Constants of grammatical reasoning:

Slides

Michael Moortgat
## Contents

1. Slides ................................................................. 4  
1.1 Logic and linguistics ............................................ 4  
1.2 Key concept: ‘Parsing-as-deduction’ .......................... 5  
1.3 ‘Derivation’ and ‘Meaning’ ...................................... 6  
1.4 Central questions .................................................. 7  
1.5 Converging ideas ................................................... 8  
1.6 Logical connectives, structural rules ........................... 9  
1.7 Logical rules: which version characterizes ‘conjunction’? .. 11  
1.8 Weakening, Contraction: loss of discrimination .......... 12  
1.9 Linear Logic: splitting connectives ............................ 13  
1.10 Recovering lost expressivity ................................. 14  
1.11 Resource sensitivity in grammar .............................. 15  
1.12 Valency principles ................................................. 16  
1.13 Resource-sensitivity of grammatical reasoning .......... 17  
1.14 Saturating incompleteness: modus ponens .................. 18  
1.15 Phrase structure rules .......................................... 19  
1.16 Hypothetical reasoning ......................................... 20  

Contents  ◀
1. Slides

1.1. Logic and linguistics

**Linguistics** study of the cognitive apparatus that underlies our linguistic abilities:

- use and understanding of language,
- learning: language acquisition
1. Slides

1.1. Logic and linguistics

Linguistics study of the cognitive apparatus that underlies our linguistic abilities:

- use and understanding of language,
- learning: language acquisition

Logic formal study of reasoning/inference, statically (proofs as objects) and dynamically (the process of derivation/computation)
1. Slides

1.1. Logic and linguistics

Linguistics study of the cognitive apparatus that underlies our linguistic abilities:
- use and understanding of language,
- learning: language acquisition

Logic formal study of reasoning/inference, statically (proofs as objects) and dynamically (the process of derivation/computation)

An integrated view on the two disciplines:

Cognition = Computation
Grammar = Logic
1.2. Key concept: ‘Parsing-as-deduction’

PARSING: ‘Is the sequence of words $w_1 \cdots w_n$ a well-formed expression of type $B$?’

\[ w_1 \cdots w_n \]
1.2. Key concept: ‘Parsing-as-deduction’

PARSING: ‘Is the sequence of words $w_1 \cdots w_n$ a well-formed expression of type $B$?’

\[
\begin{array}{c}
  w_1 & \cdots & w_n \\
  \vdots & \vdots \\
  A_1 & \cdots & A_n \vdash B \\
\end{array}
\]

\[\Gamma\]
1.2. Key concept: ‘Parsing-as-deduction’

PARSING: ‘Is the sequence of words $w_1 \cdots w_n$ a well-formed expression of type $B$?’

$$\begin{align*}
  w_1 & \cdots & w_n \\
  \vdots & \quad \vdots \\
  A_1 & \cdots & A_n \vdash B
\end{align*}$$

DEDUCTION: ‘Given $A_i$ as logical ‘parts-of-speech’ of $w_i$, Is a conclusion $B$ derivable from assumptions $A_1, \ldots, A_n$, structured as a configuration $\Gamma$?’
1.3. ‘Derivation’ and ‘Meaning’

\[ w_1 \quad \cdots \quad w_n \]
1.3. ‘Derivation’ and ‘Meaning’

\[
\begin{align*}
\vdots & \vdots \\
{w_1} & \cdots & {w_n} \\
{x_1 : A_1} & \cdots & {x_n : A_n} \vdash t : B \\
\hline
\end{align*}
\]

\[\Gamma\]
1.3. ‘Derivation’ and ‘Meaning’

\[
\begin{align*}
&w_1 & \cdots & w_n \\
&\vdots & \quad & \vdots \\
&x_1 : A_1 & \cdots & x_n : A_n & \vdash & t : B \\
\Gamma
\end{align*}
\]

The grammatical logic as a PROGRAMMING LANGUAGE: read the proof of the derivation \( A_1, \ldots, A_n \vdash B \) as an instruction for the assembly of a meaning program \( t \) with input parameters \( x_1, \ldots, x_n \).
1.4. Central questions

- What are the CONSTANTS of grammatical reasoning?

    \[ \rightarrow \text{capture} \textit{invariants} \text{ of the form/meaning correspondence across languages in terms of this vocabulary of logical constants, together with the deductive principles governing their use} \]
1.4. Central questions

- What are the CONSTANTS of grammatical reasoning?

