

Localising Barriers Theory

Michael Schiehlen*

Institute for Computational Linguistics, University of Stuttgart,
Azenbergstr. 12, W-7000 Stuttgart 1

E-mail: mike@adler.ims.uni-stuttgart.de

1 Introduction

Government-Binding Parsing has become attractive in the last few years. A variety of systems have been designed in view of a correspondence as direct as possible with linguistic theory ([Johnson, 1989], [Pollard and Sag, 1991], [Kroch, 1989]). These approaches can be classified by their method of handling *global constraints*. Global constraints are syntactic in nature: They cover more than one projection. In contrast, local constraints can be checked inside a projection and, thus, lend themselves to a treatment in the lexicon. Conditions on features have been the subject of intensive study and viable logics have been proposed for them (see e.g. the CUF formalism [Dörre and Eisele, 1991], [Dorna, 1992]). In this paper, we assume such a unification-based mechanism to take care of local conditions and focus on global constraints. One class of approaches to principle-based parsing (see [Pollard and Sag, 1991] for HPSG, [Kroch, 1989] for TAG) attempts to reduce global conditions to local constraints and thus to make them accessible to treatment in a feature framework. This strategy has been pursued only at the expense of sacrificing the precise formulation of the theory and the definitory power stemming from it. The result has been a shift from the structural perspective assumed by GB theory to the object-oriented view taken by unification formalisms. The other class of approaches ([Johnson, 1989]) has allowed the full range of possible restrictions on trees and has incurred potential undecidability for its parsers. We take up a middle stance on the matter in that we propose a separate logic for global constraints and posit that global constraints only work on ancestor lines (see 7).

We assume “movement” to be encoded by the kind of gap-threading technique familiar from HPSG, LFG. In order to integrate global constraints a “state” (information that serves to express barrier configurations in the part of the tree which has already been built up) is associated with each “chain” (information about a moved element). Following HPSG, LFG, we have in mind a *rule-based* parser. Thus, states are manipulated when rules are chained. We need a calculus that is able to derive global constraints working on a local basis. We begin by developing this calculus hand in hand with an analysis of Chomsky’s frame-

*I wish to thank Robin Cooper, Mark Johnson and Esther König-Baumer for comments on earlier versions of this paper.

work. We then go on to show that many approaches to barriers theory and a variety of diverse phenomena can be moulded into our format and conclude with an indication of ways to use the system on-line during parsing.

2 Dependencies Between Nodes

We take a *tree* T to be a structure $(N, >)$, where N is a set of *nodes* and $>$ stands for *dominance*, a binary relation on N . We say that nodes a and b are *connected* iff $a > b \vee b > a \vee a = b$. We define the relation of immediate dominance \succ between two nodes a and b as $a > b \wedge \neg \exists c : a > c \wedge c > b$. Dominance is an irreflexive partial order relation satisfying the axioms (1–3). Ancestors of a node are connected (1), there exists a (single) root (2), dominance reduces to immediate dominance (3). Variables are universally quantified unless specified otherwise.

- (1) $x > z \wedge y > z \rightarrow x$ connected with y
- (2) $\exists x \forall y : x \geq y$
- (3) $x > z \rightarrow \exists y : x \succ y \wedge y \geq z$

Chomsky [1986, 9,30] discusses several definitions for constraints on unbounded *dependencies*.

- (13) α *c-commands* β iff α does not dominate β [and β does not dominate or equal α] and every γ that dominates α dominates β .

Where γ is restricted to maximal projections we will say that α *m-commands* β .

- (18) α *governs* β iff α *m-commands* β and there is no γ , γ a barrier for β , such that γ excludes α .

