Parsing logic grammars 2

- Phrase Structure Grammars vs Logic Grammars
  - terminal and non terminal vocabularies
    - Tags belonging to a close tagset
    - Open expressions
      - terms from a 1st order predicate logic (e.g. Prolog terms)
      - Feature Structures, FS
      - ...

Parsing logic grammars 3

- Logic Grammars vs Unification Grammars
  - How important is unification
- Adaptation of classical algorithms dealing with CFG

Parsing logic grammars 4

Unification importance

Tomabechi,1991

85% - 90% of parsing time

- 75% - 95% from this time: structures copying
- > 60% of unifications in correct parses fail

- Other operations
  - subsuntion, generalization, inheritance, disjuntion, negation, reescripture, assignment

Prolog as parser 1

- Pros
  - Parsing as theorem proving
  - A specific parser is not needed
  - Incorporated to formalism: DCG
  - (Relatively) easy to write the grammar
  - (Relatively) easy to integrate

Prolog as parser 2

- Cons
  - TD fixed strategy
  - Fixed left to right management of literals
  - Fixed order of clauses (grammar rules)
  - Problems with left recursivity
  - Backtracking
    - Computation redundancy
    - Generation of useless (sub) parse trees
  - Lack of flexibility on parsing

Prolog as parser 3

- Improvements
  - (Stabler, 1990)
  - Separation of grammar clauses from lexicon ones.
  - Reordering of objetives to satisfy (literals).
    - static and dynamic (when trying to satisfy right part of the clause)
  - Unfolding.
    - Replacing part of the literals of the right part of a clause by the unfolding of the involved clauses.
  - Partial evaluation.

Prolog as parser 4

- Improvements
  - Generation and test.
    - performing tests asap within a conjuntion.
  - Using selectional functions domain specific
  - Literal reordering in the grammar using statistical techniques taking into account the number of arguments, their complexity, their instantiation level, etc...
  - dynamic pruning (e.g. limitations on the number of recursive steps)
  - Saving of intermediate results (possitive or negative)

Prolog as parser 5

- Improvements
  - Kiefer et al, 1999
  - Precompiling the lexicon
  - expansion and aplication of lexical rules
  - removing parts of FS useless for parsing
  - Improvements in unification
    - quasi-destructive algorithm of Tomabechi
    - reusing parts of input structure in the output (Pereira)
  - Precompilation of type unification
  - Precompilation of filtering rules

Prolog as parser 6

- Improvements
  - unification dynamic filters (“quick check”)
  - Reduction of the size of FS using restrictors
  - Limitation of the initial number of items of chart
  - Computing and saving best partial analysis

Prolog as parser 7

- Bottom Up Parser
  - Traduction, at compilation time, of grammar rules, in DCG, to a type of Horn clauses, BUP clauses, that will be further processed by Prolog.
  - The resulting parser incorporates in BUP clauses the control mechanism ("left corner")
  - [Matsumoto et al, 1983], [Matsumoto et al, 1985]
  - SAX (Matsumoto, Sagimura)