Head-Driven Phrase Structure Grammar

A natural language is described by

- universal principles
- language-specific principles
- the lexicon of the language
- the rules of the language

Principles of universal grammar: $P_1, \ldots, P_m$

Head Feature Principle, Subcategorization Principle, Semantics Principle, …

Principles of German: $P_{m+1}, \ldots, P_n$

Constituent-Order Principle, …

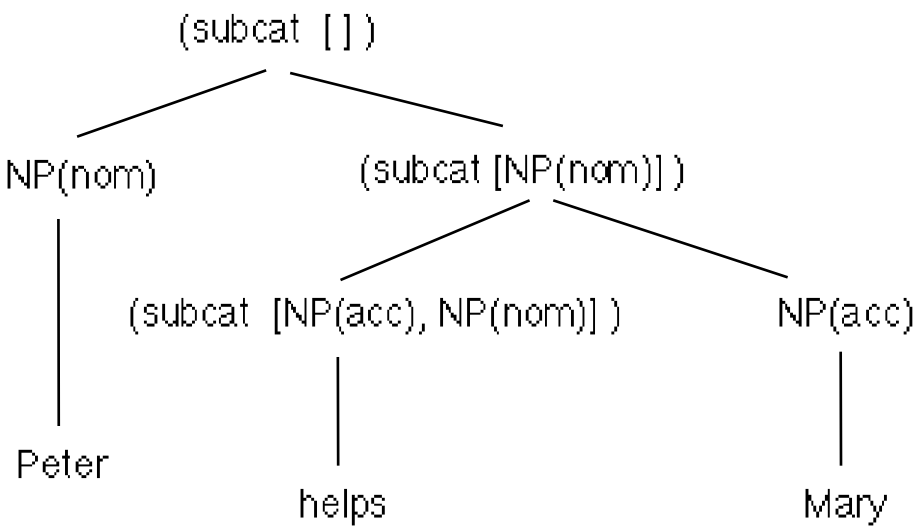
Lexicon of German: $L_1, \ldots, L_p$

Rules of German: $R_1, \ldots, R_q$

German = $P_1 \wedge \ldots \wedge P_n \wedge (L_1 \vee \ldots \vee L_p \vee R_1 \vee \ldots \vee R_q)$

Pollard/Sag 1987

Subcategorization in HPSG



Rule: (subcat Rest) → (subcat [Comp|Rest]) , Comp

Lexicon: (subcat [NP(acc), NP(nom)]) → helps

NP → Peter

NP → Mary

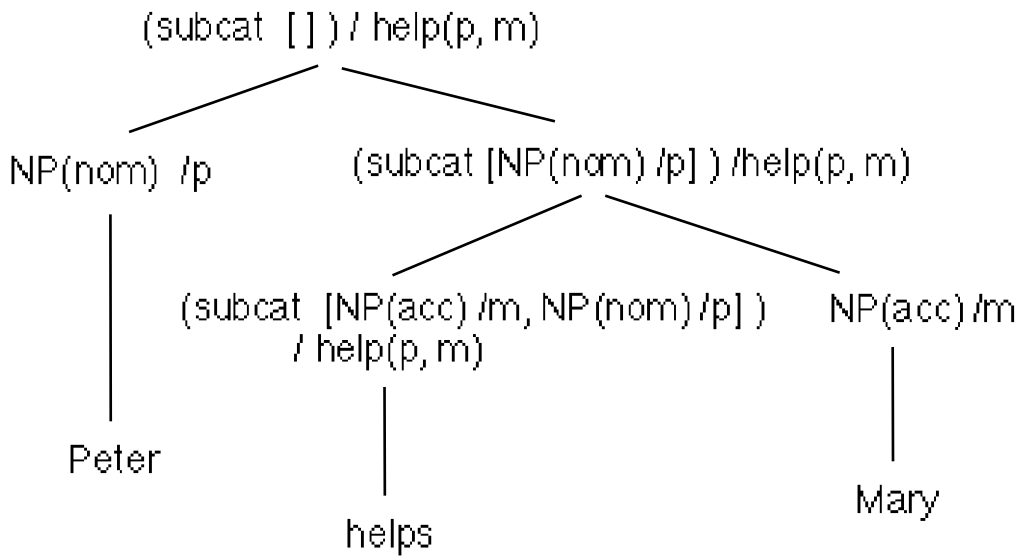
Compositional Semantics

- Tradition of Richard Montague
- (almost) every constituente has a semantics

  The semantics of a constituent is composed of the semantic of its daughter constituents

