Syntactic Formalisms

I. Context-Free Grammars

- Definition

  $G = (N, T, P, S)$

  where:

  - N is a finite set of non-terminal symbols;
  - T is a finite set of terminal symbols;
  - P is a finite set of production rules

    $$A \to \alpha \text{ where } A \in N \text{ and } \alpha \in (N \cup T)^*$$

  - $S \in N$ is the start symbol.

  $L(G) = \{\omega \in T^* \mid S \to^* \omega\}$

Example

- Grammatical rules

  S → NP VP
  VP → tV NP
  VP → stV Adj
  NP → Det N

- Lexicon

  tV → /mange/
  stV → /est/
  NP → /Pierre/
  N → /fruit/
  Adj → /intelligent/
  Det → /un/

CKY Algorithm

init

$$<A \rightarrow \bullet\alpha, i, i>$$

scan

$$\frac{<A \rightarrow \alpha \bullet a\beta, i, j>}{<A \rightarrow \alpha a \bullet \beta, i, j+1>} \quad a = a_{j+1}$$

complete

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j> \quad <B \rightarrow \gamma\bullet, j, k>}{<A \rightarrow \alpha B \bullet \beta, i, k>}$$

- Good news/Bad news

Earley Algorithm

init

$$<S \rightarrow \bullet\alpha, 0, 0>$$

scan

$$\frac{<A \rightarrow \alpha \bullet a\beta, i, j>}{<A \rightarrow \alpha a \bullet \beta, i, j+1>} \quad a = a_{j+1}$$

complete

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j> \quad <B \rightarrow \gamma\bullet, j, k>}{<A \rightarrow \alpha B \bullet \beta, i, k>}$$

predict

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j>}{<B \rightarrow \bullet\gamma, j, j>}$$

- Correct-prefix property

II. Unification Grammars

Suppose we extend our toy grammar with the following rules:

NP → NP Conj NP

Conj → /et/
NP → /Marie/
N → /pomme/
Det → /une/
Det → /des/

We get:

*Marie est intelligent

*Marie mange un pomme

*Pierre et Marie mange une pomme

?Pierre mange Marie

$$
\begin{array}{lcl}
\text{S} & \rightarrow & \text{NP}\,[X,Y]\ \ \text{VP}\,[X,Y] \\
\text{VP}\,[X,Y] & \rightarrow & \text{tV}\,[Y]\ \ \text{NP}\,[W,Z] \\
\text{VP}\,[X,Y] & \rightarrow & \text{stV}\,[Y]\ \ \text{Adj}\,[X,Y] \\
\text{NP}\,[X,Y] & \rightarrow & \text{Det}\,[X,Y]\ \ \text{N}\,[X,Y] \\
\text{NP}\,[\text{m},\text{p}] & \rightarrow & \text{NP}\,[\text{m},X]\ \ \text{Conj}\ \ \text{NP}\,[Y,Z] \\
\text{NP}\,[\text{m},\text{p}] & \rightarrow & \text{NP}\,[X,Y]\ \ \text{Conj}\ \ \text{NP}\,[\text{m},Z] \\
\text{NP}\,[\text{f},\text{p}] & \rightarrow & \text{NP}\,[\text{f},X]\ \ \text{Conj}\ \ \text{NP}\,[\text{f},Y]
\end{array}
$$

$$
\begin{array}{lcl}
\text{tV}\,[\text{s}] & \rightarrow & /\text{mange}/ \\
\text{tV}\,[\text{p}] & \rightarrow & /\text{mangent}/ \\
\text{stV}\,[\text{s}] & \rightarrow & /\text{est}/ \\
\text{stV}\,[\text{p}] & \rightarrow & /\text{sont}/ \\
\text{NP}\,[\text{m},\text{s}] & \rightarrow & /\text{Pierre}/ \\
\text{NP}\,[\text{f},\text{s}] & \rightarrow & /\text{Marie}/ \\
\text{N}\,[\text{m},\text{s}] & \rightarrow & /\text{fruit}/ \\
\text{N}\,[\text{m},\text{p}] & \rightarrow & /\text{fruits}/ \\
\text{N}\,[\text{f},\text{s}] & \rightarrow & /\text{pomme}/
\end{array}
\qquad
\begin{array}{lcl}
\text{N}\,[\text{f},\text{p}] & \rightarrow & /\text{pommes}/ \\
\text{Adj}\,[\text{m},\text{s}] & \rightarrow & /\text{intelligent}/ \\
\text{Adj}\,[\text{f},\text{s}] & \rightarrow & /\text{intelligente}/ \\
\text{Adj}\,[\text{m},\text{p}] & \rightarrow & /\text{intelligents}/ \\
\text{Adj}\,[\text{f},\text{p}] & \rightarrow & /\text{intelligentes}/ \\
\text{Det}\,[\text{m},\text{s}] & \rightarrow & /\text{un}/ \\
\text{Det}\,[\text{f},\text{s}] & \rightarrow & /\text{une}/ \\
\text{Det}\,[X,\text{p}] & \rightarrow & /\text{des}/ \\
\text{Conj} & \rightarrow & /\text{et}/
\end{array}
$$