  $\mapsto$ capture *invariants* of the form/meaning correspondence across languages in terms of this vocabulary of logical constants, together with the deductive principles governing their use

- How to reconcile the idea of ‘grammatical constants’ with the differences between languages?

  $\mapsto$ the logical space for STRUCTURAL VARIATION in the realization of the form/meaning correspondence
1.5. Converging ideas

- Linear Logic (Girard 1987)

~~~

‘splitting’ of logical constants:

core + resource management (weakening, contraction)

- Logic of production/consumption of finite resources
- Connectives (‘modalities’) for explicit control over resource multiplicity

~~~
1.5. Converging ideas

- Linear Logic (Girard 1987)

  $\leadsto$ ‘splitting’ of logical constants:
  core + resource management (weakening, contraction)
  
  - Logic of production/consumption of finite resources
  - Connectives (‘modalities’) for explicit control over resource multiplicity

- Categorial Grammar (Lambek 1958)

  $\leadsto$ left- vs right-handed implications
  
  - Logic of structured grammatical resources (‘signs’)
  - Logical constants controlling structural management of grammatical resources
1.6. Logical connectives, structural rules

In ‘standard’ logical systems, one considers sequents $\Gamma \Rightarrow A$, with $A, B, C, \ldots$ formulas, and $\Gamma, \Delta, \ldots$ multisets of formulas. (A multiset is a set of occurrences. \{a, a, b\} and \{a, b\} are the same set, but different multisets: two occurrences of a versus one.)
1.6. Logical connectives, structural rules

In ‘standard’ logical systems, one considers sequents $\Gamma \Rightarrow A$, with $A, B, C, \ldots$ formulas, and $\Gamma, \Delta, \ldots$ multisets of formulas. (A multiset is a set of occurrences. \{a, a, b\} and \{a, b\} are the same set, but different multisets: two occurrences of $a$ versus one.)

A derivation proceeds from axioms $A \Rightarrow A$, using logical rules (telling you how to introduce a connective to the left or to the right of $\Rightarrow$), and structural rules (telling you how you can manipulate the formulas in a derivation). The standard structural rules are:

- **Weakening**: ‘waste’ of assumptions
  \[
  \dfrac{\Gamma \Rightarrow B}{\Gamma, A \Rightarrow B} W
  \]

- **Contraction**: ‘duplication’ of assumptions
  \[
  \dfrac{\Gamma, A \Rightarrow B}{\Gamma, A \Rightarrow B} C
  \]
1.7. Logical rules: which version characterizes ‘conjunction’?

Below two ways of formulating the logical rules for ‘conjunction’. They differ critically in the rule that introduces the connective to the right of $\Rightarrow$. 
1.7. Logical rules: which version characterizes ‘conjunction’?

Below two ways of formulating the logical rules for ‘conjunction’. They differ critically in the rule that introduces the connective to the right of $\Rightarrow$.

- ‘context-sensitive’ version: the same $\Gamma$ in the two premises

\[
\begin{align*}
\Gamma \Rightarrow A & \quad \Gamma \Rightarrow B \\
\frac{}{\Gamma \Rightarrow A \land B} \quad \land R
\end{align*}
\]

\[
\begin{align*}
\Gamma, A_i \Rightarrow C \\
\frac{}{\Gamma, A_1 \land A_2 \Rightarrow C} \quad \land L, i \in \{1, 2\}
\end{align*}
\]
1.7. Logical rules: which version characterizes ‘conjunction’?

