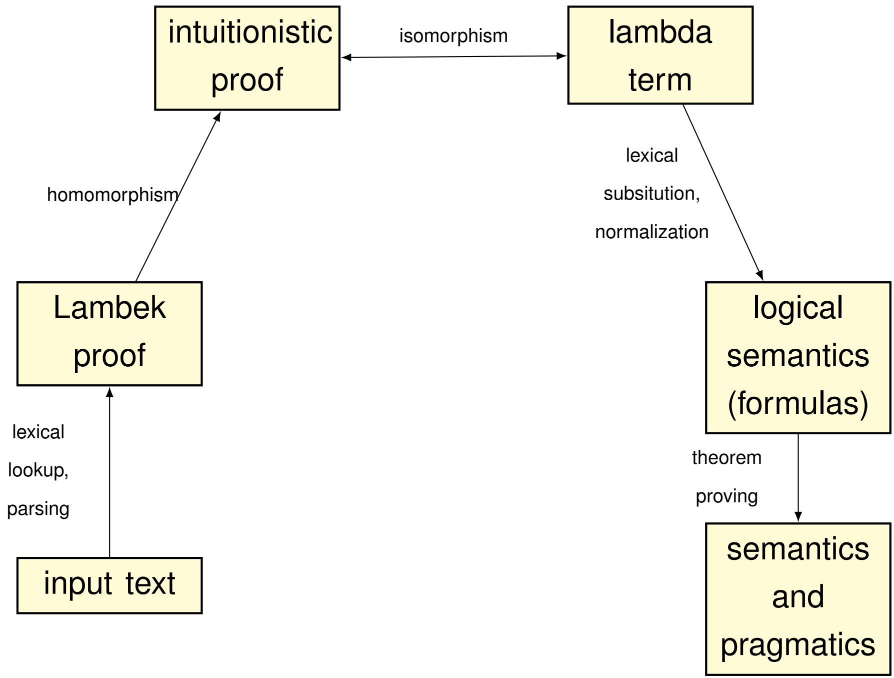
# C Montague Semantics

## C.1. Overview

- Montague Grammar and the simply typed lambda calculus (reminder)
- Curry-Howard formulas-as-types interpretation
- Montague semantics for the Lambek calculus

## C.2. Architecture



| | Introduction rules | Elimination rules |
|---|---|---|
| Intuitionistic | $\dfrac{\begin{array}{c}[A]^n \\ \vdots \\ B\end{array}}{A \to B} \to I_n$ | $\dfrac{A \quad A \to B}{B} \to E$ |
| Lambek | $\dfrac{\begin{array}{c}[A]^n \ldots \\ \vdots \\ B\end{array}}{A \backslash B} \backslash I_n$ | $\dfrac{A \quad A \backslash B}{B} \backslash E$ |
| | $\dfrac{\begin{array}{c}\ldots [A]^n \\ \vdots \\ B\end{array}}{B / A} / I_n$ | $\dfrac{B / A \quad A}{B} / E$ |

## C.3. Types and terms: Curry-Howard

A proof of $A \to B$ is a function that maps proofs of $A$ to proofs of $B$.

Think of a formula/type as the set of its proofs.

Types are.... formulae.

$\lambda$-terms encode proofs $u : U$ means $u$ is a term of type $U$.

We will also write $u : U$ as $u^U$.

## C.4. Terms: Curry-Howard

1. *hypotheses* variables of each type which are terms of this type
2. *constants* there can be constants of each type
3. *abstraction* if $x : U$ is a **variable** and $t : T$ then $(\lambda x^U.\, t) : U \to V$.
4. *application* if $f : U \to V$ and $t : U$ then $(f\ t) : V$

With such typed terms we can faithfully encode proofs.

Variables are hypotheses (that are simultaneously cancelled).

## C.5. Reduction and Normalisation

Reduction: $(\lambda x : U.\ t)^{U \to V} u^{U}$ reduces to $t[x := u] : V$.

Every simply typed lambda term reduces to a unique normal form, regardless the reduction strategy used.

