that features are represented by constants and so cannot be substituted; only variables can. Now if σ is a substitution replacing each variable p_i by the formula $P_i = \sigma(p_i)$ then assume $\langle \mathbb{T}, \ell, \gamma \rangle \models \mathfrak{d}$ for a particular assignment $\gamma : Var \to 2^T$ of the variables. Now the map $\sigma \circ \gamma : Var \to 2^T$ is also a valuation and $\langle \mathbb{T}, \ell, \gamma \rangle \models \sigma(\mathfrak{d})$ iff $\langle \mathbb{T}, \ell, \sigma\circ\gamma \rangle \models \mathfrak{d}$ the latter being guaranteed since $\langle \mathbb{T}, \ell \rangle \models \mathfrak{d}$. The last claim in the proposition is straightforward from earlier discussion. ■

Thus the external part of $Th^{\mathbb{D}}(\mathbb{B})$ extends **Lil** by some axioms. $Th^{\mathbb{D}}(\mathbb{B}) =$ **Lil** is mostly the case but cannot be concluded if $\mathbb{B}$ generates only trees of bounded depth. In fact, the equation holds iff there is no upper bound on depth of trees that $\mathbb{B}$ generates. Moreover, if $\mathbb{B}$ generates trees of depth $\leq n+1$ then $Th^{\mathbb{D}}(\mathbb{B})$ contains an axiom that guarantees models of size at most n (!).

Theorem 7.3.2 *$Th^{\mathbb{D}}(\mathbb{B})$ contains the following axioms.*

- *a for every $a \in \mathbb{L}(F)$ such that $\mathfrak{Fr}_{\mathbb{B}}(F)/EQ \models -a$*
- *$a \to b$ for every pair $a, b \in \mathbb{L}(F)$ such that $\mathfrak{Fr}_{\mathbb{B}}(F)/EQ \models b \leq a$*

Proof. Recall that $\mathfrak{Fr}_{\mathbb{B}}(F)/EQ$ is the algebra of labels of $\mathbb{B}$. So if $\mathfrak{Fr}_{\mathbb{B}}(F)/EQ \models -a$ then no node satisfies a whence it generates the relation $\top$. And if $\mathfrak{Fr}_{\mathbb{B}}(F)/EQ \models b \leq a$ then nodes satisfying b are nodes satisfying a, whence the relation generated by a is included in the relation generated by b. ■

This concerns the inside part of $Th^{\mathbb{D}}(\mathbb{B})$. The last theorem in this series connects inside and outside.

Theorem 7.3.3 $p \cup q. \leftrightarrow .p \wedge q \in Th^{\mathbb{D}}(\mathbb{B})$. ■

Recall also the following fact about definability.

Proposition 7.3.4 $\mathbb{D}_{\perp}(F) = \mathbb{D}_1(F), \mathbb{D}_{\top}(F) = \mathbb{D}_0(F), \mathbb{D}_{\wedge}(F) = \mathbb{D}_{\cup}(F)$. ■

7.4 The Extent of a Domain Axiom

The converse question of what classes of grammars are defined by a certain domains axiom is not a usual type of question and requires some rather unusual methods, too. To approach it we first deal with grammars which are *domain equivalent*. Two boolean CFGs $\mathbb{B}, \mathbb{C}$ are called **domain equivalent**, in symbols $\mathbb{B} \sim \mathbb{C}$, if $Th^{\mathbb{D}}(\mathbb{B}) = Th^{\mathbb{D}}(\mathbb{C})$. It will turn out that there are only finitely many equivalence classes of this relation. Each equivalence class contains a *unary grammar*. A grammar is called **unary** if all rules are of type $A \rightarrow \Gamma$ with $\sharp\Gamma = 1$; unary grammars generate linear strings rather than trees, or in other words, they generate non-branching trees. For each grammar $\mathbb{B} = \langle START, STOP, F, EQ, R\rangle$ we put $\mathbb{B}^\flat = \langle START, STOP, F, EQ, R^\flat\rangle$ with $R^\flat = \{A \rightarrow B | (\exists\Gamma, \Delta)(A \rightarrow \Gamma\ B\ \Delta \in R\}$, or alternatively, $R^\flat = \{A \rightarrow \bigcup\Gamma | A \rightarrow \Gamma \in R)\}$, with $\bigcup\Gamma = \bigcup\langle B | B \in \Gamma\rangle$.

Theorem 7.4.1 $\mathbb{B} \sim \mathbb{B}^\flat$.

Proof. Recall that $Th^{\mathbb{D}}(\mathbb{B}) = \bigcap\langle Th^{\mathbb{D}}(\langle\mathbb{T}, \ell\rangle) | \mathbb{B} \gg \langle\mathbb{T}, \ell\rangle\rangle$. Moreover, if β is a branch, then ℓ induces a labelling on β which we also denote by ℓ. Now the crucial observation is $Th^{\mathbb{D}}(\langle\mathbb{T}, \ell\rangle) = \bigcap\langle Th^{\mathbb{D}}(\langle\beta, \ell\rangle) | \beta$ a branch of $\mathbb{T}\rangle$. It is easy to verify that $\mathbb{B}^\flat$ generates the set of all branches of trees generated by $\mathbb{B}$. Hence

$$\begin{aligned} Th^{\mathbb{D}}(\mathbb{B}) &= \bigcap\langle Th^{\mathbb{D}}(\langle\beta, \ell\rangle) | \beta \text{ a branch of } \mathbb{T}, \mathbb{B} \gg \langle\mathbb{T}, \ell\rangle\rangle \\ &= \bigcap\langle Th^{\mathbb{D}}(\langle\beta, \ell\rangle) | \mathbb{B}^\flat \gg \langle\beta, \ell\rangle\rangle \\ &= Th^{\mathbb{D}}(\mathbb{B}^\flat). \ \blacksquare \end{aligned}$$

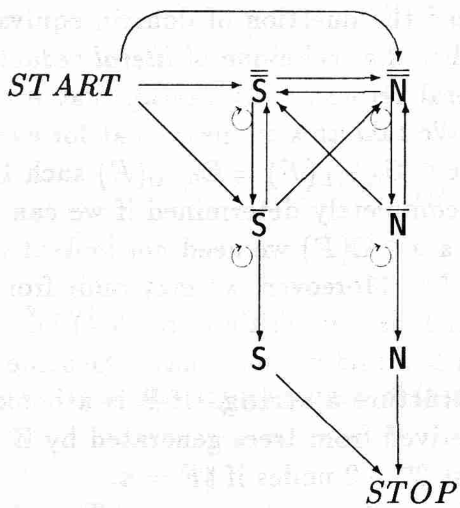
For the sake of concreteness let us take $\overline{X}$-syntax over two labels, S and N; it consists of the equation S $= -$N and the rules listed below. (We are assuming a head-final language here, we allow for recursion on non-branching symbols and we also exclude pruning ($\overline{\overline{X}} \rightarrow X$). This is of course only for the purpose of illustrating our point.) We write $\overline{\mathbb{X}}$(S,N) for this grammar.

$$\begin{array}{lll} START & \rightarrow & \text{BAR:2} \quad \text{S} \cap \text{BAR:1} \\ \text{S} \cap \text{BAR:2} & \rightarrow & \text{S} \cap (\text{BAR:2} \cup \text{BAR:1}) \\ \text{S} \cap \text{BAR:2} & \rightarrow & \text{BAR:2} \quad \text{S} \cap (\text{BAR:2} \cup \text{BAR:1}) \\ \text{S} \cap \text{BAR:1} & \rightarrow & \text{S} \cap (\text{BAR:1} \cup \text{BAR:0}) \\ \text{S} \cap \text{BAR:1} & \rightarrow & \text{BAR:2} \quad \text{S} \cap (\text{BAR:1} \cup \text{BAR:0}) \\ \text{N} \cap \text{BAR:2} & \rightarrow & \text{N} \cap (\text{BAR:2} \cup \text{BAR:1}) \\ \text{N} \cap \text{BAR:2} & \rightarrow & \text{BAR:2} \quad \text{N} \cap (\text{BAR:2} \cup \text{BAR:1}) \\ \text{N} \cap \text{BAR:1} & \rightarrow & \text{N} \cap (\text{BAR:1} \cup \text{BAR:0}) \\ \text{N} \cap \text{BAR:1} & \rightarrow & \text{BAR:2} \quad \text{N} \cap (\text{BAR:1} \cup \text{BAR:0}) \\ \text{BAR:0} & \rightarrow & STOP \end{array}$$

This amounts to a total of 28 rules in an ordinary CFG. Now the unary companion $\overline{\mathbb{X}}(\mathsf{S}, \mathsf{N})^\flat$ has the equation $\mathsf{S} = -\mathsf{N}$ and the rules

START	$\rightarrow$	BAR:2. ∪ .S ∩ BAR:1
S ∩ BAR:2	$\rightarrow$	S ∩ BAR:1
S ∩ BAR:1	$\rightarrow$	S ∩ (BAR:1 ∪ BAR:0)
N ∩ BAR:2	$\rightarrow$	N ∩ BAR:1
N ∩ BAR:1	$\rightarrow$	N ∩ (BAR:1 ∪ BAR:0)
BAR:1 ∪ BAR:2	$\rightarrow$	BAR:2
BAR:0	$\rightarrow$	*STOP*

This amounts to a total of 19 rules. Unary grammars can also be represented in a transition diagram or a directed graph.



An immediate corollary of Theorem 7.4.1 is that over a finite set F there exist only finitely many distinct classes of $\sim$.

Corollary 7.4.2 *Suppose that* $\natural F = n$. *Then there are at most* $2^{(2^n+1)^2} < 2^{2^{2n+1}}$ *grammars which are not domain equivalent.*

Proof. It suffices to count the number of unary grammars. Over n labels there are at most 2^n atoms in the boolean algebra of labels. A unary CFG over 2^n symbols is basically a directed graph over these symbols plus *START*, *STOP*. As *START* is always at the start of an arrow and *STOP* always at the end, we may for the purpose

of counting unary grammars conflate $START$ and $STOP$ in a single symbol. Then there are at most as many unary grammars as there are binary relations on a $1+2^n$-element set. This number is $2^{(2^n+1)(2^n+1)} = 2^{[2^n \cdot 2^n + 2\cdot 2^n + 1]} = 2 \cdot 2^{[2^{2n}+2^{n+1}]} < 2^{2^{2n+1}}$.
■

Write $\mathbb{B} \preccurlyeq \mathbb{C}$ if $Th^{\mathbb{D}}(\mathbb{B}) \subseteq Th^{\mathbb{D}}(\mathbb{C})$. Then $\mathbb{B} \sim \mathbb{C}$ if both $\mathbb{B} \preccurlyeq \mathbb{C}$ and $\mathbb{C} \preccurlyeq \mathbb{B}$. Also, if $\mathfrak{d} \in Th^{\mathbb{D}}(\mathbb{B})$ then also $\mathfrak{d} \in Th^{\mathbb{D}}(\mathbb{C})$ by definition and hence intents of domain axioms are upward closed with respect to $\leq$.