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

All of these can be moulded into the general format introduced in (4): Two nodes can only stand in a relation R if they are unconnected and, furthermore, at most n barriers for the second node do not dominate the first one. The notion of a barrier B remains to be specified. For now, we only demand that barrierhood entail dominance. We call relations that satisfy axiom (4) *definable with barrier concepts*, for short BC-definable.

$$(4) \ aRb \leftrightarrow a, b \text{ unconnected} \wedge |\{c \mid B(c, b) \wedge \neg c > a\}| \leq n$$

Balanced relations like government require a definition in terms of two BC-definable relations: $R_1(a, b)$ and $R_2(b, a)$.

$$(5) \ B(c, b) \rightarrow c > b$$

We can show several properties of BC-definable relations. The nodes are unconnected.

$$(6) \ aRb \rightarrow a, b \text{ unconnected}$$

In order to investigate BC-definable relations it suffices to investigate the *ancestor lines* of their second argument b (that is $\{y \mid y \geq b\}$).

$$(7) \ x > y \wedge x > a_1 \wedge \neg y \geq a_1 \wedge x > a_2 \wedge \neg y \geq a_2 \wedge y \geq b \rightarrow (a_1 Rb \leftrightarrow a_2 Rb)$$

(7) gives rise to equivalence classes for the first argument of R . For a particular pair (a, b) we can always find a y as defined in (8).

$$(8) \ a \in [y]_b \leftrightarrow x > y \wedge x > a \wedge \neg y \geq a \wedge y \geq b$$

Definable relations are never empty. Barriers are preserved in the upward direction of the ancestor line:

$$(9) \ [y] R y$$

$$(10) \ x > y \wedge \neg [y] R b \rightarrow \neg [x] R b$$

(10) is less innocent than it looks. I give a revealing binding example from Kamp and Reyle [1993].

If $[_{CP=x} [_{CP=y} \text{he}_i \text{ sees Mary }] \text{ and she smiles}] \text{John}_i \text{ is happy.}$

$*[_{CP=x} [_{CP=y} \text{He}_i \text{ sees Mary }] \text{ and John}_i \text{ is happy}].$

3 Barrier Definitions

3.1 Adjunction

Adjunction rules raise a problem for algebraic investigations of barriers theory (e.g. [Kracht, 1992]): They insert material into a tree but do not create new projections. Thus, adjunction rules imply a distinction between *projections* and *segment nodes* that correspond to graph-theoretical nodes. We shall use Greek letters to refer to projection nodes and Latin letters for segment nodes. The only way to create projections covering more than one segment is through adjunction. Since adjunction rules have equivalent mother and daughter nodes, projections are coherent in the sense that:

$$\forall \alpha \notin \beta \forall b_1, b_2 \in \beta : a > b_1 \rightarrow a > b_2$$

Chomsky [1986] defines projection dominance so that α *dominates* β only if every segment of α dominates (every segment of) β . In case this definition is not empty, (1) guarantees a unique minimal segment a_{min} of α . Thus, we can rephrase Chomsky's definition in terms of segment nodes and get that α dominates β just in case the minimal segment of α dominates some segment of β .

$$(11) \ \text{dominate}(\alpha, \beta) \leftrightarrow a \in \alpha \wedge b \in \beta \wedge \text{minimal segment}(a) \wedge a > b$$

Likewise, Chomsky's definition of exclusion, viz that α *excludes* β if no segment of α dominates (any segment of) β , can be transformed to the equivalent condition that α excludes β if the maximal segment of α does not dominate a segment of β .

$$(12) \ \text{exclude}(\alpha, \beta) \leftrightarrow a \in \alpha \wedge b \in \beta \wedge \text{maximal segment}(a) \wedge \neg a > b$$

This way, we reduce projection dominance to segment dominance. In (13–15), conditions of segment minimality or maximality are included where they are appropriate by (11) and (12).

3.2 Chomsky's Theory

Chomsky [1986, 14] gives the following two core definitions for barriers. We are not concerned about the exact formulation of L-marking (for a definition see [Chomsky, 1986, 24]).

$$(25) \ \gamma \text{ is a blocking category for } \beta \text{ iff}$$

γ is not L-marked and γ dominates β .

$$(26) \ \gamma \text{ is a barrier for } \beta \text{ iff (a) or (b):}$$

- a. γ immediately dominates δ ,
 δ a blocking category for β ;
- b. γ is a blocking category for β , $\gamma \neq \text{IP}$.