  ⇒ **Principle of composition**
- Use of different theories of semantics possible (extensions to FOPL are common; ⇒ **logical form**)

Integration of Syntax and Semantics



Rule (format in Definite Clause Grammar):

`<Syntax> / <Semantik>`

(subcat [Rest]) /Sem → (subcat [Comp|Rest]) /Sem , Comp

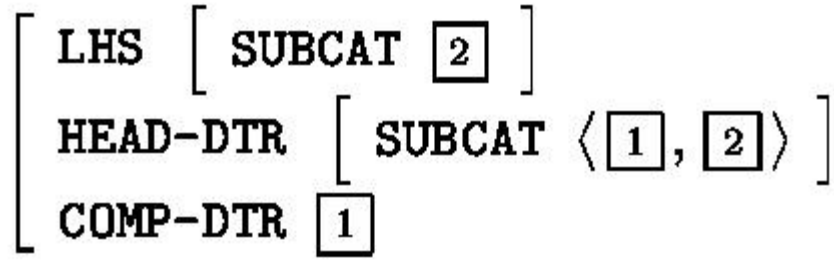
Lexikon:

(subcat [NP(acc) /Obj, NP(nom) /Subj]) /help(Subj, Obj) → helps

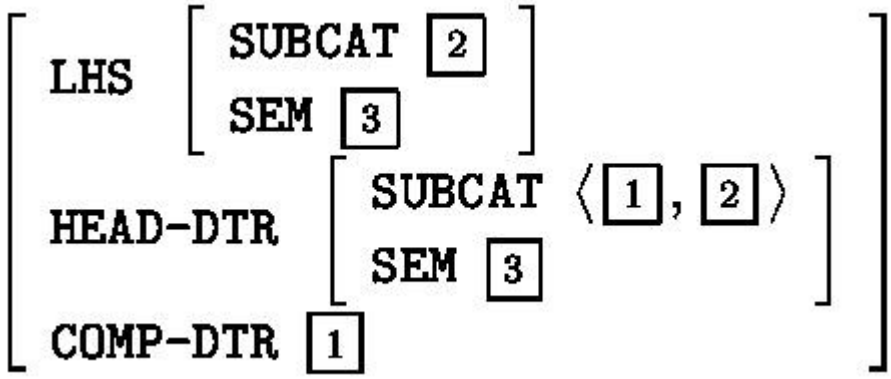
