Logic as a framework for NL semantics

- Approximate NL meaning as truth conditions.
- Logic supports precise, consistent and controlled meaning representation via truth-conditional interpretation.
- Logic provides deduction systems to model inference processes, controlled through a formal entailment concept.
- Logic supports uniform modelling of the semantic composition process.
Logic as a framework for NL semantics

- Approximate NL meaning as truth conditions.
- Logic supports precise, consistent and controlled meaning representation via truth-conditional interpretation.
- Logic provides deduction systems to model inference processes, controlled through a formal entailment concept.
- Logic supports uniform modelling of the semantic composition process.

Outline

- A reminder: First-order predicate logic (FOL).
- The limits of FOL as a formalism for semantic representations.
- Type theory.
- Modal operators in logic.
Dolphins

Dolphins are mammals, not fish.
\( \forall d \ (\text{dolphin}(d) \rightarrow \text{mammal}(d) \land \neg \text{fish}(d)) \)

Dolphins live in pods.
\( \forall d \ (\text{dolphin}(d) \rightarrow \exists x \ (\text{pod}(p) \land \text{live-in}(d, p))) \)

Dolphins give birth to one baby at a time.
\( \forall d \ (\text{dolphin}(d) \rightarrow \forall x \ \forall y \ \forall t \ (\text{give-birth-to}(d, x, t) \land \text{give-birth-to}(d, y, t) \rightarrow x=y)) \)

Syntax of FOL [1]

- Non-logical expressions:
  - Individual constants: IC
  - \( n \)-place predicate symbols: \( \text{RC}^n \ (n \geq 0) \)
- Individual variables: IV
- Terms: \( T = IV \cup IC \)
- Atomic formulas:
  - \( R(t_1, \ldots, t_n) \) for \( R \in \text{RC}^n \), if \( t_1, \ldots, t_n \in T \)
  - \( s=t \) for \( s, t \in T \)
Syntax of FOL [2]

- FOL formulas: The smallest set $For$ such that:
  - All atomic formulas are in $For$
  - If A, B are in $For$, then so are $\neg A$, $(A \land B)$, $(A \lor B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$
  - If x is an individual variable and A is in $For$, then $\forall x A$ and $\exists x A$ are in $For$.

Dolphins in FOL

*Dolphins are mammals, not fish.*
\[\forall d \ (\text{dolphin}(d) \rightarrow \text{mammal}(d) \land \neg \text{fish}(d))\]

*Dolphins live in pods.*
\[\forall d \ (\text{dolphin}(d) \rightarrow \exists x \ (\text{pod}(p) \land \text{live-in}(d,p)))\]

*Dolphins give birth to one baby at a time.*
\[\forall d \ (\text{dolphin}(d) \rightarrow \forall x \ \forall y \ \forall t \ (\text{give-birth-to}(d,x,t) \land \text{give-birth-to}(d,y,t) \rightarrow x = y))\]
Semantics of FOL [1]

- **Model structures** for FOL: $M = <U, V>$
  - $U$ (or $U_M$) is a non-empty *universe* (domain of individuals)
  - $V$ (or $V_M$) is an *interpretation function*, which assigns individuals ($\in U_M$) to individual constants and n-ary relations between individuals ($\in U_M^n$) to n-place predicate symbols.

- **Assignment function** for variables $g$: $IV \rightarrow U_M$

Semantics of FOL [2]

- Interpretation of terms (with respect to a model structure $M$ and a variable assignment $g$):
  - $[[\alpha]]^{M,g} = V_M(\alpha)$, if $\alpha$ is an individual constant
  - $[[\alpha]]^{M,g} = g(\alpha)$, if $\alpha$ is a variable
Semantics of FOL [3]

- Interpretation of formulas (with respect to model structure $M$ and variable assignment $g$):
  
  - $[[R(t_1, ..., t_n)]]^M,g = 1 \iff \langle [[t_1]]^M,g, ..., [[t_n]]^M,g \rangle \in V_M(R)$
  
  - $[[s=t]]^M,g = 1 \iff [[s]]^M,g = [[t]]^M,g$
  
  - $[[\neg \phi]]^M,g = 1 \iff [[\phi]]^M,g = 0$
  
  - $[[\phi \land \psi]]^M,g = 1 \iff [[\phi]]^M,g = 1 \text{ and } [[\psi]]^M,g = 1$
  
  - $[[\phi \lor \psi]]^M,g = 1 \iff [[\phi]]^M,g = 1 \text{ or } [[\psi]]^M,g = 1$
  
  - $[[\phi \rightarrow \psi]]^M,g = 1 \iff [[\phi]]^M,g = 0 \text{ or } [[\psi]]^M,g = 1$
  
  - $[[\phi \leftrightarrow \psi]]^M,g = 1 \iff [[\phi]]^M,g = [[\psi]]^M,g$
  
  - $[[\exists x \phi]]^M,g = 1 \iff \text{ there is } a \in U_M \text{ such that } [[\phi]]^M,g[x/a] = 1$
  
  - $[[\forall x \phi]]^M,g = 1 \iff \text{ for all } a \in U_M : [[\phi]]^M,g[x/a] = 1$

- $g[x/a]$ is the variable assignment which is identical with $g$ except that it assigns the individual $a$ to the variable $x$.

Semantics of FOL [4]

- Formula $A$ is true in the model structure $M$ iff $[[A]]^M,g = 1$ for every variable assignment $g$. This works best if $A$ has no free variables.

- A model structure $M$ satisfies a set of formulas $\Gamma$ (or: $M$ is a model of $\Gamma$) iff every formula $A \in \Gamma$ is true in $M$.