Theorem 7.4.3 *Suppose that* $\mathbb{B} = \langle START, STOP, F, EQ, R\rangle$ *and* $\mathbb{C} = \langle START, STOP, F, EQ^*, R^*\rangle$. *Then if* $EQ \subseteq EQ^*$ *and* $R^* \subseteq R$, $\mathbb{B} \preccurlyeq \mathbb{C}$.

Proof. Suppose first $R = R^*$. If $EQ \subseteq EQ^*$ then every precisification of a rule ρ of $\mathbb{C}$ is a precisification of the same rule ρ of $\mathbb{B}$. Thus every tree generated by $\mathbb{C}$ is a tree generated by $\mathbb{B}$. This holds a fortiori if $R \supseteq R^*$. Thus $Th^{\mathbb{D}}(\mathbb{B}) \subseteq Th^{\mathbb{D}}(\mathbb{C})$. ■

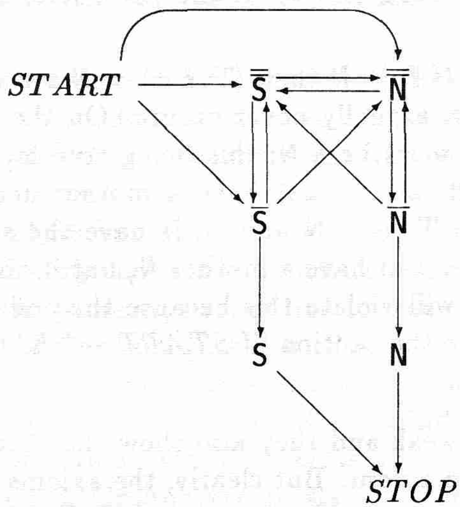
Now that we have reduced the question of domain equivalence to unary grammars we should also remember the technique of *literal reduction* (see § 4). We can summarize the effect of literal reduction by saying that every command relation $\mathfrak{d} \in \mathbb{D}(F)$ is **locally tight**. We take this to mean that for every labelled tree $\langle \mathbb{T}, \ell\rangle$ and every $x \in T$ there is a $\mathfrak{e} \in \mathbb{D}_{\wedge,\top,\perp}(F) = \mathbb{D}_{0,1,\cup}(F)$ such that $\langle \mathbb{T}, \ell, x\rangle \models \mathfrak{d} = \mathfrak{e}$. It should be clear that $\mathfrak{d}$ is completely determined if we can name $\mathfrak{e}$ depending on $\langle \mathbb{T}, \ell, x\rangle$. To determine $\mathfrak{e}$ for a $\mathfrak{d} \in \mathbb{D}(F)$ we need not look at the entire tree; rather, suffices to take the position $\uparrow x$. Moreover, we may omit from that part of the upward cone of x all nodes which are not critical (see § 4) i.e. all nodes m for which there is a n such that $m > n > x$ and m and n have the same label(s). Let us agree to call such a reduced substructure a **string**. If $\mathbb{B}$ is a boolean grammar, we call the set $\sigma(\mathbb{B})$ of all strings derived from trees generated by $\mathbb{B}$ the $\mathbb{B}$**–strings**. Note that a $\mathbb{B}$**–string** has at most $2^n + 2$ nodes if $\sharp F = n$.

Proposition 7.4.4 *If* $\mathbb{B}$ *and* $\mathbb{C}$ *have the same strings then* $\mathbb{B} \sim \mathbb{C}$. ■

Corollary 7.4.5 *Let* $\mathbb{B}$ *be a unary grammar; put* $\mathbb{B}^\circ = \langle START, STOP, F, EQ, R^\circ\rangle$ *with* $R^\circ = R - \{A \to A | A \in \mathbb{L}(F)\}$. *Then* $\mathbb{B} \sim \mathbb{B}^\circ$. ■

Thus immediate loops can be dropped. To continue our example, not only is $\overline{\mathbb{X}}(\mathsf{S}, \mathsf{N})$ domain equivalent with its unary companion $\overline{\mathbb{X}}(\mathsf{S}, \mathsf{N})^\flat$ but also with the grammar $\overline{\mathbb{X}}(\mathsf{S}, \mathsf{N})^{\flat\circ}$ where the one element loops (i. e. immediate recursions) are eliminated. This grammars has only 15 rules and can be written as follows.

START	$\to$	S ∩ BAR:1. ∪ .N ∩ BAR: 2
S ∩ BAR:2	$\to$	S ∩ BAR:1. ∪ .N ∩ BAR: 2
S ∩ BAR:1	$\to$	S ∩ BAR:0. ∪ .BAR: 2
N ∩ BAR:2	$\to$	N ∩ BAR:1. ∪ .S ∩ BAR: 2
N ∩ BAR:1	$\to$	N ∩ BAR:0. ∪ . ∩ BAR: 2
BAR:0	$\to$	*STOP*



An essential corollary of all this is the following.

Corollary 7.4.6 *'$\mathbb{B} \sim \mathbb{C}$' is decidable; moreover, it is possible to list all equivalence classes in finite time.*

Proof. The notation '$\mathbb{B} \sim \mathbb{C}$' is used to denote *the problem whether or not* $\mathbb{B} \sim \mathbb{C}$. Now any $\mathfrak{d} \in \mathbb{D}(F)$ is locally tight. Hence there exist only finitely many distinct $\mathbb{D}(F)$. The problem '$\mathbb{B} \models \mathfrak{d}$' is also decidable by first computing the $\mathbb{B}$-strings and then deciding '$\sigma(\mathbb{B}) \models \mathfrak{d}$', which is a problem that requires only finitely many computations. Hence, by all this '$\mathbb{B} \preccurlyeq \mathbb{C}$' is decidable and so is '$\mathbb{B} \sim \mathbb{C}$'. To calculate the equivalence classes we only look at the unary representatives. ■

7.5 Conditions Expressed by One- and Two-Letter Formulae

Recall that there are only six **Li**ℓ-formulae based on a single letter p. Below we map out the condition on $\mathbb{B}$ expressed by such formulae; we can also say that we give a description of the extent of these axioms. (Temporarily, we use $X \twoheadrightarrow Y$ to denote the fact that there is a rule ρ of which $X \to \Gamma\, Y \Delta$ is a precisification for some Γ, Δ. $X \twoheadrightarrow^+ Y$ is the transitive closure of this relation.)

AXIOM	EXTENT
N	not: $START \twoheadrightarrow^+ \mathsf{N}$
$\neg\,\mathsf{N}$	not: $START \twoheadrightarrow^+ -\mathsf{N}$
$\neg\neg\mathsf{N}$	not: $START \twoheadrightarrow^+ -\mathsf{N}$
$\mathsf{N} \vee \neg\mathsf{N}$	if $START \twoheadrightarrow^+ \mathsf{N}$ then not: $\mathsf{N} \twoheadrightarrow -\mathsf{N}$

The reasoning is as follows. If $\mathbb{B} \models \mathsf{N}$ then $\langle \mathbb{T}, \ell, x\rangle \models \mathsf{N}$ for all $\mathbb{B} \gg \langle \mathbb{T}, \ell\rangle$ and all $x \in T$. This is true only if N actually never occurs. On the other hand, $\mathbb{B} \models \neg\mathsf{N}$ means that the mother of x must be a N; this being true for all x means $-\mathsf{N}$ can never occur. For $\mathbb{B} \models \neg\neg\mathsf{N}$, x may not have a mother node satisfying N since then $f_{\neg\neg\mathsf{N}}(x)$ is its mother. Thus, $\neg\mathsf{N}$ and $\neg\neg\mathsf{N}$ have the same extent! Finally, $\langle \mathbb{T}, \ell\rangle \models \mathsf{N} \vee \neg\mathsf{N}$ means that x can have a mother N, but if not, there is no N node above x. Any rule $\mathsf{N} \twoheadrightarrow -\mathsf{N}$ will violate this because the node satisfying $-\mathsf{N}$ has a daughter. It is crucial to have the caution 'if $START \twoheadrightarrow^+ \mathsf{N}$' to make sure that the rule $\mathsf{N} \twoheadrightarrow -\mathsf{N}$ is actually used.

These axioms are rather weak and they also show that there is no obvious way to tell the extent of a domain axiom. But clearly, the axioms of type N do express something relevant in grammar, namely that non-occurring labels define the total domain. Moreover, $\mathsf{N} \vee \neg\mathsf{N}$ has a natural interpretation when we look at BAR: 2. Obviously, BAR: 0 $\vee$ $\neg$BAR: 0 is an axiom that is valid in $\overline{X}$-syntax!

For two constants S, N we get the following table. This time we drop the condition 'if $START \twoheadrightarrow^+ X$'. We assume that we are only talking about grammars in which both symbols occur at least in one generated tree.

AXIOM	EXTENT
$\neg(\mathsf{S} \wedge \mathsf{N})$	$-\mathsf{N} \cap -\mathsf{S} = 1$
$\neg\mathsf{S} \vee (\mathsf{N} \to \mathsf{S})$	not: $\mathsf{N} \twoheadrightarrow^+ \mathsf{S}$ if there is a rule $\mathsf{S} \twoheadrightarrow -\mathsf{N} \cap -\mathsf{S}$ and if $\mathsf{S} \leq \mathsf{N}$
$\mathsf{S} \to \mathsf{N}. \vee .(\mathsf{S} \vee \neg\mathsf{N})$	not: $START \twoheadrightarrow^+ \mathsf{N}$ if there is a rule $\mathsf{N} \to -\mathsf{N} \cap -\mathsf{S}$ and if $\mathsf{N} \leq -\mathsf{S}$
$\neg\mathsf{N} \to \neg\mathsf{S}$	not: $START \twoheadrightarrow^+ \mathsf{N}$ if $\mathsf{N} \leq -\mathsf{S}$
$\neg\neg(\mathsf{N} \vee \mathsf{S})$	$\mathsf{N} \leq -\mathsf{S}$
$\neg\mathsf{S} \to (\mathsf{N} \vee \neg\mathsf{N})$	not: $\mathsf{N} \twoheadrightarrow^+ \mathsf{S}$ if $\mathsf{S} \leq -\mathsf{N}$
$(\mathsf{N} \vee \mathsf{S}. \to .\mathsf{N} \wedge \mathsf{S}). \to .\mathsf{N} \vee \neg\mathsf{N}$	$\mathsf{N} \twoheadrightarrow -\mathsf{N} \cap -\mathsf{S}$ and $\mathsf{N} \leq \mathsf{S}$

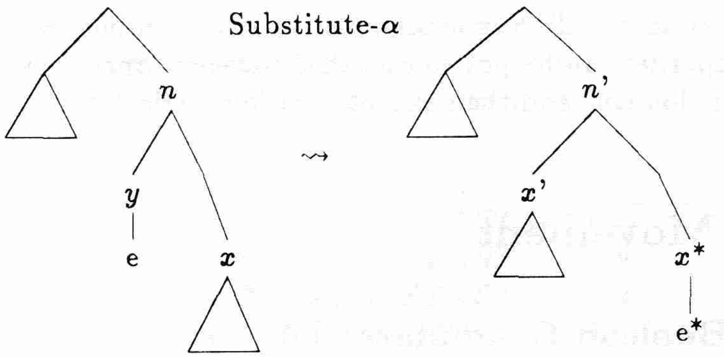
Here we meet a number of known axioms. Of course, $\neg(\mathsf{N} \wedge \mathsf{S}) = \neg(\mathsf{N} \cup \mathsf{S})$ so that the extent is that of a one-letter formula. Most of the axioms allows us to state that a transition from one node N to a node S is never possible, not even indirectly. It is hard to see how such properties can be put to use; thus we are warned now that the correspondence between domains and their extent is rather intricate.