NP/p → Peter

NP/m → Mary

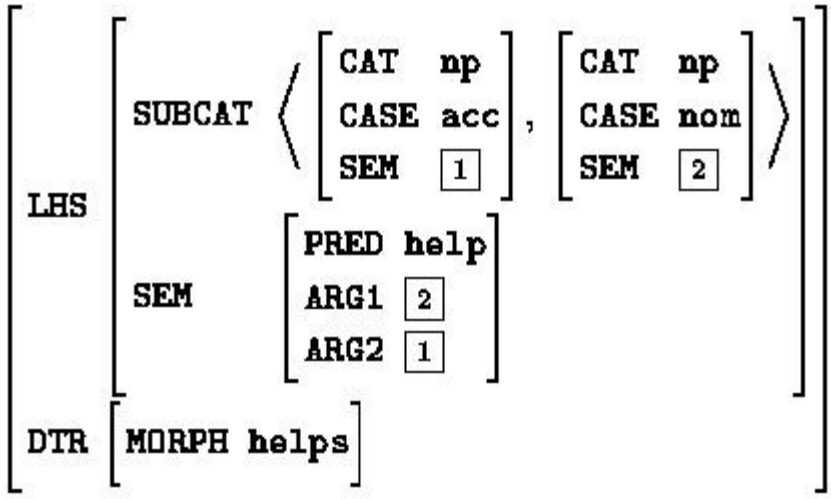
Feature Structures in Matrix Form



(subcat REST) → (subcat [COMP | REST]), COMP

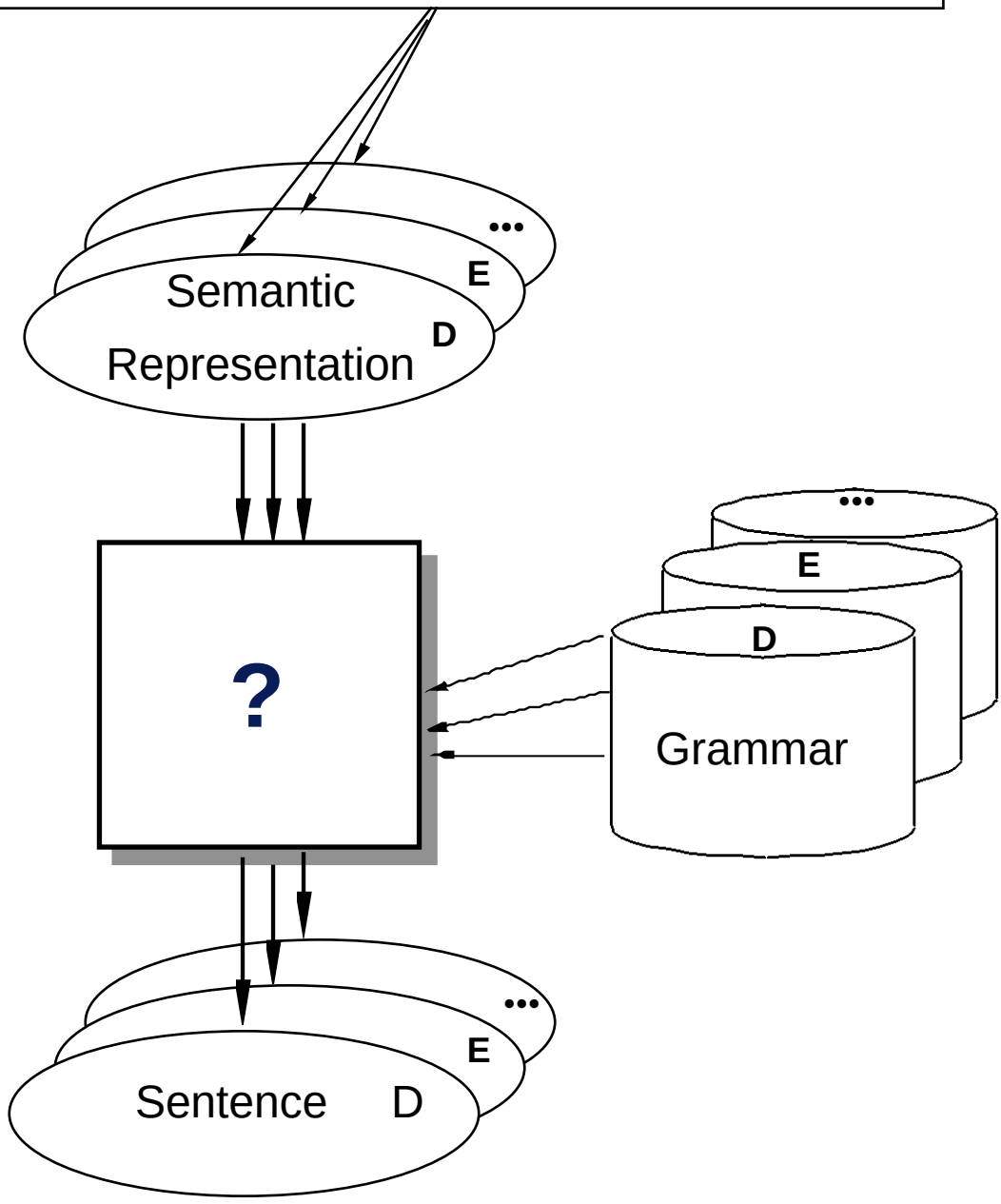


(subcat REST) / SEM → (subcat [COMP | REST]) / SEM, COMP



(subcat [NP(acc) / OBJ, NP(nom) / SUBJ]
/ help(SUBJ, OBJ)) → [helps]

- Natural Language Dialogue Systems
- Systems for Machine Translation
- Text generation systems