Earley Algorithm

init

$$<S \rightarrow \bullet\alpha, 0, 0>$$

scan

$$\frac{<A \rightarrow \alpha \bullet a\beta, i, j>}{<A \rightarrow \alpha a \bullet \beta, i, j+1>} \quad a = a_{j+1}$$

complete

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j> \quad <C \rightarrow \gamma\bullet, j, k>}{<(A \rightarrow \alpha B \bullet \beta)\sigma, i, k>} \quad \sigma = \mathrm{mgu}(B, C)$$

predict

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j>}{<(C \rightarrow \bullet\gamma)\sigma, j, j>} \quad \sigma = \mathrm{mgu}(B, C)$$

- equality up to variable renaming, subsumption, completeness ?

Carl Pollard and Ivan A. Sag: Head-Driven Phrase Structure Grammar. University of Chicago Press, 1994.

Joan Bresnan: Lexical-Functional Syntax, Oxford: Blackwell Publishers Ltd, 2001.

Anne Abeillé: Les Nouvelles syntaxes. Armand Colin, 1993.

III. Tree Adjoining Grammars

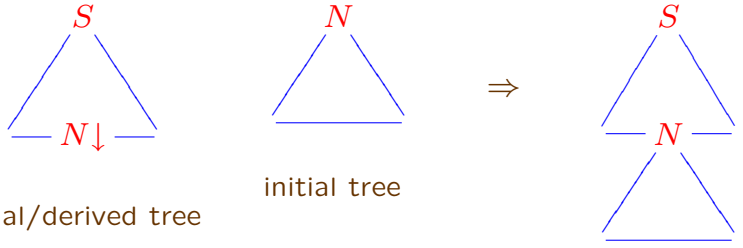
- Lexicalization
- Weak Equivalence versus Strong Equivalence
- Definition

  $G = (N, T, I, A, S)$

  where:
  - N is a finite set of non-terminal symbols;
  - T is a finite set of terminal symbols;
  - I is a finite set of trees called initial trees;
  - A is a finite set of trees called auxiliary trees;
  - $S \in N$ is the start symbol.

The trees in $I \cup A$ are called elementary trees. The inner nodes of the elementary trees are labeled by non-terminals. Their leaves are labeled by non-terminals or by terminals. In each auxiliary tree, there is one distinguished leaf (called the foot) whose label is the same non-terminal as the label of the root.