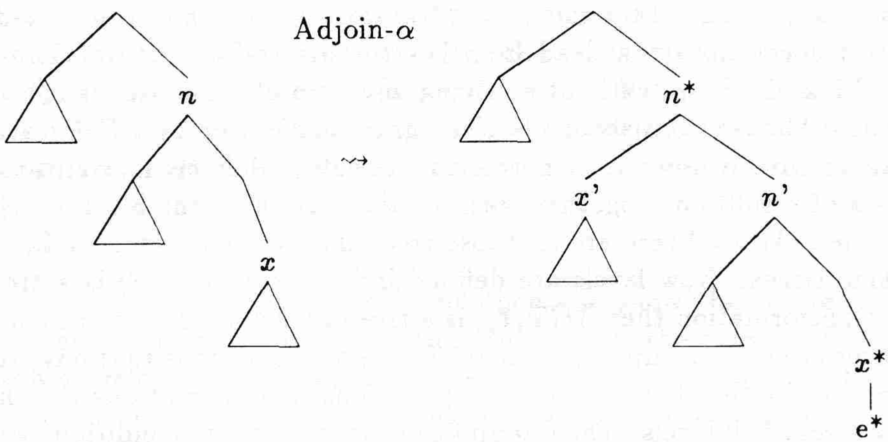
8 Domains for Movement

8.1 Movement in Boolean Grammars

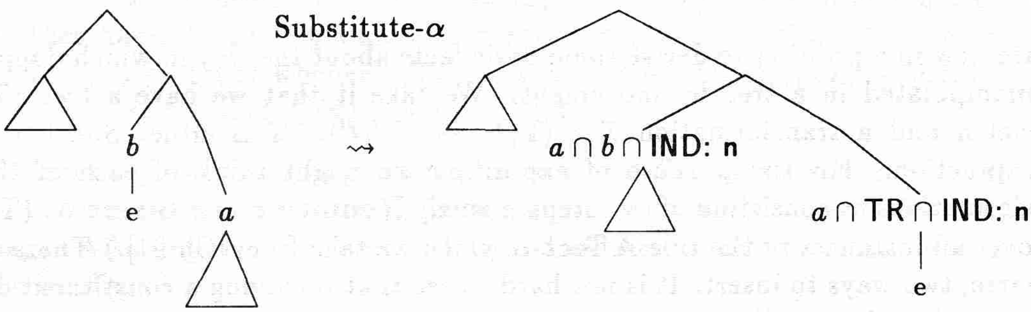
Let us now concentrate on one central aspect of transformational grammar – that of movement. Although modern versions of transformational grammar are more complex the essential problem of the movement transformation can be illustrated by looking at the transformations that lead from D-structure to S-structure. Thus we ignore the levels LF and PF but without implying any theoretical consequences. We say that a **bistratal theory** consists of a boolean grammar $\mathbb{B}$ over a set F of features – the so-called **base component**, a set of tree-to-tree rules called **transformations** and, finally, a set of conditions together with a specification of the level at which they have to be met. At level zero are all those trees that are generated by $\mathbb{B}$; they are called **D-structures**. Now levels are defined inductively; if $\langle \mathbb{T}, \ell \rangle$ is a tree of level n and $\mathfrak{T}$ a transformation then $\mathfrak{T}(\langle \mathbb{T}, \ell \rangle)$ is a tree of level $n+1$. Conditions are classified into three groups. Group (α) consists of all those conditions that have to be met at level zero, i.e. by the D-structure, Group (ι) consists of conditions that have to be fulfilled by trees of all levels. The Group (ω) consists of those conditions which define an S-structure. They need only be satisfied when we want to stop applying transformations. An S-structure then necessarily satisfies conditions of type ι and ω. One may wonder about the necessity to have conditions of type α when in fact they have to be satisfied by the D-structures. There are two reasons; first, the idea is that the burden of generating 'correct' D-structures is shared between the base-component and the α-conditions in the hope to make both individually small and attractive, and second, there might be conditions that cannot be guaranteed to hold by using a context-free grammar and so they have to be added explicitly.

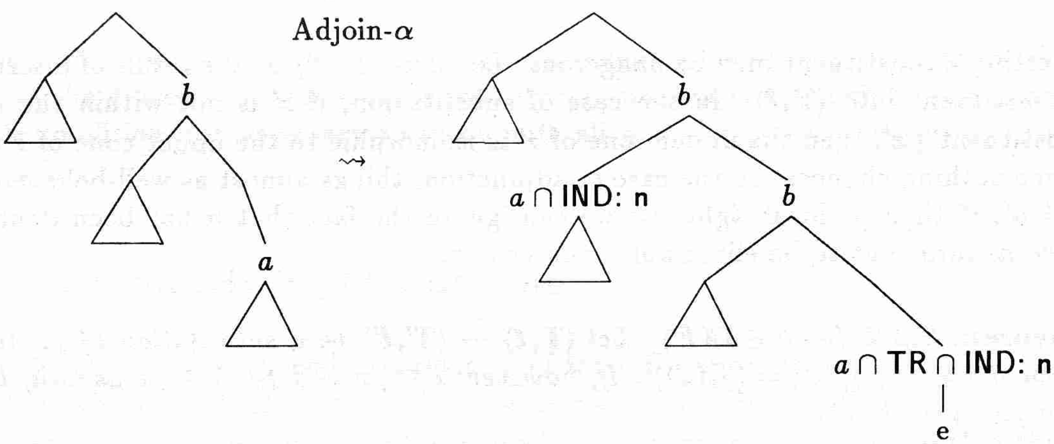
We have extensively dealt with the base component. We will now attack the problems that are created by the transformations. It is currently assumed that there is only one transformation called **Move-α**. Although its action is mostly described as *move anything anywhere* this slogan is not quite accurate. In fact, Move-α is a conglomerate of two rules only one of which deserves this name. We dub the rules **Substitute-α** and **Adjoin-α**. The idea of Substitute-α is to take a constituent $\downarrow x$ and tag it onto another constituent in the existing tree. The idea of Adjoin-α is to take any constituent and Chomsky-adjoin it to an existing node. Both actions are pictured below. It is agreed that both transformations leave behind an α-trace bearing an index identical to the index tagged onto the moved constituent.





The way the notation is designed is as follows. We have put a prime onto every node that has been moved somewhere in the tree and a star onto a node that is newly created from an existing one. Note that it might be convenient to view substitution as a transformation where two constituents change places. It is then apparent that substitution never results in a tree that has more nodes than the original one. But for conceptual reasons it is better not to think this way. There are a number of additional details to be filled in as concerns the labels for the nodes. We will generally assume that a transformation will relate a certain node z with a certain node z' and that $\ell(z) = \ell(z')$. Exceptions are only made with the nodes marked in the picture. e always denotes a terminal node, so it has no label, likewise e'. First substitution. Central are x' and x^*. We will take it that x and y are somehow merged into one node and consequently $\ell'(x') = \ell(x) \cap \ell(y)$ (cf. the discussion below). For x^* there seems to be no immediate solution; in that case we opt for $\ell'(x^*) = \ell(x)$. Now for adjunction. Here it is clear that $\ell'(x') = \ell'(x^*) = \ell(x)$ and that $\ell'(n') = \ell'(n^*) = \ell(n)$. However, there are now extra features to be distributed. The first is the feature TR indicating that an element has been emptied by a transformation. One might think of TR as a flag set to inhibit further movement into this empty space. Second, the moved constituent and the trace both receive an identical index IND: n, $n \in \omega$. Although both TR and IND: n are features, they are not seen as features which are assigned by $\mathbb{B}$. There might be reasons to incorporate TR into the base-component, but definitely indices are out since they immediately create non-context freeness. But we will not dwell on that issue here. Let us just picture the resulting trees with the appropriate labellings.





Note that this indeed allows movement from anywhere into any empty constituent. However, an NP may not move into an empty VP position and so there clearly are restrictions. This particular movement is in fact ruled out by a requirement that all new assignments are consistent. Moreover, we will force a stronger condition which we call the **as-if-principle**. The idea is that as regards the features of F (i.e. excluding TR, IND: n, and possibly others) the tree that is the output of the transformation is such that it could have been generated by the base-component. To formalize this principle let $\ell \cap F$ denote the assignment which gives as output for x exactly $\ell(x) \cap F$.

[AS-IF] For every tree $\langle \mathbb{T}, \ell \rangle$ of level $n \in \omega$ the tree $\langle \mathbb{T}, \ell \cap F \rangle$ is a D-structure, that is, $\mathbb{B} \gg \langle \mathbb{T}, \ell \cap F \rangle$.

This implies among other the following principle.

[CONSISTENCY] For every tree $\langle \mathbb{T}, \ell \rangle$ of level $n \in \omega$ and any $x \in T$, $\mathfrak{Tr}_{\mathbb{B}}(F)/EQ \nvDash -\ell(x)$.

8.2 Interaction of Movement with Domains

We are now in a position to derive some basic facts about the way in which domains are manipulated in a tree by movement. We take it that we have a tree $\langle \mathbb{T}, \ell \rangle$ of level n and a transformation $\mathfrak{T} : \langle \mathbb{T}, \ell \rangle \rightsquigarrow \langle \mathbb{T}', \ell' \rangle$. $\mathfrak{T}$ is either Substitution or Adjunction. For the purpose of exposition we might think of each of these transformations as consisting of two steps, namely **Remove-α** and **Insert-α**. (Thus we cover all instances of the rule **Affect-α** which we take from [Ch 91].) There are, of course, two ways to insert. It is not hard to see that removing a constituent does not affect domains at all.