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).

Formulation of these definitions in first order logic yields (13–15). In order to obtain an open-ended definition scheme the equivalence of the above definitions is held implicit: Barrier concepts are true iff they comply with a manifest definition (see also 22 and 23).

$$(13) \ \text{blocking category}(c, b) \leftarrow \text{maximal projection}(c) \wedge$$

- \neg L-marked(c) \wedge
 minimal segment(c) \wedge
 $c > b$.
 (14) barrier(c, b) \Leftarrow
 maximal projection(c) \wedge
 minimal segment(c) \wedge
 $\exists d$: blocking category(d, b) \wedge
 $c > d \wedge$
 $\forall e : c > e > d \rightarrow$
 \neg (maximal projection(e) \wedge
 minimal segment(e)).
 (15) barrier(c, b) \Leftarrow
 blocking category(c, b) \wedge
 \neg IP(c).

We regard unary predicates as local conditions (L) and binary predicates as global concepts (B for “barrier concept”). Abstracting over the particular predicates involved we end up with the following definition schemes (16 for 13 and 15, 17 for 14).

- (16) B(c, b) \Leftarrow
 L(c) \wedge
 $c > b$.
 (17) B(c, b) \Leftarrow
 L(c) \wedge
 $\exists d : B(d, b) \wedge$
 $c > d \wedge$
 $\forall e : c > e > d \rightarrow \neg L(e)$.

We call the existential subformula of (17) an *inheritance clause* I. The only global conditions in our system are inheritance clauses and $c > b$, a condition that always holds for barrier concepts (see 5). We will discuss in detail a way to derive inheritance clauses on a rule to rule basis. For the sake of conciseness we adopt the following abbreviation for inheritance clauses.

$$\begin{aligned} & \exists \delta : B(d, b) \wedge c > d \wedge \forall e : c > e > d \rightarrow \neg L(e) \\ & \iff \\ & I(c, b, B, L) \end{aligned}$$

3.3 Negative Inheritance Clauses

It has interesting repercussions to incorporate a scheme with a negated inheritance clause, viz. (18).

- (18) B(c, b) \Leftarrow
 L(c) \wedge
 $c > b \wedge$
 $\neg \exists d : B(d, b) \wedge$
 $c > d \wedge$
 $\forall e : c > e > d \rightarrow \neg L(e)$.

For illustration we discuss several applications for negative inheritance clauses.

Chomsky [1986, 37] talks about IPs as inherent barriers, this effect being restricted to the *most deeply embedded* tensed IP. To capture this concept we once again need a negative inheritance clause: An IP is most deeply embedded if it does not dominate any other IP.

- (20) barrier(γ, β) \Leftarrow
 tensed IP(γ) \wedge
 $\gamma > \beta \wedge$
 $\neg \exists \delta : IP(\delta, \beta) \wedge$
 $\gamma > \delta$.
 IP(γ, β) \Leftarrow IP(γ) \wedge $\gamma > \beta$.

A feature of negative inheritance clauses that is desirable in many cases is that they allow to cancel barriers higher up in the tree. They can be used to circumvent (24). Classical GB theory has had to resort to a variety of tricks to account for discontinuous domains. A case in point is the coherent infinitive construction found in German and Dutch¹. A standard account is to reanalyse ϑ -structure into another structure that lacks the annoying barrier-generating nodes. Different submodules of the theory will work on different structures. Consider the following example.

daß [_{CP} [_{IP} PRO [_{VP} [_{NP} der Wagen] zu reparieren]]] [_V versucht] wurde

In this example V governs NP but not “PRO” even though “PRO” intervenes between V and NP. CP might be called a *phantom* barrier. Generally, a phantom (like CP, IP above) is a barrier just in case it does not dominate a non-phantom (VP above). Thus CP shields “PRO” but remains open for government of NP. This state of affairs can be caught in the present framework by a negative inheritance clause.