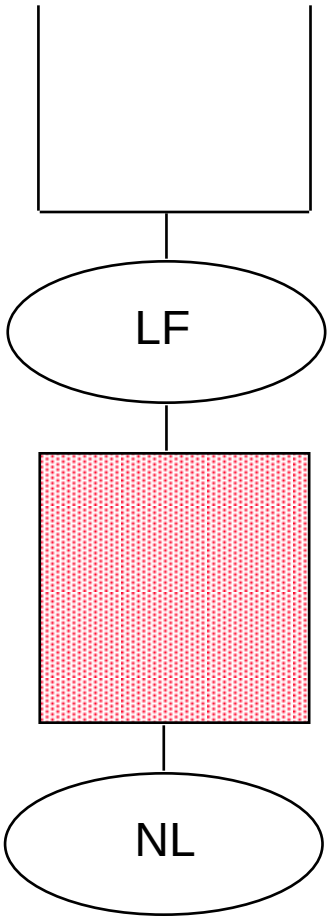
? = Efficient, multilingual generation with reversible grammars

Reversibility

- **Reversible grammars**

  Parser and Generator use the same grammar

- **Reversible processes**



- Decision-making

- deduction-based

  Shieber 1988, Neumann 1998

- type instantiation

  Emele/Zajac 1990

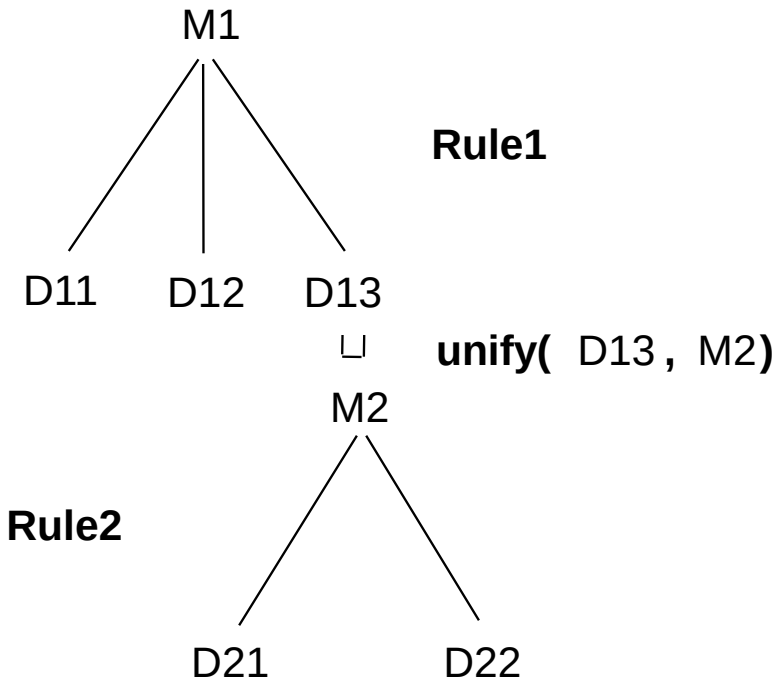
The Generation Task

- **Given a semantic representation and a grammar; which are the terminal strings admissible by the grammar?**

- Processing of syntax and formal semantics are at the focus of research
- Input is the semantic representation (logical form) of a sentence
- This representation is language-specific, i.e. it differs between e.g. German and English

$\Rightarrow$ Task is **word order** and **word inflection**

Processing Steps for Unification



- **Bottom-Up Step**
  - 1. current node M2
  - 2. unification with D13
  - 3. new current node from {D11, D12}
- **Top-Down Step**
  - 1. current node D13
  - 2. unification with M2
  - 3. new current node from {D21, D22}