Proposition 8.2.1 *Let $\mathfrak{d} \in \mathbb{D}(F)$ and $\langle \mathbb{T}, \ell \rangle$ a labelled tree. If $\langle \mathbb{T}', \ell' \rangle$ is nothing but $\langle \mathbb{T}, \ell \rangle$ minus a constituent, then for every $x' \in T'$ $f'_{\mathfrak{d}}(x') = (f_{\mathfrak{d}}(x))'$.*

Inserting a constituent may be dangerous. Let thus $\langle \mathbb{T}', \ell' \rangle$ be the result of inserting a constituent into $\langle \mathbb{T}, \ell \rangle$. In the case of substitution, if z' is not within the new constituent $\downarrow x'$, then the upper cone of z' is isomorphic to the upper cone of z and hence nothing changes. In the case of adjunction, things almost as well-behaved. If $z \neq n', n^*$ then by local tightness we can ignore the fact that n has been doubled since n' turns out to be eliminable (non-critical).

Theorem 8.2.2 *Let $\mathfrak{d} \in \mathbb{D}(F)$. Let $\langle \mathbb{T}, \ell \rangle \rightsquigarrow \langle \mathbb{T}', \ell' \rangle$ be a substitution of $\downarrow x$ to n. Then if $z \notin \downarrow x$, $f'_{\mathfrak{d}}(z') = (f_{\mathfrak{d}}(z))'$. If, however, $z \in \downarrow x$ and $f_{\mathfrak{d}}(z) \in \downarrow x$ as well, then also $f'_{\mathfrak{d}}(z') = (f_{\mathfrak{d}}(z))'$.*

Proof. The last claim needs to be investigated. It follows from the next lemma. ■

Lemma 8.2.3 *Let $\langle \beta, \ell \rangle$ be a position. Suppose that $f^{\beta}_{\mathfrak{d}}(x) < y$ for some $\mathfrak{d}, x, y$. Now let $\gamma = [x, y]$ be an interval. Then $f^{\gamma}_{\mathfrak{d}}(x) = f^{\beta}_{\mathfrak{d}}(x)$.*

Proof. By induction on $\mathfrak{d}$ we prove $f^{\gamma}_{\mathfrak{d}}(x) = f^{\beta}_{\mathfrak{d}}(x)$ if $f^{\beta}_{\mathfrak{d}}(x) < y$, and $f^{\gamma}_{\mathfrak{d}}(x) = y$ else. This certainly holds for the primitives $L, L \in \mathbb{L}(F)$ and so also for $\top, \bot$. If $\mathfrak{d} = \mathfrak{a} \wedge \mathfrak{b}$ then $f^{\beta}_{\mathfrak{d}}(x) = \min \{f^{\beta}_{\mathfrak{a}}(x), f^{\beta}_{\mathfrak{b}}(x)\}$. Assume that $f^{\beta}_{\mathfrak{d}}(x) = f^{\beta}_{\mathfrak{a}}(x) < y$. Then by induction hypothesis $f^{\beta}_{\mathfrak{a}}(x) = f^{\gamma}_{\mathfrak{a}}(x)$. If $f^{\gamma}_{\mathfrak{b}}(x) < f^{\gamma}_{\mathfrak{a}}(x)$ then $f^{\gamma}_{\mathfrak{b}}(x) = f^{\beta}_{\mathfrak{b}}(x)$ which contradicts our assumptions. Thus $f^{\gamma}_{\mathfrak{a}}(x) \leq f^{\gamma}_{\mathfrak{b}}(x)$ and so $f^{\gamma}_{\mathfrak{d}}(x) = f^{\beta}_{\mathfrak{d}}(x)$. Finally, assume $f^{\beta}_{\mathfrak{d}}(x) \geq y$. Then both $f^{\beta}_{\mathfrak{a}}(x), f^{\beta}_{\mathfrak{b}}(x) \geq y$ from which $f^{\gamma}_{\mathfrak{a}}(x), f^{\gamma}_{\mathfrak{b}}(x) = y$ both follow by induction hypothesis and then $f^{\gamma}_{\mathfrak{d}}(x) = y$. Now let $\mathfrak{d} = \mathfrak{a} \vee \mathfrak{b}$. If $f^{\beta}_{\mathfrak{d}}(x) < y$ then both $f^{\beta}_{\mathfrak{a}}(x), f^{\beta}_{\mathfrak{b}}(x) < y$ and so by IH $f^{\gamma}_{\mathfrak{a}}(x) = f^{\beta}_{\mathfrak{a}}(x), f^{\gamma}_{\mathfrak{b}}(x) = f^{\beta}_{\mathfrak{b}}(x)$ and the conclusion follows. However, if $f^{\beta}_{\mathfrak{d}}(x) \geq y$ then for at least one of the two, say $\mathfrak{a}$, $f^{\beta}_{\mathfrak{a}}(x) \geq y$ from which by IH $f^{\gamma}_{\mathfrak{a}}(x) = y$ and so $f^{\gamma}_{\mathfrak{d}}(x) = y$ as well. Finally, $\mathfrak{d} = \mathfrak{a} \rightarrow \mathfrak{b}$. Suppose first $f^{\beta}_{\mathfrak{d}}(x) < y$. Then obviously $f^{\beta}_{\mathfrak{d}}(x) = f^{\beta}_{\mathfrak{b}}(x) < y$ and by IH $f^{\gamma}_{\mathfrak{b}}(x) = f^{\beta}_{\mathfrak{b}}(x) < y$. $f^{\gamma}_{\mathfrak{a}}(x) \leq f^{\gamma}_{\mathfrak{b}}(x)$ will lead to a contradiction since then $f^{\gamma}_{\mathfrak{a}}(x) = f^{\beta}_{\mathfrak{b}}(x)$. Consequently, $f^{\gamma}_{\mathfrak{d}}(x) = f^{\gamma}_{\mathfrak{b}}(x) = f^{\beta}_{\mathfrak{b}}(x) = f^{\beta}_{\mathfrak{d}}(x)$, as required. If on the other hand $f^{\beta}_{\mathfrak{d}}(x) \geq y$ then either $f^{\beta}_{\mathfrak{b}}(x) < y$ and so $f^{\beta}_{\mathfrak{a}}(x) \leq f^{\beta}_{\mathfrak{b}}(x) < y$ and by IH $f^{\gamma}_{\mathfrak{a}}(x) \leq f^{\gamma}_{\mathfrak{b}}(x)$ whence $f^{\gamma}_{\mathfrak{d}}(x) = y$ or indeed $f^{\beta}_{\mathfrak{a}}(x), f^{\beta}_{\mathfrak{b}}(x) \geq y$ from which $f^{\gamma}_{\mathfrak{a}}(x) = f^{\gamma}_{\mathfrak{b}}(x) = y$ and $f^{\gamma}_{\mathfrak{d}}(x) = y$ as well. ∎

This lemma presents an important property of definable CRs, namely that if the value of the generating function for some x is z then everything above z can be changed with impunity with respect to z; thus the computation can be carried out quite locally.

Theorem 8.2.4 *Let $\mathfrak{d} \in \mathbb{D}(F)$ and $\langle \mathbb{T}, \ell \rangle \rightsquigarrow \langle \mathbb{T}', \ell' \rangle$ an adjunction of $\downarrow x$ to n. Then if $z \notin \downarrow x$ and $z \neq n$, $f'_{\mathfrak{d}}(z') = (f_{\mathfrak{d}}(z))'$. If $z \in \downarrow z$ as well as $f_{\mathfrak{d}}(z) \in \downarrow x$ then also $f'_{\mathfrak{d}}(z') = (f_{\mathfrak{d}}(z))'$.* ∎

To cut this story short, movement is only problematic for those domains which go across the moved constituent. It is, however, in presence of something like a strict-cycle condition not necessary to recompute all domains since they will not all be needed.

8.3 Absolutely Tight Domains

In [Ch 86], Chomsky considers a revision of the GB theory via a notion of a barrier that should do multiple duty in defining domains. The idea is that in general the ideal domains for movement are those in which no barrier is crossed and that things get worse the more barriers intervene. As we will see, subjacency is undefinable and thus any domain which allows crossing an inherent barrier does not fit well our framework. In the next chapter we will investigate this situation. For now we will put to test the notion of 0-subjacency. We will restrict our discussion to inherent barriers as they can be defined in absolute terms, that is, not relative to ones position in a tree. Following Chomsky, any blocking category different from IP is an inherent barrier. Introducing a feature BC helps to define the inherent barriers as IBR = BC$\cap -$INFL. That we may not cross an inherent barriers means in domain terminology that the domain for movement is IBR-command. A transformation Substitute-α or Adjoin-α respects this domain if it substitutes or adjoins x to n and n is in the domain of x; in the present case this means that x inherent-barrier-commands n. The original conception was that those barriers are quite frequent;

most maximal projections turn out to be barriers. But a constituent may escape such a barrier by adjoining to it; this is spelled out in the notion of *inclusion*. Adjunction 'stretches' the n into n' and n^* which have equal labelling and thereby count as one; let us say that a maximal convex subset $\mathfrak{n}$ such that all $x \in \mathfrak{n}$ have the same labelling is a **quasi-node**. And let us say that $\mathfrak{n}$ **includes** $\mathfrak{m}$ – in symbols $\mathfrak{m} \leq \mathfrak{n}$ – if for all $m \in \mathfrak{m}$ and for all $n \in \mathfrak{n}$ $m < n$; and that $\mathfrak{n}$ **excludes** $\mathfrak{m}$ if for no $m \in \mathfrak{m}$ and no $n \in \mathfrak{n}$ we have $m < n$. Then if x is adjoined to n, the quasi-node containing n', $\mathfrak{N}$ does not include the quasi-node of x' and thus does not count as a barrier for x. (This is reminiscent of 'tunneling' in quantum-mechanics where although a black hole constitutes an absolute death threat to all particles, they may, due to their part-time character of waves still escape.) Such an analysis as proposed by Chomsky is not internalizable in this system of domains for the simple reason that the node x' does not have any access to information about n' since $x' \not\leq n'$. Thus x' can only judge from n^* where it can go. (See also next chapter.) But now, as barriers are defined purely in terms of labels, n^* has the same label as n' and hence inherits barrierhood from n'. Now x' could still adjoin to n^* – only to find out that the newly created n^{**} is again a barrier. In this way, no constituent ever stands a chance of escaping a barrier. Let us state this rather explicitly by defining first that a CR is **absolutely tight** if it cannot be escaped by substitution nor adjunction.

Theorem 8.3.1 *A command relation is* TIGHT *if and only if it is* ABSOLUTELY TIGHT. ■

This theorem states explicitly that the idea of a inherent barrier being a barrier only half-way is not viable. We will, for the sake of comparison, nevertheless sketch a way to circumvent this dilemma. This solution comprises in giving up the AS-IF partly; we assume that in the base-component every node is given the label BASE and that the effect of adjunction will be to leave the labelling of n' intact but to redefine the label of n^* as being $-$BASE. Then a barrier can only be a node which carries the label BASE. By adjunction, the moved constituent is dominated by a non-base node which cannot be a barrier for that constituent any more.