- Substitution

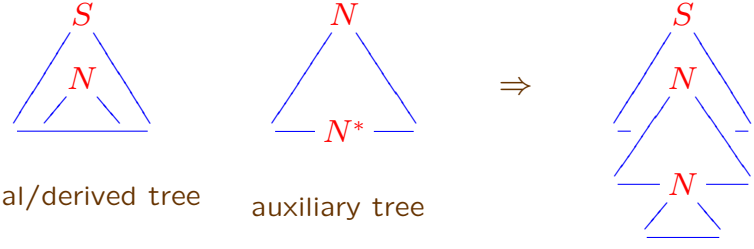


initial/derived tree

initial tree

derived tree

- Adjunction



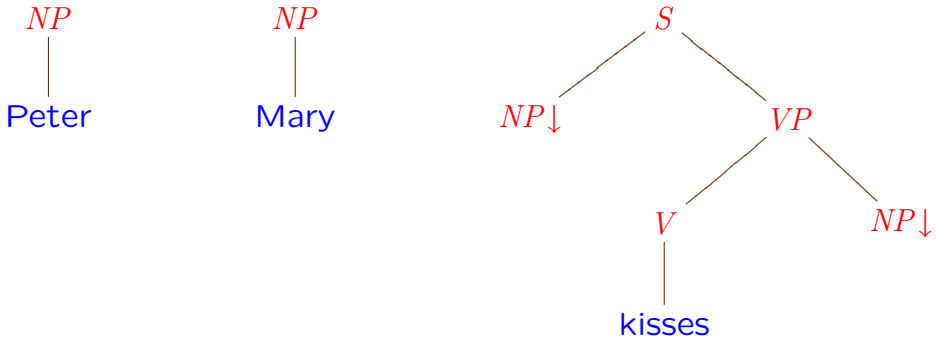
initial/derived tree

auxiliary tree

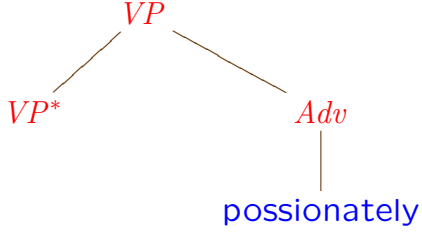
derived tree

Example

- Initial trees



- Auxiliary tree



CKY Algorithm

init

$$<A \rightarrow \bullet\alpha, i, \_, \_, i>$$

scan

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j, k, l>}{<A \rightarrow \alpha B \bullet \beta, i, j, k, l+1>}$$

label$(B) = a_{l+1}$

complete

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j, k, l> \quad <B \rightarrow \gamma\bullet, l, m, n, o>}{<A \rightarrow \alpha B \bullet \beta, i, j \sqcup m, k \sqcup n, o>}$$

$i \sqcup \_ = i$
$\_ \sqcup i = i$
undefined otherwise

foot

$$\frac{<A \rightarrow \alpha \bullet B\beta, i, \_, \_, j>}{<A \rightarrow \alpha B \bullet \beta, i, j, k, k>}$$

B is the foot of an auxiliary tree

adjoin

$$\frac{<A \rightarrow \alpha\bullet, i, j, k, l> \quad <B \rightarrow \beta\bullet, m, i, l, n>}{<A \rightarrow \beta\bullet, m, j, k, n>}$$

B is the root of an auxiliary tree
label(A) = label(B)

substitute

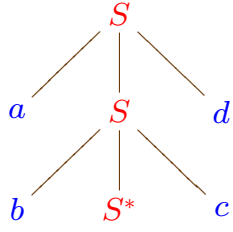
$$\frac{<A \rightarrow \alpha \bullet B\beta, i, j, k, l> \quad <C \rightarrow \gamma\bullet, l, \_, \_, m>}{<A \rightarrow \alpha B \bullet \beta, i, j, k, m>}$$

C is the root of an initial tree
label(B) = label(C)

Example

- Initial tree

- Auxiliary tree



- Let $\omega = {}_0a_1a_2b_3b_4e_5c_6c_7d_8d_9$

$$\dfrac{\dfrac{\dfrac{\dfrac{<A_2 \rightarrow \bullet bA_3c, 3, \_, \_, 3>}{<A_2 \rightarrow b \bullet A_3c, 3, \_, \_, 4>}\,\text{scan}}{<A_2 \rightarrow bA_3 \bullet c, 3, 4, 5, 5>}\,\text{foot}}{<A_2 \rightarrow bA_3c\bullet, 3, 4, 5, 6>}\,\text{scan} \qquad \dfrac{\dfrac{<A_1 \rightarrow \bullet aA_2d, 1, \_, \_, 1>}{<A_1 \rightarrow a \bullet A_2d, 1, \_, \_, 2>}\,\text{scan} \qquad \dfrac{\dfrac{\dfrac{<A_2 \rightarrow \bullet bA_3c, 2, \_, \_, 2>}{<A_2 \rightarrow b \bullet A_3c, 2, \_, \_, 3>}\,\text{scan}}{<A_2 \rightarrow bA_3 \bullet c, 2, 3, 6, 6>}\,\text{foot}}{<A_2 \rightarrow bA_3c\bullet, 2, 3, 6, 7>}\,\text{scan}}{<A_1 \rightarrow aA_2 \bullet d, 1, 3, 6, 7>}\,\text{complete}}{<A_1 \rightarrow aA_2d\bullet, 1, 3, 6, 8>}\,\text{scan}}{<A_2 \rightarrow aA_2d\bullet, 1, 4, 5, 8>}\,\text{adjoin}$$