Below two ways of formulating the logical rules for ‘conjunction’. They differ critically in the rule that introduces the connective to the right of \( \Rightarrow \).

- ‘context-sensitive’ version: the same \( \Gamma \) in the two premises

\[
\frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \land B} \quad \wedge R
\]

\[
\frac{\Gamma, A_i \Rightarrow C}{\Gamma, A_1 \land A_2 \Rightarrow C} \quad \wedge L, i \in \{1, 2\}
\]

- ‘context-free’ version: distinct \( \Gamma \) and \( \Delta \) in the two premises

\[
\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \land B} \quad \wedge R'
\]

\[
\frac{\Gamma, A, B \Rightarrow C}{\Gamma, A \land B \Rightarrow C} \quad \wedge L'
\]
1.8. **Weakening, Contraction: loss of discrimination**

In the presence of Contraction/Weakening, $\land R$ and $\land R'$ ($\land L$, $\land L'$) are interderivable: the context-free and the context-sensitive rules define the same connective.
1.8. Weakening, Contraction: loss of discrimination

In the presence of Contraction/Weakening, $\land R$ and $\land R'$ ($\land L$, $\land L'$) are interderivable: the context-free and the context-sensitive rules define the same connective.

- deriving $\land R'$ from $\land R$ with Weakening:

$$
\frac{
\frac{
\frac{\Gamma \Rightarrow A}{\Gamma, \Delta \Rightarrow A} \quad W \quad \frac{\Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow B}
}{\Gamma, \Delta \Rightarrow A \land B} \quad W
}{\Gamma, \Delta \Rightarrow \land R}
$$
1.8. Weakening, Contraction: loss of discrimination

In the presence of Contraction/Weakening, \( \land R \) and \( \land R' \) (\( \land L, \land L' \)) are interderivable: the context-free and the context-sensitive rules define the same connective.

- deriving \( \land R' \) from \( \land R \) with Weakening:

\[
\begin{align*}
\Gamma \Rightarrow A & \quad \Delta \Rightarrow B \\
\Delta \Rightarrow B & \quad \Delta \Rightarrow B
\end{align*}
\]

\[
\frac{\Gamma, \Delta \Rightarrow A \quad W}{\Gamma, \Delta \Rightarrow A \land B} \quad \frac{\Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \land B} \quad W
\]

\[
\land R
\]

- deriving \( \land R \) from \( \land R' \) with Contraction:

\[
\begin{align*}
\Gamma \Rightarrow A & \quad \Gamma \Rightarrow B \\
\Gamma \Rightarrow B & \quad \Gamma \Rightarrow B
\end{align*}
\]

\[
\frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \land B} \quad \land R'
\]

\[
\frac{\Gamma \Rightarrow A \land B}{\Gamma \Rightarrow A \land B} \quad C
\]
1.9. Linear Logic: splitting connectives

In the absence of Contraction/Weakening, assumptions become finite *resources*; two distinct senses of ‘conjunction’ can be discriminated:
1.9. Linear Logic: splitting connectives

In the absence of Contraction/Weakening, assumptions become finite resources; two distinct senses of ‘conjunction’ can be discriminated:

- ‘context-free’, multiplicative tensor: resource composition

\[
\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \otimes B} \otimes R \quad \frac{\Gamma, A, B \Rightarrow C}{\Gamma, A \otimes B \Rightarrow C} \otimes L
\]
1.9. Linear Logic: splitting connectives

In the absence of Contraction/Weakening, assumptions become finite *resources*; two distinct senses of ‘conjunction’ can be discriminated:

- ‘context-free’, *multiplicative* tensor: resource composition

\[
\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \otimes B} \otimes R \quad \frac{\Gamma, A, B \Rightarrow C}{\Gamma, A \otimes B \Rightarrow C} \otimes L
\]

- ‘context-sensitive’ *additive* ‘and’:

\[
\frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \sqcap B} \sqcap R \quad \frac{\Gamma, A_i \Rightarrow C}{\Gamma, A_1 \sqcap A_2 \Rightarrow C} \sqcap L, i \in \{1, 2\}
\]
1.10. Recovering lost expressivity