- (21) barrier(γ, β) \Leftarrow
 phantom(γ) \wedge
 $\gamma > \beta \wedge$
 $\neg \exists \delta : \text{nonphantom}(\delta, \beta) \wedge$
 $\gamma > \delta$.
 nonphantom(γ, β) \Leftarrow nonphantom(γ) \wedge $\gamma > \beta$.

Similar cases arise with negation. Again, the literature adopts different lines of argument to account for the phenomenon. Kamp and Reyle [1993] handle the binding case below with a rule of double negation elimination, an operation that deletes structure.

*Either he_i owns a Porsche or John_i hides it.

Either he_i does not own a Porsche or John_i hides it.

¹Müller and Sternefeld [1991] propose to treat this construction within the framework of barrier theory.

The examples below are drawn from Cinque [1990, 83]. He uses a superscription convention to annotate the scope of the negation and assumes an LF amalgamation process triggered by coindexing of this sort. CP is no barrier anymore for LF-amalgamated elements since they become wh-movable. We might model amalgamation with the “nonphantom” clause of (21). Then, this clause would have to hold true for inherently wh-movable elements (bare quantifiers in Cinque’s analysis) as well.

*Molti amici, [_{CP} ha invitato t, che io sappia.

Molti amici, [_{CP} [_{NegP} non ha invitato t, che io sappia.

3.4 Properties of the Definition Schemes

In this paragraph we further investigate properties of the three definition schemes we are dealing with. We summarize scheme (16) in (22). *def* is a variable ranging over the given definitions.

$$(22) B(c, b) \leftrightarrow \exists def: L_{def}(c) \wedge c > b$$

We can collapse all definitions *def* into a single definition with local condition $K(c) \leftrightarrow \bigvee_{def} L_{def}(c)$. In order to summarize the schemes (16–17) we introduce vectors of definitions \vec{def} of length *n* and corresponding sequences of nodes \vec{x} of length *n* + 1. x_1 is fixed to *c* and x_{n+1} to *b*.

$$(23) B(c, b) \leftrightarrow \exists \vec{def}, \vec{x}: \forall i \in \{1, \dots, n\}: L_{def(i)}(x_i) \wedge x_i > x_{i+1}.$$

For definitions conforming to type (16–17) we can show the following property: If we have found a son *y* violating the relation *R* all descendants *b* of the father *x* will be inaccessible to *R*.

$$(24) x \succ y \wedge aRx \wedge \neg aRy \wedge x > b \rightarrow \neg aRb$$

In a full-fledged definition scheme where (16–18) are available (24) ceases to hold. In the example discussed above *a* does not govern *y* but does govern *b*.

$$a \text{ [_{CP=}_x [_{VP=}_y b$$

In pre-Barriers GB theory and most current computational approaches only inherent barriers are allowed (scheme 16) and the violating number of barriers in axiom (4) is set to null. Note that under these provisos, barriers theory shrinks to command theory:

$$(4') aRb \leftrightarrow a, b \text{ unconnected} \wedge \forall c: K(c) \wedge c > b \rightarrow c > a$$

The following constraint holds in this configuration: A barrier as in (24) is not affected by the triggering first argument.

$$(25) x \succ y \wedge \exists a: [aRx \wedge \neg aRy] \wedge bRx \rightarrow \neg bRy$$

Chomsky [1986, 11] discusses (25) at some length. In his example (see below) “decide”=*a* does not govern “PRO”, but “e”=*b* would. He shows that if either of the mentioned requirements (*n*=0 and intrinsic barriers) is not met the theorem is refuted.

$$(21) \text{John decided [_{CP} e [_{IP} PRO to [_{VP} see the movie]]]$$

If (16–18) are given then we can show the following theorem: Brothers are equivalent when occurring as a second argument of a BC-definable relation.

$$(26) a, b_1 \text{ unconnected} \wedge a, b_2 \text{ unconnected} \wedge b_f \succ b_1 \wedge b_f \succ b_2 \rightarrow (aRb_1 \leftrightarrow aRb_2)$$