Top-Down Generation
Using Structural Subcategorization

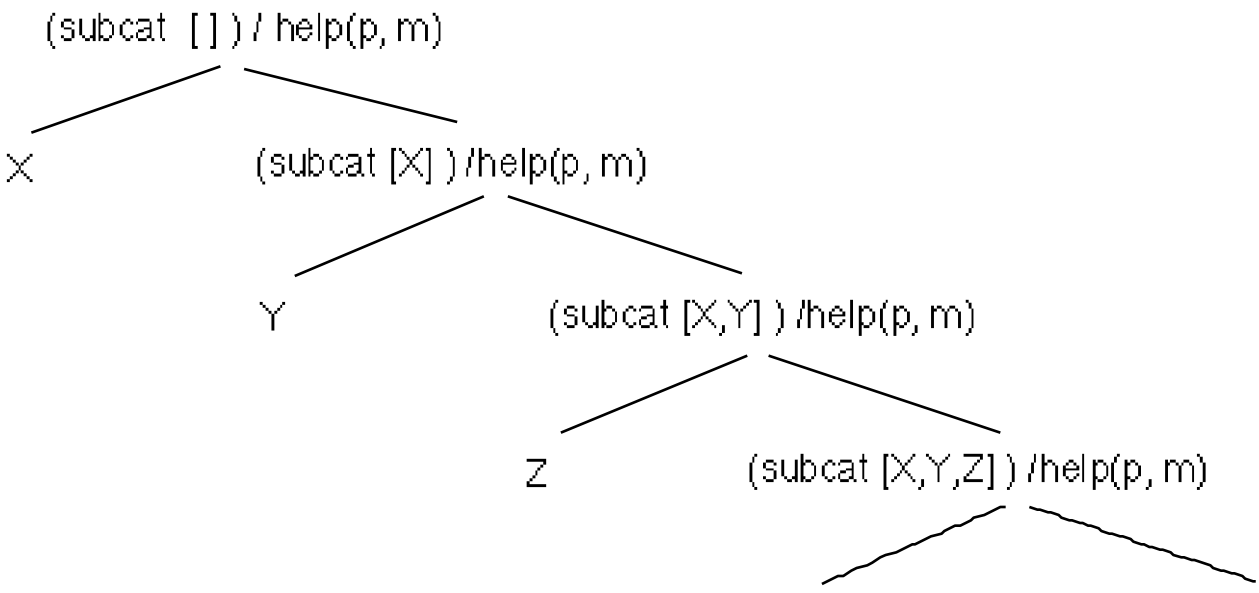
Input Structure:

(subcat [ ]) /help(p, m)

Rule Application:

(subcat [ ]) /help(p, m) → (subcat [X]) /help(p, m), X

The same rule can be applied again to the result; the procedure does not terminate:



[Van Noord 1990]

Semantic-Head-Driven Generation: Pivot Nodes

- **Idea**: Strategy utilizing known information during generation, i.e. the semantics
- **Pivot**: deepest node in the derivation tree, such that this node and all nodes higher up in the path possess the same semantics

  $\Rightarrow$ *semantic head of the root node*
- **Generation** of derivation tree both top-down and bottom-up, starting from the respective pivots

  $\Rightarrow$ *semantic-head-driven*

Shieber et al. 1990

SHDG algorithm
(following Shieber et al. 1990)

```
generate(Root) :-
   applicable_non_chain_rule(Root, Pivot, RHS),
   generate_rhs(RHS),
   connect(Pivot, Root).

generate_rhs([]).
generate_rhs([First|Rest]) :-
      generate(First),
      generate_rhs(Rest).

connect(Pivot, Root) :-
      applicable_chain_rule(Pivot, LHS, Root, RHS),
      generate_rhs(RHS),
      connect(LHS, Root).
connect(Pivot, Root) :-
      unify(Pivot, Root).

applicable_non_chain_rule(Root, Pivot, RHS) :-
      node_semantics(Root, Sem),
      node_semantics(Pivot, Sem),
      non_chain_rule(LHS, RHS),
      unify(Pivot, LHS).

applicable_chain_rule(Pivot, Parent, Root, RHS) :-
      chain_rule(Parent, RHS, SemHead),
      unify(Pivot, SemHead).
```

Semantic-Head-Driven Generation: Generating the Derivation Tree

Chain Rules form a subset of the grammar rules:

The semantics of one right-hand side element is identical to the semantics of the left-hand side

$\Rightarrow$ *semantic head of the rule*

Top-Down:

- Expansion of pivot using a Non-Chain Rule
- The daughters of a NCR are processed recursively
- Termination: There are no (more) nonterminal daughters, i.e. the NKR is a lexicon entry

Bottom-Up:

- Reduction of pivot towards the root using a Chain Rule
- The non-semantic-head daughters of a CR are processed recursively
- Termination: the pivot and the root nodes unify