$$\dfrac{\dfrac{<A_0 \rightarrow \bullet e, 4, \_, \_, 4>}{<A_0 \rightarrow e\bullet, 4, \_, \_, 5>}\,\text{scan} \qquad \dfrac{\dfrac{\dfrac{<A_1 \rightarrow \bullet aA_2d, 0, \_, \_, 0>}{<A_1 \rightarrow a \bullet A_2d, 0, \_, \_, 1>}\,\text{scan} \qquad \vdots \quad <A_2 \rightarrow aA_2d\bullet, 1, 4, 5, 8>}{<A_1 \rightarrow aA_2 \bullet d, 0, 4, 5, 8>}\,\text{complete}}{<A_1 \rightarrow aA_2d\bullet, 0, 4, 5, 9>}\,\text{scan}}{<A_0 \rightarrow aA_2d\bullet, 0, \_, \_, 9>}\,\text{adjoin}$$

- Good news/Bad news
- Earley algorithms (Schabes & Joshi, 1988; Schabes 1991; Nederhof 1997)
- Expressive power
- Adjoining constraints

Aravind K. Joshi, K. Vijay-Shanker: Some Computational Properties of Tree Adjoining Grammars. ACL 1985: 82-93

Yves Schabes, Aravind K. Joshi: An Earley-Type Parsing Algorithm for Tree Adjoining Grammars. ACL 1988: 258-269

Yves Schabes: The valid prefix property and left to right parsing of tree-adjoining grammar. IWPT 1991: 2–30

Mark-Jan Nederhof: The Computational Complexity of the Correct-Prefix Property for TAGs. Computational Linguistics 25(3): 345-360 (1999)

Eric Villemonte de la Clergerie: Tabulation et traitementde la langue naturelle. Tutorial 1999
http://pauillac.inria.fr/~clerger/

IV. Range Concatenation Grammars

- Expressive Power versus Tractability
- Definition

$G = (N, T, V, P, S)$

where:
  - N is a ranked alphabet of predicate names;
  - T is a finite set of terminal symbols;
  - V is a finite set of variable symbols;
  - P is a finite set of clauses

$$\phi_0 \rightarrow \phi_1 \dots \phi_n$$

where $\phi_0, \phi_1 \dots \phi_n$ are predicates of the form

$$A(\alpha_1, \dots, \alpha_p)$$

whith $A \in N$ and $\alpha_1, \dots, \alpha_n \in (T \cup V)^*$;
  - $S \in N$ is the start symbol.

- Notion of range
- Given a word $\omega \in T^*$, an instantiated clause is such that the variables and the predicate arguments are bound to ranges in $\omega \in T^*$
- Example:

$$\begin{array}{lcl} S(XY) & \rightarrow & S(X)\,E(X,Y) \\ S(a) & \rightarrow & \epsilon \\ E(Xa,Ya) & \rightarrow & E(X,Y) \\ E(\epsilon,\epsilon) & \rightarrow & \epsilon \end{array}$$

- Complete for PTIME.
- Closed by Union, Intersection, Concatenation, Iteration, Complementation.

See papers by Pierre Boullier
http://atoll.inria.fr

V. Categorial Grammars

- Radical approach to lexicalism
- An algebra of syntactic categories
- A finite set of grammatical composition rules

A notion of syntactic category

Let $\mathcal{A}$ be a finite set of atomic syntactic categories. The set of syntactic categories is inductively defined as follows:

$$\mathcal{T}_{\mathcal{A}} ::= \mathcal{A} \mid (\mathcal{T}_{\mathcal{A}} \bullet \mathcal{T}_{\mathcal{A}}) \mid (\mathcal{T}_{\mathcal{A}} \setminus \mathcal{T}_{\mathcal{A}}) \mid (\mathcal{T}_{\mathcal{A}} / \mathcal{T}_{\mathcal{A}})$$

Interpretation:

$(\alpha \bullet \beta)$ is the category of the phrases obtained by concatenating a phrase of category α with a phrase of category β.

$(\alpha \setminus \beta)$ is the category of the phrases that yield a phrase of category β when appended to a phrase of category α.

(β / α) is the category of the phrases that yield a phrase of category β when a phrase of category α is appended to them.