4 Localising the Global Constraints

The next step is to localise the definitions (16–18). For ease of reference we repeat the definition schemes.

$$(27) B(c, b) \leftrightarrow \exists def: [L_1(c) \wedge c > b] \vee [L_1(c) \wedge I(c, b, B, L_2)] \vee [L_1(c) \wedge c > b \wedge \neg I(c, b, B, L_2)]$$

We only take into account nodes *c* that separate *a* from *b* in the sense that they sit on the ancestor line of *b* but not on that of *a* (see also the restrictions of 4 and 5). Theorem (28) specifies a connection between the inheritance clauses valid on a father node *x* and those valid on the son *y*. Recall that inheritance clauses are the only global conditions we consider.

$$(28) x \succ y \wedge y \geq b \wedge \neg y \geq a \rightarrow (B(y, b) \vee (I(y, b, B, L) \wedge \neg L(y)) \leftrightarrow I(x, b, B, L))$$

In parsing, an unbounded dependency (formally, a relation *R*) is triggered by a node n_1 (e.g. because it lacks a ϑ -role or cannot take up a ϑ -role assigned to it) and successfully terminates when a corresponding node n_2 is found (that can supply the missing ϑ -role or absorb a superfluous ϑ -role). When searching, ancestor lines are either ascended or descended. Accordingly we have to make a distinction between the upward and downward state of dependency information.

4.1 Upward States

Upward states supply information about barrier nodes encountered on the ancestor line below. They are constructed when the second argument b of a relation R has been found and the tree is being searched for the first argument a . Formally, upward states are sets (standing for conjunctions) associated with some node c and some dependency coming from b .

$$\langle B, L \rangle \in \text{UState}(c, b) \iff I(c, b, B, L)$$

Any inheritance clause that can be derived at c on the basis of the lower upward state and the rule schemes (27–28) is included in c 's upward state. If a clause is not in the state, it cannot be inferred by (16–18). Consequently, the negation of a missing clause must hold. We assume a counter for c and b to be increased and checked as defined by the theory (computing the number n of passed barriers, see 4).

$$\text{IncreaseCounter}(c, b) \leftrightarrow B(c, b)$$

We use the upward state to break off search as soon as we can infer from the theory that an element a cannot possibly be found in the rest of the tree. Theorem (29) stands to express that as soon as we have found a node y violating the definitions upward search becomes obsolete.

$$(29) \neg[y]Rb \wedge \neg y > a \rightarrow \neg aRb$$

4.2 Downward States

Downward states encode information about barrier nodes encountered on the ancestor line above. They are computed when the second argument b of a relation R is being expected because a first argument a has been discovered. Formally, downward states are first order formulae associated with some node c , some ancestor node c_1 of c , and some dependency leading to b . Atomic formulae of $\text{DState}(c, c_1, b)$ are inheritance clauses I with respect to c and b .

$$\begin{aligned} &\text{formula} \in \text{DState}(c, c_1, b) \\ &\iff \\ &\text{formula}(c, b) \leftrightarrow \text{IncreaseCounter}(c_1, b) \end{aligned}$$

The rule schemes (27–28) supply all sufficient and necessary conditions for transfer of inheritance clauses between nodes. Accordingly an atomic formula in the upper downward state can be transformed into a formula holding for the lower node c . False formulae are discarded, while true formulae increase the counter.

We use downward states to restrict the search space. By (24) we can sometimes infer that search into a subtree will be pointless. Negative inheritance

clauses, however, can only be checked when a candidate for b has been encountered. When the parser descends paths while searching, it always assumes that the current path will dominate b . For upward states, in contrast, the ancestor line of b is fixed. Only downward states scan trees. (26) shows that a state will not change for brother nodes. So we only have to store one downward state per rule (e.g. under its mother node).