But there are also more problems with the barriers approach. As [Li 88] point out, its success rests on a number of assumptions on adjunction. We must in particular assume that we may not adjoin more than once an NP to a VP and that we may not adjoin to certain nodes etc. But if movement is freely applicable, what processes are there to stop adjunction from being repeated? In particular, if one assumes that transformations need as input all and only the information that is in the tree we need something in the tree that stops adjunction. But in our formulation there is nothing that does so. For the VP there is help if we forbid adjunction to VP and generate an extra empty node to which NPs may be substituted. But that leaves us with the problem of banning adjunction when we do not want it. One may use the AS-IF principle, but that will be rather desastrous for VP as it is clear that we do want at least base-generated VP-adjuncts.

The moral is that adjunction is dangerous in that it may overgenerate and has therefore be out on leash. But what is more, there is the empirical question of how we are going to prove that the traces left behind by adjunction can be attested. Can there be decisive evidence that proves that elements must adjoin to each barrier rather than moving to their eventual landing site in one fell swoop? It appears here that the original attraction of GB in explaining phenomena of non-extractability is lost. It was assumed then that there is a limited number of escape hatches which elements may substitute into on their way out, thereby blocking the same possibility for other elements by leaving a trace. Such arguments are not available for adjunction, since adjunction creates its own escape hatches. Thus we should be aiming at a definition of domains for movement which do not create the need for intermediate adjunction. Such a domain is available in the present case; we just have to define a barrier in the new sense to be any barrier in the old sense to which adjunction is ruled out. This leads more or less to the system of relativised minimality in [Rz 90], the difference being that movement may not go further than to the next governing node. Later we will show how to handle this requirement as well.

8.4 Small Print

The definition $\ell'(x') = \ell(x) \cap \ell(y)$ seems to be problematic in the case of head-movement. For example, if a V moves into INFL we should expect as a result a label $\mathsf{V} \cap \mathsf{INFL}$ and not INFL. This certainly disappoints some of the expectations about movement but it is not altogether unreasonable. The problem lies here precisely in the fact that category labels are assumed to be mutually exclusive and so for example the attribute $\mathsf{V} \cap \mathsf{INFL}$ is contradictory and so V-to-I-movement is disallowed on the basis of CONSISTENCY. But precisely the assumption that category labels exclude each other is questionable. It might be correct with respect to major categories such as N and V – so that indeed transformations such as N-to-V-movement are excluded –; but it certainly is problematic with respect to INFL and V since INFL exists for theory internal reasons and thus there is no a priori reason to force INFL to be inconsistent with V even less so when they can be co-instantiated in a single lexeme. Even though this may hurt other theories it is quite plausible to assume that only lexical categories are exclusive while there is still some choice as to which functional categories may co-occur and which of them can co-exist with which lexical category. Such assumptions naturally constrain the choices for head-movement. A lexical head can move to only to a lexical head position of identical category and furthermore it may visit some functional head positions.

This discussion is based on the assumption that CONSISTENCY holds – which we derived from AS-IF. As concerns AS-IF it is explicitly rejected in [Ch 86]: 'The X-bar constraints are satisfied at D-structure, but not at either levels of representation if adjunction has taken place in a derivation.' (p. 3.) Given our discussion it seems, however, that rather than adjunction only head-to-head movement is a prima facie violation of AS-IF. Incidentally, in [Ch 91] Chomsky seems to think that AS-IF is indeed valid (at least until S-structure).

9 Problems with Subjacency

9.1 Outline of Subjacency

So far we have created the impression that $\mathbb{I}$-definable relations are the type of relations to which all linguistically relevant notions of nearness are reducible. There are, however, clear cases where this is not so. Let us take an arbitrary example of a definition of subjacency, in this case from [vR 86]:

[SUBJACENCY CONDITION] No rule can relate X, Y in the structure

$\ldots X \ldots [_{\alpha} \ldots [_{\beta} \ldots Y \ldots$ (or: $\ldots Y \ldots]_{\beta} \ldots]_{\alpha} \ldots X \ldots$)

where α and β are bounding nodes.

This rather crucial condition has been reformulated several times; quite thoroughly so in [Ch 86] where a rather sophisticated definition of subjacency is given which we will examine below. For now we stick to the one given above. This definition presents an instance of counting intervening nodes; calling x **n-subjacent** to y if the $n+1^{st}$ bounding node strictly above x dominates y then the SUBJACENCY CONDITION says that a rule can only relate x and y if one is at least 1-subjacent to the other. Let us rephrase this in our terminology of relations by introducing a label BD which is true of exactly the bounding nodes; then under standard assumptions for English BD = BAR: 2 $\cap$ (N $\cup$ COMP). We now define a CR N-SUB via its associated function f_n. For every node x, f_n picks the $n+1^{st}$ BD-node strictly above x (else r).

[SUBJACENCY CONDITION] If a rule relates x, y in a structure then x 1-SUB-commands y.

What remains to be investigated is the CR N-SUB. It is quite clear that N-SUB is locally tight if and only if $n = 0$ in which case we have a relation better known as KOMMAND (see [La 76]). The other relations – among which we find our desired 1-subjacency – are thus not $\mathbb{I}$-definable. It is therefore necessary to have formal means to generate these relations as well. The most sobre construction is that of *relational composition.* Given two binary relations $R, S \in Re_2(\mathbb{T})$ we define the **composition** $R \circ S$ to be the set of pairs $\langle x, y \rangle$ such that there is a z with xRz and zSy. Before we engage in a formal analysis let us see how this fixes our problems. The key is to observe that f_n can be recursively defined by

$$\begin{aligned} f_0(x) &= f_{\mathsf{BD}}(x) \\ f_{n+1}(x) &= f_{\mathsf{BD}}(f_n(x)) = (f_{\mathsf{BD}} \circ f_n)(x) \end{aligned}$$

Here, $f_{\mathsf{BD}} \circ f_n$ denotes the composition of the associated functions, which is defined in exactly the same way. The next step is to show that the composition of the associated functions is indeed the associated function of the likewise composed relations; this proof will be given below. Thus we get

$$\begin{aligned} f_0 &= f_{\text{0-SUB}} &= f_{\mathsf{BD}} \\ f_{n+1} &= f_{\mathsf{BD}} \circ f_n &= f_{\mathsf{N\text{-}SUB \circ BD}} \end{aligned}$$

Proposition 9.1.1 *The relations* N-SUB *are recursively defined by*

$$\begin{aligned} \text{0-SUB} &= \mathsf{BD} \\ \text{N+1-SUB} &= \mathsf{N\text{-}SUB} \circ \mathrm{BD}. \ \blacksquare \end{aligned}$$

Using exponential notation we might write BD^{n+1} to denote the n-fold composition of BD with itself.

9.2 Barriers

The power and simplicity of our domain language will be demonstrated with an example which for many people is the prime example of an unformalizable theory – that of barriers as introduced in [Ch 86]. Below we will list the definitions of that book which define barriers. (At least these are the initial definitions; the proposal in [Ch 86] consists in a series of alternative definitions, which can all be treated in a similar way.)

(26) γ is a **barrier** for β iff either (a) or (b):
a. γ immediately dominates δ, δ a BC for β,
b. γ is a BC for β, $\gamma \neq$ IP.

(25) γ is a **BC** for β iff γ is not L-marked and γ dominates β.

"We understand γ in (25) and (26) to be a maximal projection, and we understand "immediately dominate" in (26a) to be a relation between maximal projections (so that γ immediately dominates δ in this sense even if a nonmaximal projection intervenes)." (p. 14 - 15 loc. cit.) The use of barriers lies in the notion of n-subjacency which is defined as follows.

(59) β is n-**subjacent** to α iff there are fewer than $n+1$ barriers for β that exclude α.

(17) α **excludes** β if no segment of α dominates β.

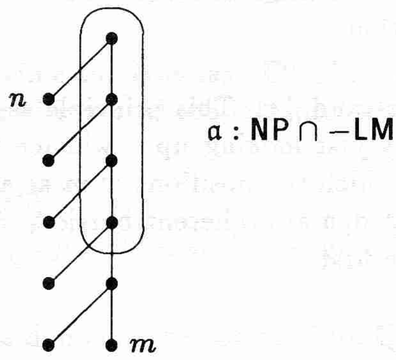
Segment here means 'element of the quasi-node' as defined earlier. Now given a labelled tree we first explicate (59). Barriers are quasi-nodes; but we want (59) to speak of nodes m, n and ultimately also reduce barriers to nodes rather than quasi-nodes. Consider then first the case where quasi-nodes are nodes. Then exclusion is non-dominance; (59) reduces to saying that β is n-subjacent to α if the $n+1^{st}$ barrier for β dominates α. That sounds like the original definition of subjacency; but it is not for the reason that barriers cannot be reduced to bounding nodes. To see what it in fact means we have to unpack (26). Consider a label BC true of blocking categories. Then (25) tells us that $\mathsf{BC} = -\mathsf{LM} \cap \mathsf{BAR}\colon 2$, where LM selects the L-marked nodes. Then, according to (26), the barriers for a node m are found as follows. Either (b) a node is a $\mathsf{BC} \cap -\mathsf{INFL}$, in which case it is a barrier for m; or (a) the node is the next BAR: 2-node above a blocking category strictly dominating m. The generating function for 0-SUB is therewith defined and yields the following effect:

(‡) $\quad$ $\text{0-SUB} = \mathsf{BC} \cap -\mathsf{INFL} \wedge (\mathsf{BC} \circ \mathsf{BAR}\colon 2)$.

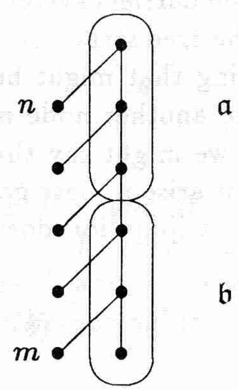
This concludes a strict definition of 0-subjacency; for the general case it is better to have some more notation. Let us denote by $\mathtt{i}$ the relation $\mathsf{BC} \cap -\mathsf{INFL}$ ("inherent barrier command") and by $\mathtt{u}$ the relation $\mathsf{BC} \circ \mathsf{BAR}\colon 2$ ("inherited barrier command"). Then as (‡) shows the first barrier is defined by $\mathtt{i} \wedge \mathtt{u}$, the second by $\mathtt{i} \circ \mathtt{i} \wedge \mathtt{i} \circ \mathtt{u} \wedge \mathtt{u} \circ \mathtt{i} \wedge \mathtt{u} \circ \mathtt{u}$, as can be checked. Thus n-subjacency in the new formulation is

$$\text{N-SUB} = \bigwedge \langle \sigma(0) \circ \sigma(1) \circ \ldots \circ \sigma(n) | \sigma(i) \in \{\mathtt{i}, \mathtt{u}\} \rangle$$

If we now come back to quasi-nodes we meet a rather unexpected difficulty with (17). Consider the case of m being 0-subjacent to n when the next barrier is inherent.