- Contraction/Weakening can be reintroduced in a controlled form, by means of a modality/exponential ‘!’.
1.10. Recovering lost expressivity

- Contraction/Weakening can be reintroduced in a controlled form, by means of a modality/exponential ‘!’:

  - logical rules:

\[
\frac{\Gamma, A \implies B}{\Gamma, !A \implies B} !L \quad \frac{!\Gamma \implies A}{!\Gamma \implies !A} !R
\]
1.10. Recovering lost expressivity

- Contraction/Weakening can be reintroduced in a controlled form, by means of a modality/exponential ‘!’:

  - logical rules:

    $$
    \frac{\Gamma, A \Rightarrow B}{\Gamma, !A \Rightarrow B} \quad !L \quad \frac{!\Gamma \Rightarrow A}{!\Gamma \Rightarrow !A} \quad !R
    $$

  - structural rules:

    $$
    \frac{\Gamma \Rightarrow B}{\Gamma, !A \Rightarrow B} \quad !W \quad \frac{\Gamma, !A, !A \Rightarrow B}{\Gamma, !A \Rightarrow B} \quad !C
    $$
1.10. Recovering lost expressivity

- Contraction/Weakening can be reintroduced in a controlled form, by means of a modality/exponential ‘!’:

  - logical rules:

    \[
    \frac{\Gamma, A \Rightarrow B}{\Gamma, !A \Rightarrow B} \quad L \quad \frac{!\Gamma \Rightarrow A}{!\Gamma \Rightarrow !A} \quad R
    \]

  - structural rules:

    \[
    \frac{\Gamma \Rightarrow B}{\Gamma, !A \Rightarrow B} \quad W \quad \frac{!\Gamma, !A \Rightarrow B}{\Gamma, !A \Rightarrow B} \quad C
    \]

- Modalities establish *communication* between ‘split’ connectives \( \otimes, \sqcap \):

  \[!A \otimes !B \iff !(A \sqcap B)\]
1.11. Resource sensitivity in grammar

LOCAL DEPENDENCIES. Subcategorization and grammatical ‘incompleteness’. Compare:

\begin{itemize}
\item \text{a} NOT ENOUGH! *the Mad Hatter offered
\item \text{b} the Mad Hatter offered Alice a cup of tea
\item \text{c} TOO MUCH! *the Cheshire Cat grinned Alice a cup of tea
\item \text{d} the Cheshire Cat grinned
\end{itemize}
1.11. Resource sensitivity in grammar

LOCAL DEPENDENCIES. Subcategorization and grammatical ‘incompleteness’. Compare:

\[
\begin{align*}
 a & \quad \text{NOT ENOUGH!} \quad *\text{the Mad Hatter offered} \\
b & \quad \text{the Mad Hatter offered Alice a cup of tea} \\
c & \quad \text{TOO MUCH!} \quad *\text{the Cheshire Cat grinned Alice a cup of tea} \\
d & \quad \text{the Cheshire Cat grinned}
\end{align*}
\]

UNBOUNDED DEPENDENCIES. Long-distance ‘movement’.

\[
\begin{align*}
 a & \quad \text{the tarts which the Hatter offered the March Hare} \\
b & \quad *\text{the tarts which the Hatter offered the March Hare a present} \\
c & \quad \text{the tarts which Alice thought the Hatter offered the March Hare} \\
d & \quad \text{the tarts which the Dormouse said Alice thought} \ldots
\end{align*}
\]
1.12. Valency principles

Completeness and Coherence An f-structure is *locally complete* iff it contains all the governable functions that its predicate governs, and *locally coherent* iff all the governable grammatical functions that it contains are governed by a local predicate. (cf Lexical Functional Grammar, Bresnan ea)

Subcategorization Principle In a headed phrase, the SUBCAT value of the head daughter is the concatenation of the phrase’s SUBCAT list with the list of SYNSEM values of the complement daughters. (cf HPSG, Pollard & Sag)