4.3 Example

Consider the chain of “how” in the following example

how do [IP^* you [VP^* t [VP remember
[CP t/*why [IP Bill t behaved t]]]]]]

In a left-to-right top-down parse, the first barrier to be encountered would be IP^* if it dominated either a blocking category (BC) or no other tensed IP. VP^* is no BC or barrier since it does not dominate the intermediate trace (it is not the minimal segment of the VP node). CP is L-marked and hence a barrier only if it dominates a BC. If “why” excludes a trace in SpecCP, the BC IP occurs between CP and the next trace. Due to the ϑ -role of “how”, government is violated leading to an ungrammatical sentence. If an intermediate trace is allowed, a new chain is started and no BC occurs. IP refutes the hypothesis that IP^* is the deepest embedded tensed IP, and it turns out to be this IP as soon as the variable is found. So only one subjacency barrier occurs: The sentence is grammatical.

5 Conclusion

We have described a mechanism that handles global constraints on long movement from a local basis. The device has been derived from a logical formulation of Chomsky's [1986] theory so that equivalence to this theory is easily proved. We have sketched methods to use the logic for early determination of ungrammatical readings in a parser. In my thesis ([Schiehlen, 1992]) the technique has been implemented in an Earley parser that generates all readings in parallel. In this system local conditions are couched into feature terms. Feature clashes lead to creation and abolition of dependencies modelling the GB notion of failed feature assignment and last resource. The barriers logic restricts rule choice for the predictor (descending ancestor lines) and discards analyses in the completer (ascending ancestor lines). Ongoing work is centred around an application of the barriers framework to the generation of semantic structure (Discourse Representation Structure). Kracht's [1992] approach to analysing barriers theory is related to the one presented here. However, Kracht's emphasis is not so much on parsing.

A Proofs

Proof of (6) is trivial.

The theorem (7) is symmetric for a_1 and a_2 . Suppose $a_1 R b \wedge \neg a_2 R b$. a_2 and b are unconnected. So there exist k_1 barriers not dominating a_1 ($k_1 \leq n$) and k_2 barriers not dominating a_2 ($k_2 > n$). Suppose c is a barrier not dominating a_2 but dominating a_1 (there are at least $k_2 - k_1 \geq 1$ such barriers). $c > b$ and $y \geq b$, hence c and y are connected. But $y \geq c$ entails $y > a_1$. If $c > y$ then either $x > c > y$ or $c > x$. But $c > x$ implies $c > a_2$.

To prove (9) note that all barriers for y dominate y by (5). Hence they also dominate $a \in [y]$.

We now turn to (10). Take $a_1 \in [x]$ and $a_2 \in [y]$. a_2 and y are not connected. We show that if $\neg c > a_2$ and $c > b$ then $\neg c > a_1$. Assume $c > b$ and $c > a_1$. Then x and c are connected both dominating b . We know that $\neg x \geq c > a_1$. Hence $c > x > y$. Suppose y_f is y 's father. Then $c > x \geq y_f \succ y$ and equally $c > x \geq y_f \succ a_2$. We obtain that $\{c \mid B(c, b) \wedge \neg c > a_1\} \supseteq \{c \mid B(c, b) \wedge \neg c > a_2\}$. Hence $\neg[x]Rb$.

We prove (24). Suppose c is a barrier for x . Then by (23) there is a sequence of nodes $x_1 = c$ and $x_n > x_{n+1} = x$. But $x_n > x > b$, so c is a barrier for b as well. a and y are unconnected. Suppose c is a barrier for y but not x . Then $x_1 = c$ and $x_n > x_{n+1} = y$. x_n and x are connected both dominating y . We know that $\neg x > x_n > y$ and $\neg x_n > x$ else c would be a barrier for x . Hence $x_n = x$ and we get $x_n = x > b$. There are at least as many barriers for b as there are for y , so $\neg a R b$.

To prove (25) we adopt the argumentation of the foregoing proof and infer that x is a barrier for y . $b R x$ shows that b , x are unconnected, hence $\neg x > b$ and $\neg b R y$.