With the quasi-node $\mathfrak{a}$ as depicted we have that $\mathfrak{a}$ does not exclude $\mathfrak{n}$ the quasi-node of n but it does not include it either; and thus $\mathfrak{m}$ does not inherent barrier command $\mathfrak{n}$. Our definition (‡) fails to be adequate. Likewise the situation below characterizes a 0-subjacent $\mathfrak{m}$ for $\mathfrak{n}$ when both quasi-nodes $\mathfrak{a}, \mathfrak{b}$ are inherent barriers.

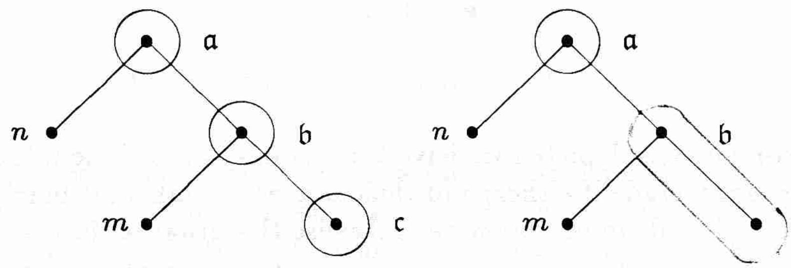


This does not follow from the above definitions but is clearly intended (in [Fa 87] the necessary adaptations are made to cover this case). It can be demonstrated that 0-subjacency is not $\mathbb{M}$-definable; for if it were, it would be defined by a $\mathfrak{d}$ with $\ell g(\mathfrak{d}) \leq \lambda$. Now consider the situation where $\mathfrak{a}$ has more than λ segments; then $\langle m, n\rangle \notin \overline{\ell}(\mathfrak{d})$. If that is so, there seems little that we can do but leave it for the linguists to decide the consequences of this.

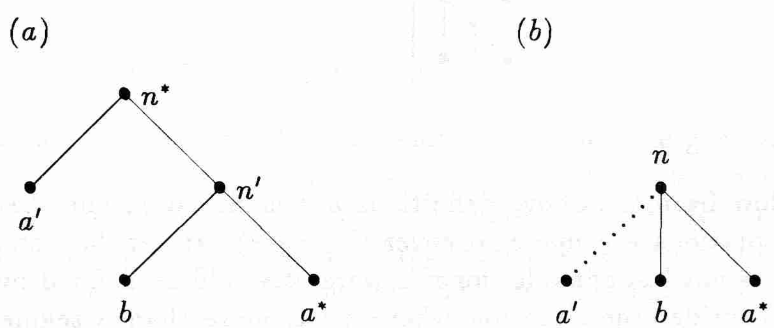
One might ask whether we just have not given ourselves enough power to define the relevant domains. Maybe there are some connectives that can achieve an internalization of barriers with quasi-nodes. But here we meet a rather strict dictum that tells us that domains in our sense are not able to define the domains of the barriers system no matter what connectives we grant ourselves.

[LOOK UPWARDS] The domain of a node in a tree depends only on the isomorphism type of its position.

By position we understand $\uparrow x$. This principle says no more than that a node can decide its domains by just looking up – whence the name. In the pictures below we have two trees in which the positions of m as are isomorphic as labelled objects. Yet if we assume that $\mathfrak{a}, \mathfrak{b}$ are inherent barriers, m is 0-subjacent to n only in the second tree not in the first.



An explanation for the fact that the barriers system is unanalyzable with our theory lies in the way Chomsky treats the tree structures formally. In the book, he is not using trees any more but something that might be called **vague trees** . In vague trees, a node m can either include another node n or exclude n or neither include nor exclude n. In the latter case we might say that m *vaguely includes* or *vaguely excludes* n. The third case does not arise in base generated structures but is created by adjunction; under this analysis adjunction does not yield the structure (a) but something like (b):



The dotted line denotes vague inclusion. The vague tree contains less information than (a) and thus one might wonder whether this is an adequate rendering of the structures of Barriers. But it seems that in this theory the relative C-command relation between two adjoined nodes does not play a role at all. For vague trees, however, the theory as presented here does not work at all since the relation of dependency between two node is assumed to be strictly bivalent.

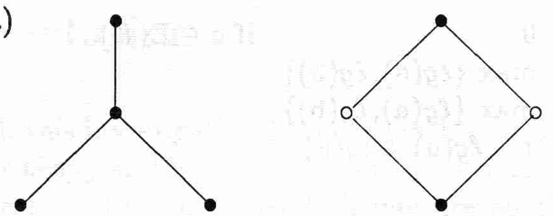
9.3 Constructing Domains with Composition

In a completely parallel way we define the language $\mathbb{D}^\circ(F)$, which has as the only extra over $\mathbb{D}(F)$ the relation composition $\circ$ whose interpretation is given by

$$\begin{aligned} \overline{\ell}(\mathfrak{a} \circ \mathfrak{b}) &= \overline{\ell}(\mathfrak{a}) \circ \overline{\ell}(\mathfrak{b}) \\ &= \{\langle x, y\rangle \mid (\exists z)(\langle x, z\rangle \in \overline{\ell}(\mathfrak{a}) \wedge \langle z, y\rangle \in \overline{\ell}(\mathfrak{b}))\}. \end{aligned}$$

Most results proved for $\mathbb{D}$-definable domains are not valid for $\mathbb{D}^\circ$-definable ones. To give just one example, there is no a priori bound for the number of $\mathbb{D}^\circ$-definable relations over a given finite set of features; nevertheless, the addition of composition does not at all mean that any relation can now be generated. Consider the tree (3a) of § 2.

(3a)



$\circ$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\top$	$\top$	$\top$

The only tight relations are $\top$ and $\bot$ and they combine under composition as given. We can see that they are closed under composition and so in this example the only $\mathbb{D}^\circ$-definable relations are $\top$ and $\bot$. Another, less trivial example is the fact that within $\mathbb{X}(\mathsf{S}, \mathsf{N})$ (the 'first branching' interpretation of) C-command is not $\mathbb{D}^\circ$-definable.

It is also not clear that our earlier results on decidability still go through for the simple reason that literal reduction does not work – which in turn is due to the fact that the associated function of a compositon of relations might not in all cases be the composition of the associated functions. Yet, if we restrict the discussion to MONOTONE relations, things are back to nomal.

Proposition 9.3.1 *Suppose that* R *and* S *are* MONOTONE. *Then* $R \circ S$ *is again* MONOTONE *and* $f_{R \circ S} = f_S \circ f_R$.

Proof. First let $\langle v, w\rangle \in R \circ S$. Then for some x, vRx and xSw; now if $v \leq v'$ then there are two cases. (a) $v' \leq x$. Then $v'Rx$ since R is MONOTONE; xSw gives $v'R \circ Sw$. (b) $x < v'$. Then $v'Rv'$ and $v'Sw$ since S is MONOTONE. And thus also $v'R \circ Sw$. To prove the second claim observe first that

(†) $f_{R\circ S}(x) = \max\{f_S(z) | z \le f_R(x)\}$.

This is so since $z \le f_R(x)$ is nothing but xRz and so $y \le \max\{f_S(z)|z \le f_R(x)\}$ means that there is a z such that $z \le f_R(x)$, i.e. xRz, and zSy. Since S is MONOTONE, $f_S(z)$ is maximal if z is maximal. But the greatest z we can chose is $z = f_R(x)$. Thus if S is MONOTONE (†) reduces to $f_{R\circ S}(x) = f_S \circ f_R(x)$. ■

Since it seems that syntactic domains are always MONOTONE we feel justified to single out a special sublanguage of $\mathbb{D}^{\mathrm{o}}(F)$ to create only MONOTONE domains. We let $\mathbb{M}(F) = \mathbb{D}^{\mathrm{o}}_{\circ,\wedge,\vee}(F)$ with all inside connectives retained.

On MONOTONE relations there is an equivalent of literal reduction. Call a CR **chain-tight** if it is a composition of TIGHT relations. A $\mathbb{M}$-definable relation is chain-tight if and only if it is $\mathbb{M}_\circ$-definable, that is, definable with only $\circ$ as outside symbol. That $\mathbb{M}$-definable relations are locally chain tight is what we are now going to prove. Define first the **length** of an $\mathbb{M}$-definable relation.

$$\begin{array}{lcll} \ell g(\mathfrak{a}) & = & 0 & \text{if } \mathfrak{a} \in \mathbb{D}(F) \\ \ell g(\mathfrak{a} \wedge \mathfrak{b}) & = & \max\{\ell g(\mathfrak{a}), \ell g(\mathfrak{b})\} & \\ \ell g(\mathfrak{a} \vee \mathfrak{b}) & = & \max\{\ell g(\mathfrak{a}), \ell g(\mathfrak{b})\} & \\ \ell g(\mathfrak{a} \circ \mathfrak{b}) & = & 1 + \ell g(\mathfrak{a}) + \ell g(\mathfrak{b}) & \end{array}$$

Theorem 9.3.2 *For every $\mathbb{M}$-definable $\mathfrak{d}$, every labelled tree $\langle \mathbb{T}, \ell\rangle$ and every $x \in T$ there is a chain-tight $\mathfrak{c}$ such that*

- $f_{\mathfrak{c}}(x) = f_{\mathfrak{d}}(x)$,
- $\ell g(\mathfrak{c}) \le \ell g(\mathfrak{d})$.

Proof. By induction on the construction of $\mathfrak{d}$. If $\mathfrak{d}$ is TIGHT, $\mathfrak{c} = \mathfrak{d}$ does the job. Now assume that $\mathfrak{d} = \mathfrak{d}_1 \wedge \mathfrak{d}_2$ or $\mathfrak{d} = \mathfrak{d}_1 \vee \mathfrak{d}_2$. Then by induction hypothesis there are $\mathfrak{c}_i$ such that $\ell g(\mathfrak{c}_i) \le \ell g(\mathfrak{d}_i)$ and $f_{\mathfrak{c}_i}(x) = f_{\mathfrak{d}_i}(x)$ and without loss of generality we can also assume that $f_{\mathfrak{c}_1}(x) \le f_{\mathfrak{c}_2}(x)$. Obviously $\mathfrak{c} = \mathfrak{c}_1$ is a good choice; and if $\mathfrak{d} = \mathfrak{d}_1 \vee \mathfrak{d}_2$ then $\mathfrak{c} = \mathfrak{c}_2$ is the right choice. Finally, let $\mathfrak{d} = \mathfrak{d}_1 \circ \mathfrak{d}_2$. There is by induction a $\mathfrak{c}_1$ with $\ell g(\mathfrak{c}_1) \le \ell g(\mathfrak{d}_1)$ and $f_{\mathfrak{c}_1}(x) = f_{\mathfrak{d}_1}(x) = y$. But there also is a $\mathfrak{c}_2$ with $\ell g(\mathfrak{c}_2) \le \ell g(\mathfrak{d}_2)$ and $f_{\mathfrak{c}_2}(y) = f_{\mathfrak{d}_2}(y)$. Putting this together we get that $\ell g(\mathfrak{c}_1 \circ \mathfrak{c}_2) \le \ell g(\mathfrak{d}_1 \circ \mathfrak{d}_2) = \ell g(\mathfrak{d})$ and that $f_{\mathfrak{c}_1 \circ \mathfrak{c}_2}(x) = f_{\mathfrak{c}_2}(f_{\mathfrak{c}_1}(x)) = f_{\mathfrak{c}_2}(f_{\mathfrak{d}_1}(x)) = f_{\mathfrak{c}_2}(y) = f_{\mathfrak{d}_2}(y) = f_{\mathfrak{d}_2}(f_{\mathfrak{d}_1}(x)) = f_{\mathfrak{d}}(x)$. Given that $\mathfrak{c}_1, \mathfrak{c}_2$ are both chain-tight so is $\mathfrak{c}_1 \circ \mathfrak{c}_2$. ■

This is a decisive property of $\mathbb{M}$-definable CRs that puts many of the earlier results back into our hands; for although we can define infinitely many CRs from a given finite F, there are only finitely many up to a given length λ. By induction it can be proved that any model for such a definable CR can be reduced to a model of size $2^{\lambda n}+2$, and so it is possible to introduce the notion of a λ-string and to test grammars for producing the same λ-strings; and deciding whether they are equivalent with respect to $\mathbb{M}$-definable CRs of length $\leq \lambda$. Although we believe that the study of intents of $\mathbb{M}$-definable CRs is a profitable subject for formal language studies (rather than intents of the $\mathbb{D}$-definable CRs) this is not an enterprise that we want to start here. But there is a small result that we want to add here about definability.