Semantic-Head-Driven Generation: DCG sample grammar

sentence/decl(S) → s(finite) /S

s(Form)/S → Subj, vp(Form, [Subj]) /S

vp(Form, Subcat)/S → vp(Form, [Compl|Subcat]) /S, Compl

vp(finite, [np(_)/O, p/up, np(3-sing) /S]) /call_up(S,O) → [calls]

np(3-sing)/john → [john]

np(3-pl) /friends → [friends]

p/up → [up]

Input structure to SHDG procedure:

decl(call_up(john, friends))

Example (1)

Pivot:

sentence
/ decl(call_up(john,friends))

NCR

s(Form) / call_up(john,friends)

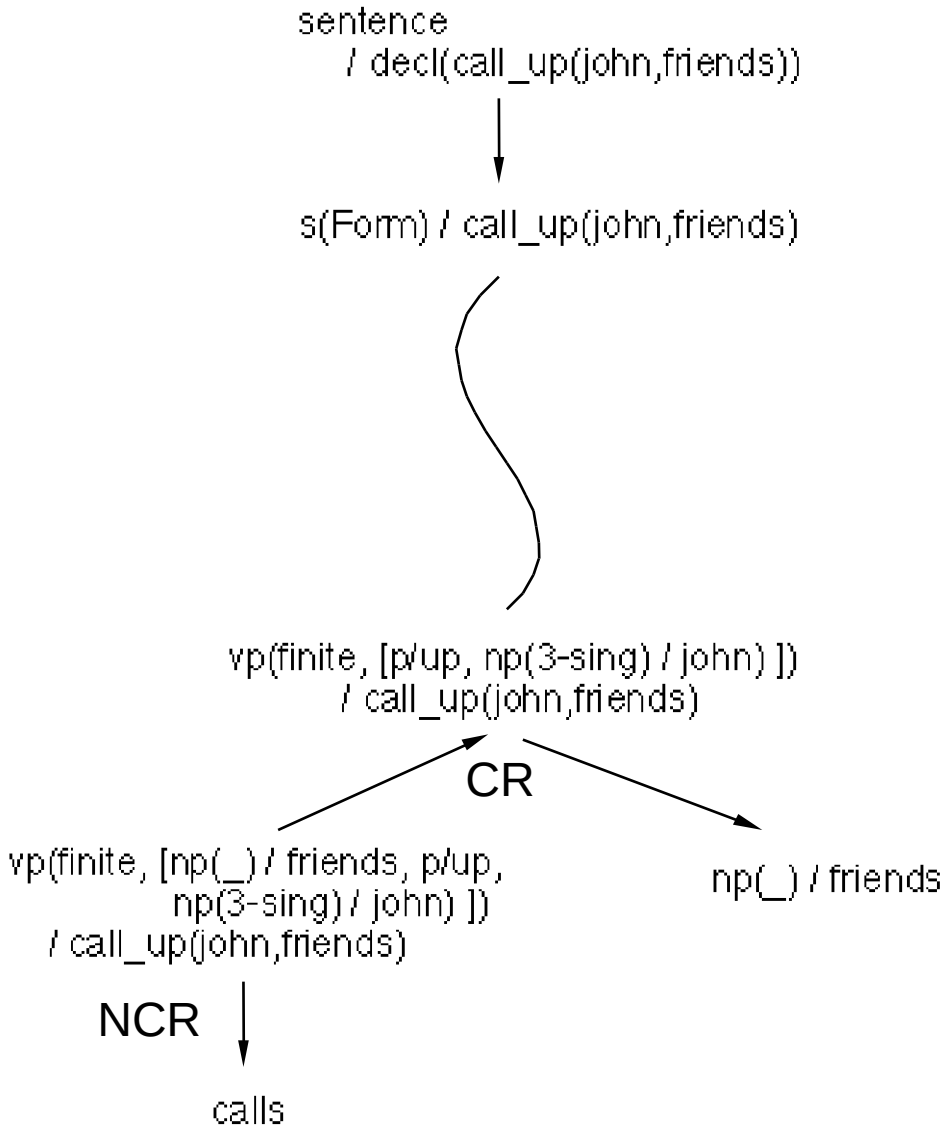
Pivot:

vp(finite, [np(_) / friends, p/up, np(3-sing) / john])
/ call_up(john,friends)