(26) follows if we prove $B(c, b_1) \leftrightarrow B(c, b_2)$ by induction. The theorem is symmetric. Assume a c such that $B(c, b_1)$. Then either scheme (16) holds: $L(c) \wedge c > b_1$ hence $c > b_2$. Or (17) and $L(c) \wedge \exists d : B(d, b_1) \wedge c > d \wedge \forall e : c > e > d \rightarrow \neg L(e)$. By induction $B(d, b_2)$ as well. For the negative scheme (18) we use symmetry to extend the implication $I(c, b_1, B, L) \rightarrow I(c, b_2, B, L)$ to an equivalence.

For (28) we give a proof by cases. Either $B(y, b) \rightarrow I(x, b, B, L)$. y is the barrier node d referred to in the consequent. Or $I(y, b, B, L) \wedge \neg L(y) \rightarrow I(x, b, B, L)$. We set the barrier node d of the first inheritance clause equal to the one of the second. Does a node e between x and d satisfy L ? y does not, nor do the nodes between y and d , and there is no node between x and y . But y and e must be connected, both dominating d . We show $I(x, b, B, L) \rightarrow B(y, b) \vee I(y, b, B, L)$. The barrier node d of the antecedent clause and y are connected, both dominating b (see

5). d cannot sit between x and y . If $d = y$ the first disjunct holds. If $y > d$ we set d equal to the barrier node of the second disjunct. No e between y and d satisfies L .

We reduce (29) to (10). If $a \geq y > b$ we make use of (6). Otherwise let x_f be the smallest node that dominates both y and a and let x be such that $x_f \succ x \geq y$. Then by (10) $\neg[x]Rb$, meaning $\neg a R b$ (see 8).

References

- [Chomsky, 1986] Noam Chomsky. *Barriers*. Linguistic Inquiry Monograph 13, MIT Press, Cambridge, Massachusetts, 1986.
- [Cinque, 1990] Guglielmo Cinque. *Types of \bar{A} -Dependencies*. Linguistic Inquiry Monograph 17, MIT Press, Cambridge, Massachusetts, 1990.
- [Dörre and Eisele, 1991] Jochen Dörre and Andreas Eisele. *A Comprehensive Unification-Based Grammar Formalism*. Deliverable R3.1.B, DYANA — ESPRIT Basic Research Action BR3175, 1991.
- [Dorna, 1992] Michael Dorna. *Erweiterung der Constraint-Logiksprache CUF um ein Typsystem*. Diplomarbeit Nr. 896, Institut für Informatik, Universität Stuttgart, 1992.
- [Johnson, 1989] Mark Johnson. *The Use of Knowledge of Language*. In *Journal of Psycholinguistic Research*, 18(1), 1989.
- [Kamp and Reyle, 1993] Hans Kamp and Uwe Reyle. *From Discourse to Logic, Vol I*. to appear: Kluwer, Dordrecht, 1993.
- [Kracht, 1992] Marcus Kracht. *The Theory of Syntactic Domains*. Logic Group Preprint Series No. 75, Department of Philosophy, University of Utrecht, February 1992.
- [Kroch, 1989] Anthony S. Kroch. *Asymmetries in Long-Distance Extraction in a Tree-Adjoining Grammar*. In Mark Baltin and Anthony Kroch, eds. *Alternative Conceptions of Phrase Structure*. University of Chicago Press, Chicago, 1989.
- [Müller and Sternefeld, 1991] Gereon Müller and Wolfgang Sternefeld. *Extraction, Lexical Variation, and the Theory of Barriers*. Universität Konstanz, September 1991.
- [Pollard and Sag, 1991] Carl Pollard and Ivan A. Sag. *Agreement, Binding and Control*. draft, June 1991.
- [Rizzi, 1990] Luigi Rizzi. *Relativized Minimality*. Linguistic Inquiry Monograph 16, MIT Press, Cambridge, Massachusetts, 1990.
- [Schiehlen, 1992] Michael Schiehlen. *GB-Parsing am Beispiel der Barrierentheorie* Studienarbeit Nr. 1168, Institut für Informatik, Universität Stuttgart, 1992.