An algebra of syntactic categories

The set of syntactic categories is provided with a preorder:

$$\alpha \leq \beta$$

Interpretation :

Any phrase of category α is a phrase of category β.

$\leq$ is a preorder:

$$\alpha \leq \alpha$$

$$\alpha \leq \beta,\ \beta \leq \delta \Rightarrow \alpha \leq \delta$$

Associativity and monotonicity of $\bullet$:

$$(\alpha \bullet \beta) \bullet \gamma \leq \alpha \bullet (\beta \bullet \gamma)$$

$$\alpha \bullet (\beta \bullet \gamma) \leq (\alpha \bullet \beta) \bullet \gamma$$

$$\alpha \leq \beta \Rightarrow \alpha \bullet \gamma \leq \beta \bullet \gamma$$

$$\alpha \leq \beta \Rightarrow \gamma \bullet \alpha \leq \gamma \bullet \beta$$

Cancellation laws:

$$\alpha \bullet (\alpha \setminus \beta) \leq \beta$$

$$(\beta \,/\, \alpha) \bullet \alpha \leq \beta$$

AB-Grammars

$G = (\mathcal{A}, \Sigma, \mathcal{L}, s)$, where :

$\mathcal{A}$ is a finite set of atomic categories;

Σ is a finite vocabulary;

$\mathcal{L} : \Sigma \rightarrow 2^{\mathcal{T}_A}$ is a lexicon that assigns a finite set of syntactic categories to any element of the vocabulary;

$s \in \mathcal{T}_A$ is a distinguished category.

Let $G = \langle \mathcal{A}, \Sigma, \mathcal{L}, s \rangle$ be an AB-grammar.

The language $L(G)$ generated by G is the set of words

$$a_0 \dots a_n \in \Sigma^*$$

such that there exist

$$\alpha_0 \in \mathcal{L}(a_0), \dots, \alpha_n \in \mathcal{L}(a_n)$$

with

$$\alpha_0 \bullet \cdots \bullet \alpha_n \leq s$$

Expressive power

The class of AB-languages is the class of context-free languages.

Structural limitations

Pierre	:	SN
une	:	$SN \,/\, N$
pomme	:	N
mange	:	$(SN \setminus P) \,/\, SN$
qui	:	$(SN \setminus SN) \,/ (SN \setminus P)$
rapidement	:	$P \setminus P$

It is possible to generate:

Pierre qui mange une pomme

$$SN \bullet ((SN \setminus SN) \,/ (SN \setminus P)) \bullet ((SN \setminus P) \,/\, SN) \bullet (SN \,/\, N) \bullet N \leq SN$$

Pierre mange une pomme rapidement

$$SN \bullet ((SN \setminus P) \,/\, SN) \bullet (SN \,/\, N) \bullet N \bullet (P \setminus P) \leq P$$

It is not possible to generate:

Pierre qui mange une pomme rapidement

because the following inequality:

$$(SN \setminus P) \bullet (P \setminus P) \leq (SN \setminus P)$$

cannot be derived

Hypothetical reasoning

To generate:

Pierre qui mange une pomme rapidement

the following inequality is needed:

$$(SN \setminus P) \bullet (P \setminus P) \leq (SN \setminus P)$$

Similarly, to generate:

une pomme que Pierre mange

where

$$\text{que} : (SN \setminus SN) \,/\, (P \,/\, SN)$$

the following inequality is needed:

$$SN \bullet ((SN \setminus P) \,/\, SN) \leq (P \,/\, SN)$$

Both inequalities are consistent with respect to the interpretation of the preorder but cannot be derived using the algebraic laws given so far.

The algebra previously given does not capture completely the intuition we have of the connectives.

Hypothetical reasoning:

Assume:

$$\mathsf{X} : SN$$

then:

$$\mathsf{X\ mange\ rapidement} : P$$

hence:

$$\mathsf{mange\ rapidement} : SN \setminus P$$

Enriching the algebra

Add the following adjunction laws:

$$(\alpha \bullet \beta) \leq \gamma \Rightarrow \beta \leq (\alpha \setminus \gamma)$$

$$(\alpha \bullet \beta) \leq \gamma \Rightarrow \alpha \leq (\gamma / \beta)$$

Logical formalization

There exists a logical reading of:

$$\alpha_0 \bullet \cdots \bullet \alpha_n \leq \beta$$

where:

- the α_i's are seen as hypotheses;
- β plays the part of the conclusion;
- $\leq$ is interpreted as a consequence relation.