Proposition 9.3.3 $\mathbb{M}_{\wedge,\vee}(F) \subseteq \mathbb{M}_{\circ,\wedge}(F)$.

Proof. Recall that $\mathbb{M}_{\wedge,\vee}(F) = \mathbb{M}_{\vee}(F)$. We consider the simplest case where $\mathfrak{a}, \mathfrak{b}$ are both TIGHT and thus $\mathfrak{a} = \ulcorner [\![a]\!] \urcorner$, $\mathfrak{b} = \ulcorner [\![b]\!] \urcorner$. Let $c = a \cap b$. Then with $\mathfrak{c} = \ulcorner [\![c]\!] \urcorner$ we get $\mathfrak{a} \circ \mathfrak{b} = \mathfrak{a} \circ \mathfrak{b} \wedge \mathfrak{b} \circ \mathfrak{a} \wedge \mathfrak{c}$. ∎

10 Conclusion

Even though this investigation into the mathematical theory of domains has turned into a rather heavy and bulky reader with lots of solved problems, it still seems that the main work is still lying ahead. There are on the one hand pressing questions about further applications in linguistics and other disciplines and on the other there are some theoretical questions about the relation between grammars with explicit domain restrictions and others without such overt restrictions. We will turn to these questions shortly. First, let us do a critical evaluation of the work done so far.

A theoretical study such as the present one may serve two purposes. It may establish or isolate new concepts and develop a new terminology to name them and it may also yield insights into the theoretical possibilities and the consequences of basic assumptions. We hope to have contributed to both. It is of course at the beginning not easy to estimate the fruitfulness of the new concepts and to see which of the questions really deserve closer study. Only future and the investigation itself can tell. We might therefore be excused for the fact that not all theorems and considerations have a direct and foreseeable application in linguistics – at least we are unable to name such applications. Nevertheless, we feel that the questions we raised have an intrinsic motivation and therefore necessitate an attempt to answer them. At this point we shall repeat that the following problems are still unsolved.

QUESTION 1: Is every $\mathbb{I}$-definable TIGHT CR also $\mathbb{I}_{\emptyset}$-definable?

QUESTION 2: Is every $\mathbb{I}$-definable MONOTONE CR also $\mathbb{I}_{\vee}$-definable?

Similar problems arise with $\mathbb{M}$-definability, which anyway deserves a study of its own. The techniques of investigation will not be different but the study of extents of $\mathbb{M}(L)$-axioms will may well turn out to bear more fruit than the study of $\mathbb{D}(L)$-extents. In § 4.1 we have anyway stressed the point that the introduction of $\rightarrow$ will destroy an essential property of CRs namely MONOTONY.

In this essay we have applied the theory exclusively to GB; the only reason for this is that in GB domains play a rather central role and thus GB is a rather easy target for our demonstrations. But the theory of domains is itself neutral with respect to which syntactic theory we use. And that might be a great advantage. It is first of all possible to express linguistic knowledge about, say, reflexives or case assignment without being too precise about the details of the syntactic analysis. The minimum of structure that seems to be required is some kind of topology on the lexical items. For example, in an analysis of the following sentence

John eats pizza.

we can image up to $2^3 = 8$ sets of lexical items and hence (functional categories aside) up to 8 constituents. If the syntactic structure is always assumed to be a tree then not all of them can be real syntactic constituents. We will then only have these constituents (the syntactic labels are only approximations):

$[\text{John}]_{NP}$
$[\text{pizza}]_{NP}$
$[\text{eats}]_{V}$
$[\text{eats pizza}]_{VP}$
$[\text{John eats pizza}]_{S}$

But in some syntactic theories for example in Combinatorial Categorial Grammar there is no fixed sentence structure and there are many more sets than count as constituents in this grammar the price being that the syntactic labels turn out to be more exotic than in 'standard' theories. Be this as it may, some notion of a constituent being a set of lexical and/or functional items is enough to open the door for our theory of domains. We may in this way come closer to a real cross-theoretical study between grammatical theories that might otherwise be incomparable. To take an easy example, let us compare GB with GPSG. At first look, they seem to rather different as in GPSG there are no (or next to no) empty categories and GPSG is monostratal. Moreover, there is no overt correspondence between the nearness conditions in the modules and some particular part of GPSG. Nevertheless, *somewhere* in GPSG there must exist a mechanism that regulates the distribution of reflexives and the distribution of case features. The only thing that can do this is the feature percolation mechanism. Thus, we need to investigate the way in which nearness conditions on syntactic items is coded up into feature percolation in GPSG. This is indeed possible and paves the way for a systematic study of the correspondence between GB-type and GPSG-type grammars. The additional benefit will be that GB may profit from the rather rigorous way in which GPSG is defined and thus formal results about GPSG-type grammars might be transferrable into formal results about GB-type grammars. Conversely, we may be able to see how the rather high order knowledge about nearness can be mechanically coded into GPSG mechanisms and we are thus freed from the burden of actually writing these grammars.

Another point worth looking at is the particular ways in which domains get used in the various modules. This is by no means uniform; for example, indexation provides an ideal case where we only need MAX-command from the binder to the bound node. But case assignment, among other things, is less straightforwardly phrased. There we not only need a nearness condition one way from the case assigner to the case receiver; but also the other way around. This might have several explanations. We prefer to think that in addition to the requirement that the case assigner commands the case receiver there should be no other case assigner that takes precedence (in a sense to be defined). This definitions is rather similar to the definitions of [Rz 90] and akin to the normal conception of government. In our terminology, if case assignment is mediated through MAX-command (arguably) then the case assigner may only assign case to the corresponding MAX-*cell* – not the entire domain. This requires that each MAX-*chamber* contains at most one case assigner. This line of thinking applies also to θ-theory. It is rather different from the usual definition, where mutual MAX-command would be required (or indeed mutual IDC-COMMAND). In what ways these definitions differ in their consequences and which of them is to be preferred certainly deserves attention.

I conclude with a few remarks on other uses of this theory. In the introduction, it was already mentioned that discourse is also structured and that many phenomena such as availability of discourse referents are regimented by nearness conditions. But formal languages also make surprising use of domains. Consider the notion of a *bound variable*. Under a suitable analysis, an occurrence of x is **bound** if it is C-commanded by a quantifier expression quantifying over x. In addition, that quantifier expression binds this occurrence of x only if it is minimal with that property. (This again is some version of government.) We may now analyse this situation just as with case assignment either as a mutual command or as a relation between a head unique in its chamber and an x in a corresponding cell. Similar considerations apply to bound variables in programming languages when there is a possibility to have local variables such as in ALGOL.

A Glossary

B List of Symbols

References

[Ba 90] Barker, Chris and Pullum, Geoffrey K.: *A Theory of Command Relations*, Linguistics and Philosophy 13 (1990), 1 - 34

[vB 82] van Benthem, J. F. A. K.: *The logic of time*, Reidel, Dordrecht, 1982

[Bi 90] Bird, S., Klein, E.: *Phonological Events*, Journal of Linguistics 29(1990)

[Bl 91] Blackburn, Patrick: *Modal Logic for Attribute Value Structures*, in: Maarten de Rijke (ed.): Colloquium on Modal Logic 1991, Dutch Network for Language, Logic and Information

[Bu 81] Burris, S., Sankappanavar, H. P.: *A course in universal algebra*, Springer Verlag, New York, Heidelberg, Berlin, 1981

[Ch 81] Chomsky, Noam: *Lecture Notes on Government and Binding*, Foris, 1981

[Ch 86] Chomsky, Noam: *Barriers*, Linguistic Inquiry Monographs No. 13, MIT-Press, 1986

[Ch 90] Chomsky, Noam: *On Formalization and Formal Linguistics*, Natural Language and Linguistic Theory 8(1990), 143 - 147

[Ch 91] Chomsky, Noam: *Some Notes on Economy of Derivations and Representations*, in: [Fr 91]

[Co 88] Coopmans, Peter: *On Extraction from Adjuncts in VP*, Proceedings of WCCFL, 1988

[Da 91] Davey, B. A., Priestley, H. A.: *Introduction to Lattices and Order*, Cambridge Mathematical Textbooks, ^{2}1991

[Fa 87] Fanselow, Gisbert and Felix, Sascha W.: *Sprachtheorie, Bd. 2: Die Rektions- und Bindungstheorie*, UTB No. 1442, Tübingen, 1987

[Fr 91] Freidin, Robert(ed.): *Principles and Parameters in Comparative Grammar*, MIT-Press, 1991

[Ga 85] Gazdar, G., Klein, E. H., Pullum, G. K., Sag, I.: *Generalized Phrase Structure Grammar*, Blackwell, Oxford, 1985

[Ga 88] Gazdar, G., Pullum, G. K., Carpenter, R., Klein, E. H., Hukari, T. E., Levine, R. D.: *Category Structures*, Computational Linguistics 14-1(1988), 1- 14

[Gr 71] Grätzer, George: *Lattice Theory*, Freeman, San Francisco, 1971

[Gr 79] Grätzer, George: *Universal Algebra*, Springer Verlag, Berlin, Heidelberg, New York, 1979

[Hi 91] Hintikka, J., Sandu, G.: *On the methodology of linguistics: a case study*, Blackwell, Oxford, 1991

[Jo 82] Johnstone, Peter J.: *Stone Spaces*, Cambridge Studies in Advanced Mathematics 3, 1982