Theta Criterion Each argument bears one and only one $\theta$ role, and each $\theta$ role is assigned to one and only one argument. (cf GB, Chomsky)

Task. Characterize a ‘logic of grammar’ where such principles are built-in features.
1.13. Resource-sensitivity of grammatical reasoning

We model grammatical incompleteness in terms of a resource-sensitive implication $\circ\rightarrow$, i.e. an implication $A\circ\rightarrow B$ which actually ‘uses up’ an $A$ resource in the process of producing a $B$. 
1.13. **Resource-sensitivity of grammatical reasoning**

We model grammatical incompleteness in terms of a resource-sensitive implication $\implies$, i.e. an implication $A \implies B$ which actually ‘uses up’ an $A$ resource in the process of producing a $B$.

**Modus ponens.** Cf local dependencies, subcategorization.

\[
\Gamma \vdash A \quad \Delta \vdash A \implies B \quad (E \implies) \\
\Gamma \circ \Delta \vdash B
\]
1.13. Resource-sensitivity of grammatical reasoning

We model grammatical incompleteness in terms of a resource-sensitive implication \(-\circ\), i.e. an implication \(A\rightarrow\circ B\) which actually ‘uses up’ an \(A\) resource in the process of producing a \(B\).

MODUS PONENS. Cf local dependencies, subcategorization.

\[
\frac{\Gamma \vdash A \quad \Delta \vdash A \rightarrow \circ B}{\Gamma \circ \Delta \vdash B} (E\rightarrow\circ)
\]

HYPOTHETICAL REASONING. Cf Unbounded dependencies, ‘movement’.

\[
\frac{A \circ \Gamma \vdash B}{\Gamma \vdash A \rightarrow \circ B} (I\rightarrow\circ)
\]
1.14. Saturating incompleteness: modus ponens

Lexical resources:

1. Alice $\vdash np$ \hspace{0.5cm} Lex
2. talks $\vdash pp \circ (np \circ s)$ \hspace{0.5cm} Lex
3. to $\vdash np \circ pp$ \hspace{0.5cm} Lex
4. the $\vdash n \circ np$ \hspace{0.5cm} Lex
5. footman $\vdash n$ \hspace{0.5cm} Lex
1.14. Saturating incompleteness: modus ponens

Lexical resources:

1. Alice ⊢ np \hspace{1cm} Lex
2. talks ⊢ pp − ◦ (np − ◦ s) \hspace{1cm} Lex
3. to ⊢ np − ◦ pp \hspace{1cm} Lex
4. the ⊢ n − ◦ np \hspace{1cm} Lex
5. footman ⊢ n \hspace{1cm} Lex

Grammatical composition:

6. the ◦ footman ⊢ np \hspace{1cm} − ◦ E (4, 5)
7. to ◦ the ◦ footman ⊢ pp \hspace{1cm} − ◦ E (3, 6)
8. talks ◦ to ◦ the ◦ footman ⊢ np − ◦ s \hspace{1cm} − ◦ E (2, 7)
9. Alice ◦ talks ◦ to ◦ the ◦ footman ⊢ s \hspace{1cm} − ◦ E (1, 8)
1.15. Phrase structure rules

Compare: CFG rewriting rules (no logical constants!)

\[
\begin{align*}
\text{s} & \rightarrow \text{np, vp} & \text{pn} & \rightarrow \text{alice} \\
\text{np} & \rightarrow \text{pn} & \text{det} & \rightarrow \text{the} \\
\text{np} & \rightarrow \text{det, n} & \text{n} & \rightarrow \text{footman} \\
\text{vp} & \rightarrow \text{v(pp), pp} & \text{v(pp)} & \rightarrow \text{talks} \\
\text{pp} & \rightarrow \text{prep, np} & \text{prep} & \rightarrow \text{to}
\end{align*}
\]
1.16. Hypothetical reasoning