## C.6. Representing formulae within lambda calculus — connectives

Assume that the base types are $\mathbf{e}$ and $\mathbf{t}$ and that the only constants are

We need the following logical constants:

| Constant | Type |
|---|---|
| $\exists$ | $(e \to t) \to t$ |
| $\forall$ | $(e \to t) \to t$ |
| $\wedge$ | $t \to (t \to t)$ |
| $\vee$ | $t \to (t \to t)$ |
| $\supset$ | $t \to (t \to t)$ |

## C.7. Representing formulae within lambda calculus — language constants

The language constants for First Order Logic (for a start):

- $R_q$ of type $\mathbf{e} \to (\mathbf{e} \to (.... \to \mathbf{e} \to \mathbf{t}))$
  e.g. likes: $e \to e \to t$, sleeps $e \to t$
- $f_q$ of type $\mathbf{e} \to (\mathbf{e} \to (.... \to \mathbf{e} \to \mathbf{e}))$

## C.8. Formulae and normal lambda terms

**Proposition 4** *A normal lambda-term of type* $\mathbf{t}$ *using only the constants given above corresponds to a formula of first-order logic.*

## C.9. Example: From formulae to normal lambda terms

$\forall x.barber(x) \supset shaves(x, x)$

$$\forall(\lambda x^{\mathbf{e}}.\ (\supset barber(x))((shaves(x))(x)))$$

Another one?

Detailed examples: a FOL formula as a term and as a natural deduction proof.

## C.10. For Montague semantics

Non normal lambda terms of type $\mathbf{t}$ coming from syntax do not really correspond to formulae.

Hence we need:

- normalisation
- a proof that the normal terms do correspond to formulae, as we just shown.

## C.11. Montague semantics. Types.

Simply typed lambda terms

$$\mathit{types} ::= e \mid t \mid \mathit{types} \to \mathit{types}$$

*chair* , *sleep* $e \to t$

*likes* transitive verb $e \to (e \to t)$

## C.12. Montague semantics: Syntax/semantics.

| (Syntactic type)$^*$ | $=$ | Semantic type | |
|---|---|---|---|
| $s^*$ | $=$ | $t$ | a sentence is a proposition |
| $np^*$ | $=$ | $e$ | a noun phrase is an entity |
| $n^*$ | $=$ | $e \to t$ | a noun is a subset of the set of entities |
| $(A\backslash B)^* = (B/A)^*$ | $=$ | $A \to B$ | extends easily to all syntactic categories of a Categorial Grammar e.g. a Lambek CG |

Logical operations (and, or, some, all the,.....) are the lambda-term constants defined above.

## C.13. Montague semantics
## Logic within lambda-calculus

Words in the lexicon need constants for their denotation:

| | | |
|---|---|---|
| *likes* | $\lambda x \lambda y\ (\mathrm{likes}\ y)\ x$ | $x : e,\ y : e, \mathrm{likes} : e \rightarrow (e \rightarrow t)$ |
| `<<` likes `>> is a two-place predicate` | | |
| *Garance* | $\lambda P\ (P\ \mathrm{Garance})$ | $P : e \rightarrow t,\ \mathrm{Garance} : e$ |
| `<<` Garance `>> is viewed as` `the properties that <<` Garance `>> holds` | | |

## C.14. Montague semantics. Computing the semantics 1/5

1. Replace in the lambda-term issued from the syntax the words by the corresponding term of the lexicon.
2. Reduce the resulting $\lambda$-term of type $t$ to obtain its normal form, which corresponds to a logical formula, the “meaning”.

| word | syntactic type $u$<br>semantic type $u^*$<br>semantics: $\lambda$-term of type $u^*$<br>$x^v$ means that the variable or constant $x$ is o |
|---|---|
| some | $(s/(np\backslash s))/n$<br>$(e \to t) \to ((e \to t) \to t)$<br>$\lambda P^{e \to t}\ \lambda Q^{e \to t}\ (\exists^{(e \to t) \to t}\ (\lambda x^{e}(\wedge^{t \to (t \to t)}(P\ x)(Q$ |
| statements | $n$<br>$e \to t$<br>$\lambda x^{e}(\mathtt{statement}^{e \to t}\ x)$ |
| speak_about | $(np\backslash s)/np$<br>$e \to (e \to t)$<br>$\lambda y^{e}\ \lambda x^{e}\ ((\mathtt{speak\_about}^{e \to (e \to t)}\ x)\ y)$ |
| themselves | $((np\backslash s)/np)\backslash(np\backslash s)$<br>$(e \to (e \to t)) \to (e \to t)$<br>$\lambda P^{e \to (e \to t)}\ \lambda x^{e}\ ((P\ x)\ x)$ |