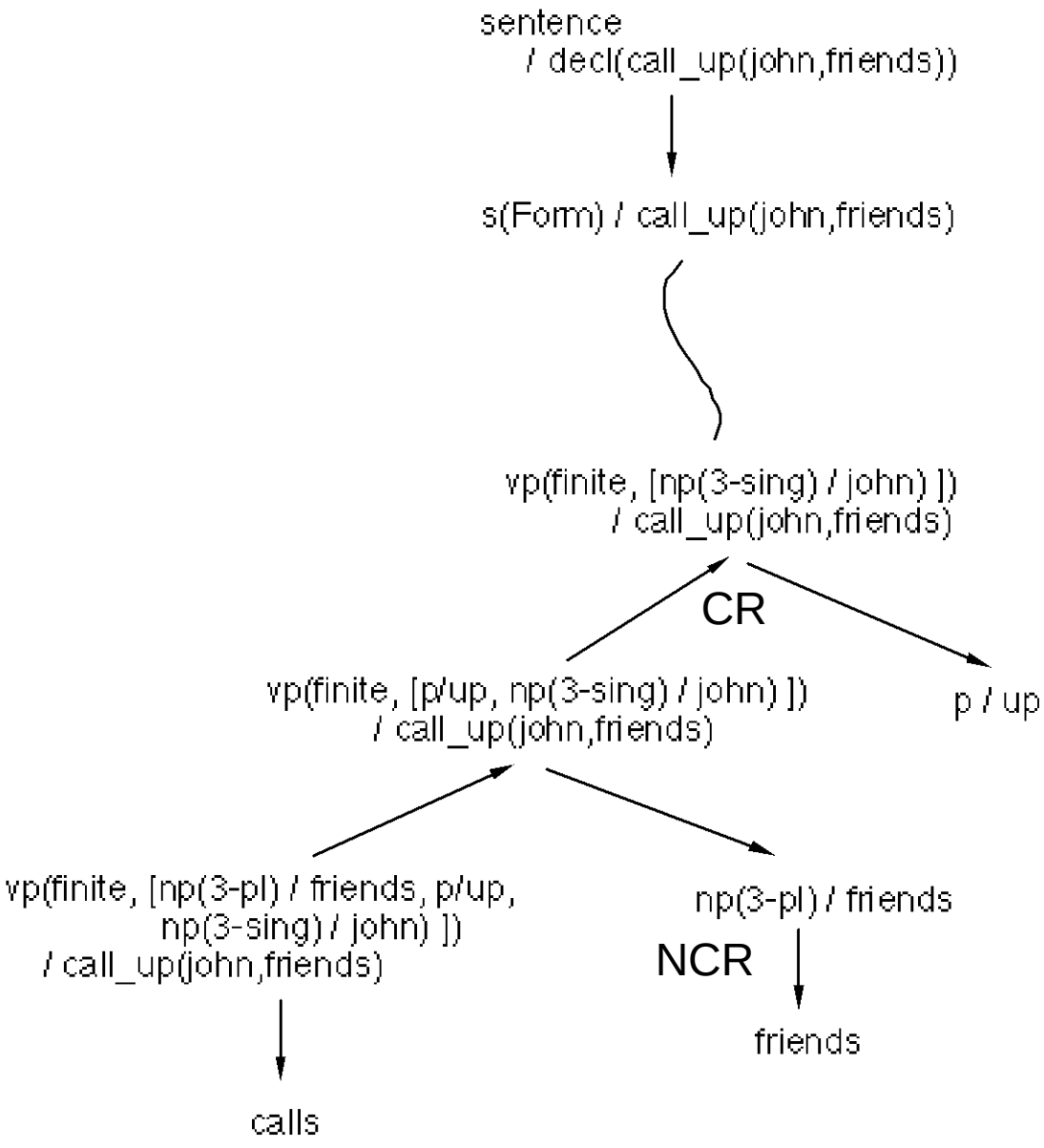
NCR

calls

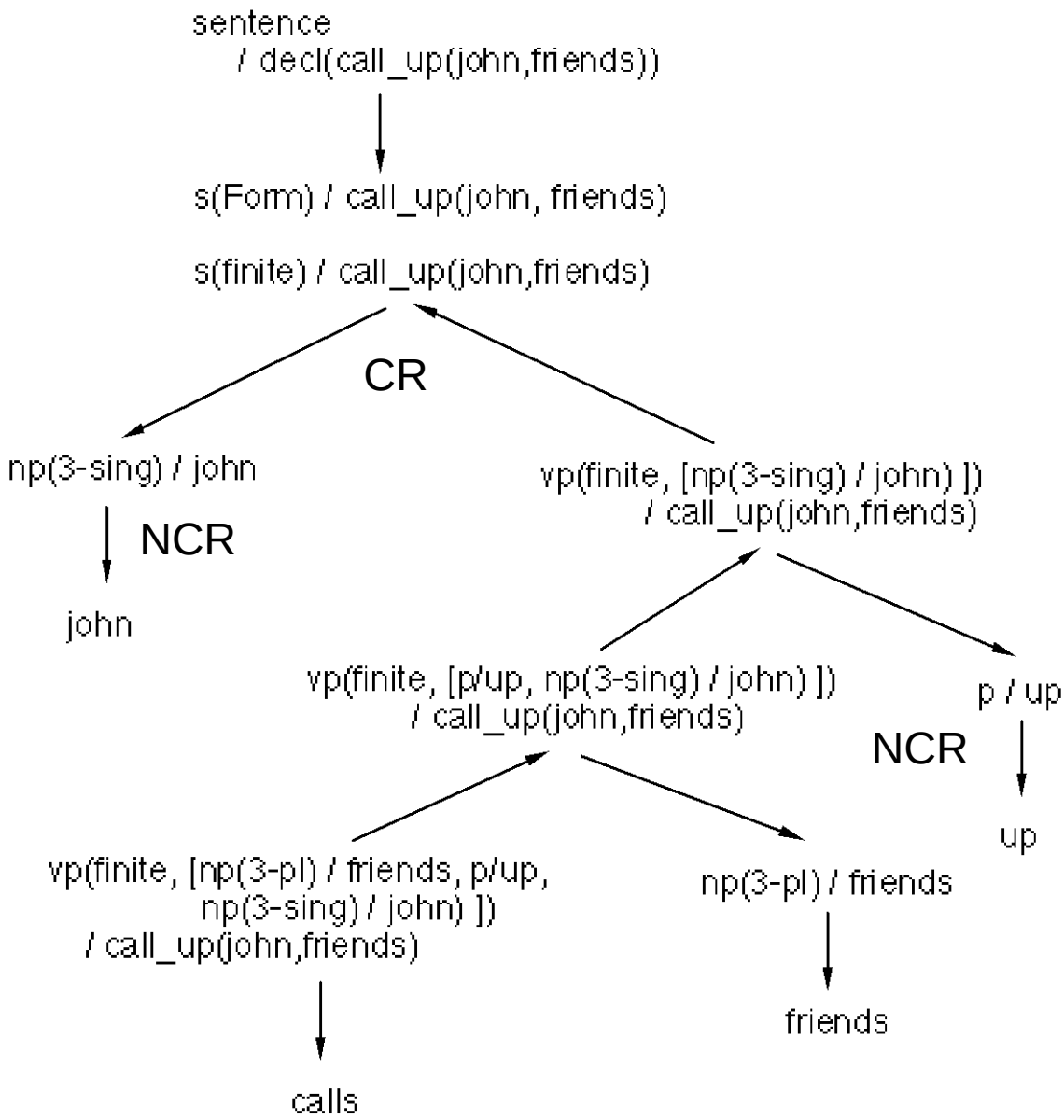
Example (2)



Example (3)



Example (4)



Logical Form (LF)

- There is no agreement on
  - which information *must* be represented in a LF
  - whether a LF must correspond to *exactly one* terminal string

- Grammars cannot derive unique LF from sentences; thus "ambiguous" *quasi-logical forms*
- Disambiguation (of anaphoric relations and scopus ambiguities) during parsing through subsequent analysis steps
- for generation, a unique LF can be assumed
- Equivalence problem for LF:

  p and q $\Leftrightarrow$ q and p ??

  John fell and lost his conscience.

  p $\Leftrightarrow$ p and (q or (not q)) ??

  John lost his conscience, and Bruce Springsteen is the boss, or Bruce Springsteen is not the Boss.

- Open research problem: What is "intended equivalence"?

Appelt 1987

Alshawi/van Eick 1989