- ‘whom’: $(np \circ s) \circ (n \circ n)$
- the higher-order formula triggers a process of conditional reasoning w.r.t. a hypothetical $np$ resource: see the $[\circ I]$ step
- the ‘filler-gap’ dependency between the relative pronoun and the pre-empted $np$ position in the relative clause body is established in deductive terms.
1.16. Hypothetical reasoning

- ‘whom’: \((np \rightarrow s) \rightarrow (n \rightarrow n)\)
- the higher-order formula triggers a process of conditional reasoning w.r.t. a hypothetical \(np\) resource: see the \([-\rightarrow I]\) step
- the ‘filler-gap’ dependency between the relative pronoun and the pre-empted \(np\) position in the relative clause body is established in deductive terms.

\[
\begin{align*}
6'. & \quad x \vdash np & Hyp \\
7'. & \quad \text{to } \circ x \vdash pp & \neg \circ E (3, 6') \\
8'. & \quad \text{talks } \circ \text{to } \circ x \vdash np \rightarrow s & \neg \circ E (2, 7') \\
9'. & \quad \text{Alice } \circ \text{talks } \circ \text{to } \circ x \vdash s & \neg \circ E (1, 8') \\
10'. & \quad \text{Alice } \circ \text{talks } \circ \text{to} \vdash np \rightarrow s & \neg I (6', 9') \\
11'. & \quad \text{whom } \circ \text{Alice } \circ \text{talks } \circ \text{to} \vdash n \rightarrow n & \neg \circ E (0, 10') \\
12'. & \quad \text{footman } \circ \text{whom } \circ \text{Alice } \circ \text{talks } \circ \text{to} \vdash n & \neg \circ E (5, 11') \\
13'. & \quad \text{the } \circ \text{footman } \circ \text{whom } \circ \text{Alice } \circ \text{talks } \circ \text{to} \vdash np & \neg \circ E (4, 12')
\end{align*}
\]
1.17. Grammatical composition: the meaning dimension

\[
\frac{\Gamma \vdash u : A}{\Gamma \circ \Delta \vdash tu : B} \quad (-\circ E) \quad \frac{x : A \circ \Gamma \vdash t : B}{\Gamma \vdash \lambda x.t : A \circ B} \quad (-\circ I)
\]
1.18. Meaning assembly

- PROOFS AS MEANING PROGRAMS. The composition of form and meaning proceeds in parallel, and is fully ‘inference-driven’. There is no structural representation level of the grammatical resources (such as ‘Logical Form’) where meaning is read off. Instead, meaning is computed from the derivational process that puts the resources together.
1.18. Meaning assembly

- PROOFS AS MEANING PROGRAMS. The composition of form and meaning proceeds in parallel, and is fully ‘inference-driven’. There is no *structural* representation level of the grammatical resources (such as ‘Logical Form’) where meaning is read off. Instead, meaning is computed from the derivational process that puts the resources together.

- MEANING PARAMETRICITY. The actual meanings of the resources that enter into the composition process are ‘black boxes’ for the Curry-Howard computation. No assumptions about the content of the actual meanings can be built into the meaning assembly process.
1.18. Meaning assembly

- PROOFS AS MEANING PROGRAMS. The composition of form and meaning proceeds in parallel, and is fully ‘inference-driven’. There is no structural representation level of the grammatical resources (such as ‘Logical Form’) where meaning is read off. Instead, meaning is computed from the derivational process that puts the resources together.

- MEANING PARAMETRICITY. The actual meanings of the resources that enter into the composition process are ‘black boxes’ for the Curry-Howard computation. No assumptions about the content of the actual meanings can be built into the meaning assembly process.