According to this reading, the syntactic categories correspond to formulas. $\bullet$ is a conjonction, and both $\backslash$ and $/$ are implications.

Lambek calculus

$$A \vdash A \quad \text{(ident)} \qquad \frac{\Gamma \vdash A \quad \Delta_1, A, \Delta_2 \vdash B}{\Delta_1, \Gamma, \Delta_2 \vdash B} \quad \text{(cut)}$$

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \bullet B, \Delta \vdash C} \quad (\bullet \text{ left}) \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \bullet B} \quad (\bullet \text{ right})$$

$$\frac{\Gamma \vdash A \quad \Delta_1, B, \Delta_2 \vdash C}{\Delta_1, \Gamma, A \setminus B, \Delta_2 \vdash C} \quad (\setminus \text{ left}) \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \setminus B} \quad (\setminus \text{ right})$$

$$\frac{\Gamma \vdash A \quad \Delta_1, B, \Delta_2 \vdash C}{\Delta_1, B / A, \Gamma, \Delta_2 \vdash C} \quad (/ \text{ left}) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash B / A} \quad (/ \text{ right})$$

Lambek grammars

Same data as an AB-grammar.

The langage $L(G)$ generated by a Lambek-grammar G is the set of words

$$a_0 \dots a_n \in \Sigma^*$$

such that there exist

$$\alpha_0 \in \mathcal{L}(a_0), \dots, \alpha_n \in \mathcal{L}(a_n)$$

with the following sequent

$$\alpha_0, \dots, \alpha_n \vdash s$$

being derivable.

Decidability

Cut elimination. Every derivable sequent is derivable without using Rule (cut).

Corollary. The Lambek calculus is decidable.

Elements of model theory

Let $\langle \Sigma^*, +, \epsilon \rangle$ be the free monoid generated by Σ.

A valuation is defined to be a fonction $\rho : \mathcal{A} \to 2^{\Sigma^*}$ that assigns a set of words to each atomic formula (syntactic category).

Given such a valuation ρ, interpret the formulas as follows:

$$[\![\alpha]\!]\rho = \rho(\alpha) \text{ where } \alpha \text{ is atomic}$$

$$[\![\alpha \bullet \beta]\!]\rho = \{u \in \Sigma^* \mid \exists a \in [\![\alpha]\!]\rho.\, \exists b \in [\![\beta]\!]\rho.\, u = a + b\}$$

$$[\![\alpha \setminus \beta]\!]\rho = \{u \in \Sigma^* \mid \forall a \in [\![\alpha]\!]\rho.\, a + u \in [\![\beta]\!]\rho\}$$

$$[\![\alpha / \beta]\!]\rho = \{u \in \Sigma^* \mid \forall b \in [\![\beta]\!]\rho.\, u + b \in [\![\alpha]\!]\rho\}$$

Let $\Gamma = \gamma_0, \ldots, \gamma_n$, and define:

$$[\![\Gamma]\!]\rho = [\![\gamma_0 \bullet \cdots \bullet \gamma_n]\!]\rho$$

A valuation ρ satisfies a sequent $\Gamma \vdash \alpha$ iff

$$[\![\Gamma]\!]\rho \subset [\![\alpha]\!]\rho$$

A sequent $\Gamma \vdash \alpha$ is valid iff it is satisfied by every valuation.

Correctness. Every derivable sequent is valid.

Completness (Pentus, 1993). Every valid sequent is derivable.

Expressive Power

(Pentus, 1992) The class of languages generated by the Lambek grammars is the class of context-free languages.

Structural limitations

The non-commutativity of the Lambek calculus is to "rigid".

Among others, it does not allow for medial wh-extraction.

For instance:

Une pomme que Pierre mange rapidement

cannot be generated.

- Multimodal grammars (Michael Moortgat)
- Type-logical grammars (Glyn Morrill)
- Combinatory Categorial Grammars (Mark Steedman)

M. Pentus: Lambek Grammars are Context Free. LICS 1993: 429-433

M. Pentus: Language completeness of the Lambek calculus. LICS 1994: 487-496

M. Moortgat: Categorial Type Logics. Chapter 2 of Handbook of Logic and Language, Elsevier, 1997