[dJ 88] de Jongh, D., Hendriks, L., Renardel de Lavalette, G. R.: *Computation in Fragments of intuitionistic propositional logic*, Logic Group Preprint Series No. 39, Dept. of Philosophy, Rijksuniversiteit Utrecht, 1988

[Ka 90] Kasper, R. T., Rounds, W. C.: *The Logic of Unification in Grammar*, Linguistics and Philosophy 13(1990), 35 - 58

[Kl 64] Klima, Edward: *Negation in English*, in J. Fodor and J. J. Katz (eds.): *The Structure of Language*, Prentice-Hall, Englewood Cliffs, 1964, 246 - 323

[Kr 89] Kracht, Marcus: *On the Logic of Category Definitions*, Computational Linguistics 15(1989), 111 - 113

[La 76] Lasnik, Howard: *Remarks on Coreference*, Linguistic Analysis 2(1976), 1 - 22

[Li 88] Lightfoot, D., Weinberg, A.: *Barriers (Review Article)*, Language 64(1988)

[Mc 87] McKenzie, R. N., McNulty, G. F., Taylor, W. F.: *Algebras, lattices and varieties*, vol. 1, Wadsworth and Brooks, Monterey, California, 1987

[Pu 89] Pullum, Geoffrey: *Formal Linguistics meets the Boojum*, Natural Language and Linguistic Theory 7(1989), 137 - 143

[Ra 79] Rautenberg, Wolfgang: *Klassische und nichtklassische Aussagenlogik*, Wiesbaden, 1979

[Re 86] Renardel de Lavalette, G. R.: *Interpolation in natural fragments of intuitionistic propositional logic*, Logic Group Preprint Series No. 15, Dept. of Philosophy, Rijksuniversiteit Utrecht, 1986

[Rn 81] Reinhart, Tanya: *Definite NP Anaphora and C-command Domains*, Linguistic Inquiry 12(1981), 605 - 635

[Rn 83] Reinhart, Tanya: *Coreference and Bound Anaphora*, Linguistics and Philosophy 6(1983), 47 - 88

[vR 86] van Riemsdijk, H., Williams, E.: *Introduction to the Theory of Grammar*, MIT-Press, Cambridge, Massachusetts, 1986

[Ri 90] Ristad, Sven Eric: *Computational Structure of GPSG Models*, Linguistics and Philosophy 13(1990), 521 - 587

[Rz 90] Rizzi, Luigi: *Relativised Minimality*, Linguistic Inquiry Monograph No. 16, MIT-Press, 1990

[Se 85] Sells, Peter: *Lectures on Contemporary Syntactic Theories*, CSLI Lecture Notes No. 3, 1985

[St 89] Stabler, Edward P. Jr.: *A Logical Approach to Syntax: Foundation, Specification and Implementations of Theories of Government and Binding*, manuscript, 1989

[Wi 90] Wille, Rudolf: *Concept Lattices and Conceptual Knowledge Systems*, Preprint No. 1340, FB Mathematik, TH Darmstadt, 1990

Logic Group Preprint Series

Department of Philosophy, University of Utrecht
Heidelberglaan 8, 3584 CS Utrecht
The Netherlands

1 C.P.J. Koymans, J.L.M. Vrancken, *Extending Process Algebra with the empty process*, September 1985
2 J.A. Bergstra, *A process creation mechanism in Process Algebra*, September 1985
3 J.A. Bergstra, *Put and get, primitives for synchronous unreliable message passing*, October 1985
4 A. Visser, *Evaluation, provably deductive equivalence in Heyting's arithmetic of substitution instances of propositional formulas*, November 1985
5 G.R. Renardel de Lavalette, *Interpolation in a fragment of intuitionistic propositional logic*, January 1986
6 C.P.J. Koymans, J.C. Mulder, *A modular approach to protocol verification using Process Algebra*, April 1986
7 D. van Dalen, F.J. de Vries, *Intuitionistic free abelian groups*, April 1986
8 F. Voorbraak, *A simplification of the completeness proofs for Guaspari and Solovay's R*, May 1986
9 H.B.M. Jonkers, C.P.J. Koymans & G.R. Renardel de Lavalette, *A semantic framework for the COLD-family of languages*, May 1986
10 G.R. Renardel de Lavalette, *Strictheidsanalyse*, May 1986
11 A. Visser, *Kunnen wij elke machine verslaan? Beschouwingen rondom Lucas' argument*, July 1986
12 E.C.W. Krabbe, *Naess's dichotomy of tenability and relevance*, June 1986
13 H. van Ditmarsch, *Abstractie in wiskunde, expertsystemen en argumentatie*, Augustus 1986
14 A. Visser, *Peano's Smart Children, a provability logical study of systems with built-in consistency*, October 1986
15 G.R. Renardel de Lavalette, *Interpolation in natural fragments of intuitionistic propositional logic*, October 1986
16 J.A. Bergstra, *Module Algebra for relational specifications*, November 1986
17 F.P.J.M. Voorbraak, *Tensed Intuitionistic Logic*, January 1987
18 J.A. Bergstra, J. Tiuryn, *Process Algebra semantics for queues*, January 1987
19 F.J. de Vries, *A functional program for the fast Fourier transform*, March 1987
20 A. Visser, *A course in bimodal provability logic*, May 1987
21 F.P.J.M. Voorbraak, *The logic of actual obligation, an alternative approach to deontic logic*, May 1987
22 E.C.W. Krabbe, *Creative reasoning in formal discussion*, June 1987
23 F.J. de Vries, *A functional program for Gaussian elimination*, September 1987
24 G.R. Renardel de Lavalette, *Interpolation in fragments of intuitionistic propositional logic*, October 1987 (revised version of no. 15)
25 F.J. de Vries, *Applications of constructive logic to sheaf constructions in toposes*, October 1987
26 F.P.J.M. Voorbraak, *Redeneren met onzekerheid in expertsystemen*, November 1987
27 P.H. Rodenburg, D.J. Hoekzema, *Specification of the fast Fourier transform algorithm as a term rewriting system*, December 1987
28 D. van Dalen, *The war of the frogs and the mice, or the crisis of the Mathematische Annalen*, December 1987
29 A. Visser, *Preliminary Notes on Interpretability Logic*, January 1988
30 D.J. Hoekzema, P.H. Rodenburg, *Gauß elimination as a term rewriting system*, January 1988
31 C. Smoryński, *Hilbert's Programme*, January 1988
32 G.R. Renardel de Lavalette, *Modularisation, Parameterisation, Interpolation*, January 1988
33 G.R. Renardel de Lavalette, *Strictness analysis for POLYREC, a language with polymorphic and recursive types*, March 1988
34 A. Visser, *A Descending Hierarchy of Reflection Principles*, April 1988
35 F.P.J.M. Voorbraak, *A computationally efficient approximation of Dempster-Shafer theory*, April 1988
36 C. Smoryński, *Arithmetic Analogues of McAloon's Unique Rosser Sentences*, April 1988

37 P.H. Rodenburg, F.J. van der Linden, *Manufacturing a cartesian closed category with exactly two objects*, May 1988

38 P.H. Rodenburg, J.L.M.Vrancken, *Parallel object-oriented term rewriting : The Booleans*, July 1988

39 D. de Jongh, L. Hendriks, G.R. Renardel de Lavalette, *Computations in fragments of intuitionistic propositional logic*, July 1988

40 A. Visser, *Interpretability Logic*, September 1988

41 M. Doorman, *The existence property in the presence of function symbols*, October 1988

42 F. Voorbraak, *On the justification of Dempster's rule of combination*, December 1988

43 A. Visser, *An inside view of EXP, or: The closed fragment of the provability logic of* $I\Delta_0+\Omega_1$, February 1989

44 D.H.J. de Jongh & A. Visser, *Explicit Fixed Points in Interpretability Logic*, March 1989

45 S. van Denneheuvel & G.R. Renardel de Lavalette, *Normalisation of database expressions involving calculations*, March 1989

46 M.F.J. Drossaers, *A Perceptron Network Theorem Prover for the Propositional Calculus*, July 1989

47 A. Visser, *The Formalization of Interpretability*, August 1989

48 J.L.M. Vrancken, *Parallel Object Oriented Term Rewriting : a first implementation in Pool2*, September 1989

49 G.R. Renardel de Lavalette, *Choice in applicative theories*, September 1989

50 C.P.J. Koymans & G.R. Renardel de Lavalette, *Inductive definitions in COLD-K*, September 1989

51 F. Voorbraak, *Conditionals, probability, and belief revision (preliminary version)*, October 1989

52 A. Visser, *On the* Σ^0_1*-Conservativity of* Σ^0_1*-Completeness*, October 1989

53 G.R. Renardel de Lavalette, *Counterexamples in applicative theories with choice*, January 1990

54 D. van Dalen, *L.E.J. Brouwer. Wiskundige en Mysticus*, June 1990

55 F. Voorbraak, *The logic of objective knowledge and rational belief*, September 1990

56 J.L.M. Vrancken, *Reflections on Parallel and Functional Languages*, September 1990

57 A. Visser, *An inside view of EXP, or: The closed fragment of the provability logic of* $I\Delta_0+\Omega_1$, revised version with new appendices, October 1990

58 S. van Denneheuvel, K. Kwast, G.R. Renardel de Lavalette, E. Spaan, *Query optimization using rewrite rules*, October 1990

59 G.R. Renardel de Lavalette, *Strictness analysis via abstract interpretation for recursively defined types*, October 1990

60 C.F.M. Vermeulen, *Sequence Semantics for Dynamic Predicate Logic*, January 1991

61 M.B. Kalsbeek, *Towards the Interpretability Logic of* $I\Delta_0+EXP$, January 1991.

62 D. van Dalen, $I<R$, *Some Intuitionistic Elementary Equivalences*, February 1991.

63 M. Bezem, *Strong termination of Logic Programs*, March 1991.

64 A. Visser, *The Unprovability of Small Inconsistency*, March 1991.

65 C.M. Jonker, *On the Semantics of Conflict Resolution in Truth Maintenance Systems*, March 1991.

66 F. Voorbraak, *A Preferential Model Semantics for Default Logic*, July 1991.

67 M. Kracht, *Splittings and the finite model property*, October 1991.

68 M. Bezem, *Impredicative recursion: terms depending on order types (extended abstract)*, November 1991.

69 E. Barendsen, M. Bezem, *Bar recursion versus polymorphism (extended abstract)*, December 1991.

70 C.F.M. Vermeulen, *Merging without Mystery*, December 1991.

71 F. Voorbraak, *Generalized Kripke models for epistemic logic*, December 1991.

72 W. Veldman, M. Bezem, *Ramsey's theorem and the pigeonhole principle in intuitionistic mathematics*, January 1992.

73 M. Kracht, *Prefinitely axiomatizable modal and intermediate logics*, January 1992.

74 C. M. Jonker, G. R. Renardel de Lavalette, *A tractable algorithm for the wellfounded model*, February 1992.

75 M. Kracht, *The Theory of Syntactic Domains*, February 1992.