## C.15. Syntactic proof

Let us first show that "*Some statements speak about themselves*" belongs to the language generated by this lexicon. So let us prove (in natural deduction) the following:

$$(s/(np\backslash s))/n\ ,\ n\ ,\ (np\backslash s)/np\ ,\ ((np\backslash s)/np)\backslash(np\backslash s) \vdash s$$

$$\dfrac{\dfrac{(s/(np\backslash s))/n \quad n}{(s/(np\backslash s))}\,/E \qquad \dfrac{(np\backslash s)/np \quad ((np\backslash s)/np)\backslash(np\backslash s)}{(np\backslash s)}\,\backslash E}{s}\,/E$$

## C.16. Syntactic Proof to Semantic proof

$$\dfrac{\dfrac{(s/(np\backslash s))/n \quad n}{(s/(np\backslash s))}\,/E \qquad \dfrac{(np\backslash s)/np \quad ((np\backslash s)/np)\backslash(np\backslash s)}{(np\backslash s)}\,\backslash E}{s}\,/E$$

Using the homomorphism from syntactic types to semantic types we obtain the following intuitionistic deduction.

$$\dfrac{\dfrac{(e\to t)\to(e\to t)\to t \quad e\to t}{(e\to t)\to t}\,\to E \qquad \dfrac{e\to e\to t \quad (e\to e\to t)\to e\to t}{e\to t}\,\to E}{t}\,\to E$$

## C.17. Semantic Proof to Lambda Term

$$\dfrac{\dfrac{(e \to t) \to (e \to t) \to t \quad e \to t}{(e \to t) \to t}\to E \qquad \dfrac{e \to e \to t \quad (e \to e \to t) \to e \to t}{e \to t}\to E}{t}\to E$$

$$\dfrac{\dfrac{\mathit{So}^{(e\to t)\to(e\to t)\to t} \quad \mathit{Sta}^{e\to t}}{(\mathit{So}\ \mathit{Sta})^{(e\to t)\to t}}\to E \qquad \dfrac{\mathit{SpA}^{e\to e\to t} \quad \mathit{Refl}^{(e\to e\to t)\to e\to t}}{(\mathit{Refl}\ \mathit{SpA})^{e\to t}}\to E}{((\mathit{So}\ \mathit{Sta})\ (\mathit{Refl}\ \mathit{SpA}))^{t}}\to E$$

## C.18. Montague semantics. Computing the semantics. 3/5

The syntax (e.g. a Lambek categorial grammar) yields a $\lambda$-term representing this deduction simply is

*((some statements) (themselves speak_about))* of type $t$

## C.19. Montague semantics. Computing the semantics. 4/5

$$\Big(\big(\lambda P^{e\to t}\,\lambda Q^{e\to t}\,(\exists^{(e\to t)\to t}\,(\lambda x^e(\wedge(P\,x)(Q\,x))))\big)$$
$$(\lambda x^e(\mathtt{statement}^{e\to t}\,x))\Big)$$
$$\Big(\big(\lambda P^{e\to(e\to t)}\,\lambda x^e\,((P\,x)x)\big)$$
$$(\lambda y^e\,\lambda x^e\,((\mathtt{speak\_about}^{e\to(e\to t)}\,x)y))\Big)$$

$$\downarrow\beta$$

$$(\lambda Q^{e\to t}\,(\exists^{(e\to t)\to t}\,(\lambda x^e(\wedge^{t\to(t\to t)}(\mathtt{statement}^{e\to t}\,x)(Q\,x)))))$$
$$(\lambda x^e\,((\mathtt{speak\_about}^{e\to(e\to t)}\,x)x))$$

$$\downarrow\beta$$

$$(\exists^{(e\to t)\to t}\,(\lambda x^e(\wedge(\mathtt{statement}^{e\to t}\,x)((\mathtt{speak\_about}^{e\to(e\to t)}\,x)x))))$$

## C.20. Montague semantics. Computing the semantics. 5/5

This term represent the following formula of predicate calculus (in a more pleasant format):

$$\exists x : e \; (\mathtt{statement}(x) \;\wedge\; \mathtt{speak\_about}(x, x))$$

This is a (simplistic) semantic representation of the analysed sentence.