- RESOURCE SENSITIVITY. Because the grammar logic has a resource-sensitive notion of inference (each assumption is used exactly once), there is no need for ‘syntactic’ book-keeping stipulations restricting variable occurrences: vacuous abstractions, closed subterms, multiple binding of variables, or unbound variables (other than the proof parameters) simply do not arise.
1.19. Lexical versus derivational meaning

Proof terms:

9. \((\text{talk} \,(\text{to} \,(\text{the} \,\text{footman})))) \, \text{Alice}\)
10'. \(\lambda x.((\text{talk} \,(\text{to} \,x)) \,\text{Alice})\)
13'. \((\text{the} \,(\,(\text{whom} \,\lambda x.((\text{talk} \,(\text{to} \,x)) \,\text{Alice})) \,\text{footman})))\)
1.19. Lexical versus derivational meaning

Proof terms:

9. ((talk (to (the footman))) Alice)
10'. λx.((talk (to x)) Alice)
13'. (the ((whom λx.((talk (to x)) Alice)) footman))

Lexical meanings (possibly non-linear!):

\[
\begin{align*}
a \quad \text{whom} : (np \circ s) \circ (n \circ n) & - \lambda x_1 \lambda x_2 \lambda x_3. (x_1 x_3) \land (x_2 x_3) \\
b \quad \text{the} (\lambda x.((\text{talk (to x)}) \ Alice) \land (\text{footman} \ x))
\end{align*}
\]
1.20. **Structured resources**

\[ S ::= \mathcal{F} \mid (S \circ S) \]

We move to sequents where the antecedent is a *structured database* $S$ of formulas $\mathcal{F}$ (instead of an unstructured multiset of formula occurrences). In this setting, one can consider new structural rules affecting the configuration of the database.
1.20. Structured resources

\[ S ::= \mathcal{F} \mid (S \circ S) \]

We move to sequents where the antecedent is a structured database \( S \) of formulas \( \mathcal{F} \) (instead of an unstructured multiset of formula occurrences). In this setting, one can consider new structural rules affecting the configuration of the database.

- Permutation: destroys linear order information

\[
\begin{align*}
\Gamma[\Delta_2 \circ \Delta_1] &\Rightarrow A \\
\Gamma[\Delta_1 \circ \Delta_2] &\Rightarrow A \\
&\Rightarrow A \quad P
\end{align*}
\]
1.20. **Structured resources**

\[ S ::= \mathcal{F} \mid (S \circ S) \]

We move to sequents where the antecedent is a *structured database* \( S \) of formulas \( \mathcal{F} \) (instead of an unstructured multiset of formula occurrences). In this setting, one can consider new structural rules affecting the configuration of the database.

- **Permutation:** destroys linear order information

\[
\begin{align*}
\Gamma[\Delta_2 \circ \Delta_1] & \Rightarrow A \\
\Gamma[\Delta_1 \circ \Delta_2] & \Rightarrow A \quad P
\end{align*}
\]

- **Associativity:** destroys hierarchical grouping (‘constituents’)

\[
\begin{align*}
\Gamma[(\Delta_1 \circ \Delta_2) \circ \Delta_3] & \Rightarrow A & \Gamma[\Delta_1 \circ (\Delta_2 \circ \Delta_3)] & \Rightarrow A \quad \text{(l)} \quad A(l) \\
\Gamma[\Delta_1 \circ (\Delta_2 \circ \Delta_3)] & \Rightarrow A & \Gamma[(\Delta_1 \circ \Delta_2) \circ \Delta_3] & \Rightarrow A \quad \text{(r)} \quad A(r)
\end{align*}
\]

(Notation: \( \Gamma[\Delta] \): database \( \Gamma \) with substructure \( \Delta \).)
1.21. **Splitting connectives: structure sensitivity**

Dropping Permutation/Associativity as hard-wired features gives rise to new connectives with greater structural discrimination.
1.21. **Splitting connectives: structure sensitivity**

Dropping Permutation/Associativity as hard-wired features gives rise to new connectives with greater structural discrimination.

- Without Permutation, two implications become distinguishable: left vs right incompleteness. Order sensitivity.

\[
A \circ A \backslash B \Rightarrow B \quad \text{versus} \quad B / A \circ A \Rightarrow B
\]

\[
\begin{align*}
A \circ \Gamma \Rightarrow B && \Gamma \circ A \Rightarrow B \\
\Gamma \Rightarrow A \backslash B && \Gamma \Rightarrow B / A
\end{align*}
\]
1.21. Splitting connectives: structure sensitivity

Dropping Permutation/Associativity as hard-wired features gives rise to new connectives with greater structural discrimination.

- Without Permutation, two implications become distinguishable: left vs right incompleteness. Order sensitivity.

\[
A \circ A \setminus B \Rightarrow B \quad \text{versus} \quad B/A \circ A \Rightarrow B
\]

\[
\frac{A \circ \Gamma \Rightarrow B}{\Gamma \Rightarrow A \setminus B} \quad \text{versus} \quad \frac{\Gamma \circ A \Rightarrow B}{\Gamma \Rightarrow B/A}
\]

- Without Associativity, the implications respect constituent structure. Restructuring (composition) is no longer derivable:

\[
\not\vdash (C \setminus B) \circ (B \setminus A) \Rightarrow C \setminus A \quad \not\vdash (A/B) \circ (B/C) \Rightarrow A/C
\]
1.22. Structural control: reordering

Instead of global structural choices, we need lexical control over resource management.

ILLUSTRATION: Latin is a language with very ‘free’ word order. But it is not the case that all permutations are well-formed.

a  cum magna laude
b  cum laude magna
c  magna cum laude
d  *magna laude cum
1.23. Structural control: restructuring

a. the Lobster loves but the Gryphon hates Turtle Soup

b. the Mad Hatter loves himself

c. *the Mad Hatter thinks Alice loves himself
1.23. Structural control: restructuring

a. the Lobster loves but the Gryphon hates Turtle Soup

b. the Mad Hatter loves himself

c. *the Mad Hatter thinks Alice loves himself

Restructuring is fine for (a): one can group ‘non-constituents’ consisting of subject+transitive verb, and conjoin them. But with global Associativity there is no way of distinguishing a simple transitive verb ((np\$)/np) from an arbitrary string that can be turned into an s by adding an np to the left and to the right. Yet ‘himself’ (of type: ((np\$)/np)\((np\$)/np\)) is OK in (b) but not in (c).
1.24. Recovering control

Lost expressivity is regained in terms of modalities ♦, □ for structural control:
1.24. Recovering control

Lost expressivity is regained in terms of modalities ◊, □ for structural control:

- Logical rules: residuation

\[◊A → B \iff A → □B\]
1.24. Recovering control

Lost expressivity is regained in terms of modalities $\Diamond$, $\Box$ for structural control:

- Logical rules: residuation

  $$\Diamond A \rightarrow B \iff A \rightarrow \Box B$$

- Structural rules under $\Diamond$ control, for example:

  $$\Diamond A \bullet B \rightarrow B \bullet \Diamond A$$
  $$\Diamond (A \bullet B) \bullet C \rightarrow A \bullet (B \bullet C)$$
1.25. Embedding theorems: communication

\[ \mathcal{L} \vdash A \rightarrow B \iff \mathcal{L}' \bowtie \vdash A^\$ \rightarrow B^\$ \]

\( \mathcal{L} \): source logic, \( \mathcal{L}' \bowtie \): target logic.

\( ^\$ \): embedding translation,

imposing constraints if \( \mathcal{L}' \) is less discriminating than \( \mathcal{L} \), or

licensing structural relaxation if \( \mathcal{L}' \) is more discriminating than \( \mathcal{L} \).
1.26. Grammar logic: general architecture

1.26. Grammar logic: general architecture


Structure. Packages of resource-management postulates. Plug-ins w.r.t. the base logic. Structural variation:

- within languages: dimensions (form $\leadsto$ meaning)
- cross-linguistically: parametric variation
1.26. **Grammar logic: general architecture**


**Structure.** Packages of resource-management postulates. Plug-ins w.r.t. the base logic. Structural variation:

- within languages: dimensions (form $\rightarrow$ meaning)
- cross-linguistically: parametric variation

**Control.** Structural modalities for communication between resource management regimes. Lexically anchored control over structural options (rather than global choice).