- $A$ is valid iff $A$ is true in all model structures.
- $A$ is satisfiable iff there is a model structure that makes it true.
- $A$ is unsatisfiable iff there is no model structure that makes it true.
- $A$ is contingent iff it is satisfiable but not valid.
Entailment and Deduction

• A set of formulas $\Gamma$ entails formula $A$ ($\Gamma \models A$) iff $A$ is true in every model of $\Gamma$.

• A (sound and complete) calculus for FOL allows us to prove $A$ from $\Gamma$ iff $\Gamma \models A$ by manipulating the formulas syntactically. There are many calculi for FOL: resolution, tableaux, natural deduction, ...

• Calculi can be implemented to obtain:
  – theorem provers: check entailment, validity, and unsatisfiability
  – model builders: check satisfiability, compute models
  – model checkers: determine whether model satisfies formula
  – find off-the-shelf implementations on the Internet

Two levels of interpretation

• Semantic interpretation of a NL expression in a logical framework is a two-step process:
  – The NL expression is assigned a semantic representation
  – The semantic representation is truth-conditionally interpreted.
The expressive power of FOL [1]

*John is a blond criminal*

\[
\text{criminal}(j) \land \text{blond}(j)
\]
The expressive power of FOL [1]

*John is a blond criminal*

\[ \text{criminal}(j) \land \text{blond}(j) \]

*John is an honest criminal*

\[ \text{criminal}(j) \land \text{honest}(j) \]
The expressive power of FOL [1]

*John is a blond criminal*

\[ \text{criminal}(j) \land \text{blond}(j) \]

*John is an honest criminal*

\[ \text{criminal}(j) \land \text{honest}(j) \]

*John is an alleged criminal*

\[ \text{criminal}(j) \land \text{alleged}(j) \]
The expressive power of FOL [2]

*John is driving fast*

\[ \text{drive}(j) \land \text{fast}(j) \]
The expressive power of FOL [2]

*John is driving fast*

\[ \text{drive}(j) \land \text{fast}(j) \]

*John is eating fast*

\[ \text{eat}(j) \land \text{fast}(j) \]
The expressive power of FOL [2]

*John is driving fast*

\[
\text{drive}(j) \land \text{fast}(j)
\]

*John is eating fast*

\[
\text{eat}(j) \land \text{fast}(j) \quad ??
\]

*John is driving very fast.*

???
The expressive power of FOL [3]

*It rains.*
*It rained yesterday.*
*It rains occasionally.*

*Bill is blond. Blond is a hair colour. (≠ Bill is a hair colour.)*

Type theory

- The types of non-logical expressions provided by FOL – terms and n-ary first-order relations – are not sufficient to describe the semantic function of all natural language expressions.
- Type theory provides a much richer inventory of types – higher-order relations and functions of different kinds.
Types

- For NL meaning representation the (minimal) set of basic types is \{e, t\}:
  - e (for entity) is the type of individual terms
  - t (for truth value) is the type of formulas
- All pairs <σ, τ> made up of (basic or complex) types σ, τ are types. <σ, τ> is the type of functions which map arguments of type σ to values of type τ.
- In short: The set of types is the smallest set T such that e, t ∈ T, and if σ, τ ∈ T, then also <σ, τ> ∈ T.

Some useful complex types for NL semantics

- Individual: e
- Sentence: t
- One-place predicate constant: <e, t>
- Two-place relation: <e, e, t>
- Sentence adverbial: <t, t>
- Attributive adjective: <<e, t>, <e, t>>
- Degree modifier: <<<e, t>, <e, t>>, <e, t>, <e, t>>
Second-order predicates

- *Bill is blond. Blond is a hair colour:*
  - Bill is represented as a term of type e.
  - ”blond” is represented as a term of type <e,t>.
  - ”hair colour” is represented as a term of type <<e,t>,t>.
  - ”Bill is a hair colour” is not even a well-formed statement.

Some useful complex types for NL semantics

- Individual: e
- Sentence: t
- One-place predicate constant: <e,t>
- Two-place relation: <e,<e,t>>
- Sentence adverbial: <t,t>
- Attributive adjective: <<e,t>,<e,t>>
- Degree modifier: <<<e,t>,<e,t>>,<<e,t>,<e,t>>>
- Second-order predicate: <<e,t>,t>
Type-theoretic syntax [1]

• Vocabulary:
  – Possibly empty, pairwise disjoint sets of non-logical constants: $\text{Con}_\tau$ for every type $\tau$

Higher-order variables

• *Bill has the same hair colour as John.*

• *Santa Claus has all the attributes of a sadist.*
Type-theoretic syntax [1]

• Vocabulary:
  – Possibly empty, pairwise disjoint sets of non-logical constants: $\text{Con}_\tau$ for every type $\tau$
  – Infinite and pairwise disjoint sets of variables: $\text{Var}_\tau$ for every type $\tau$
  – The logical operators known from FOL.

Type-theoretic syntax [2]

• The sets of well-formed expressions $\text{WE}_\tau$ for every type $\tau$ are given by:
  – $\text{Con}_\tau \subseteq \text{WE}_\tau$ for every type $\tau$
  – If $\alpha \in \text{WE}_{<\sigma, \tau_1}$, $\beta \in \text{WE}_\sigma$, then $\alpha(\beta) \in \text{WE}_\tau$.
  – If $A, B$ are in $\text{WE}_1$, then so are $\neg A$, $(A \land B)$, $(A \lor B)$, $(A \rightarrow B), (A \leftarrow B)$
  – If $A$ is in $\text{WE}_1$, then so are $\forall v A$ and $\exists v A$, where $v$ is a variable of arbitrary type.
  – If $\alpha, \beta$ are well-formed expressions of the same type, then $\alpha = \beta \in \text{WE}_1$. 
Building well-formed expressions

*Bill drives fast.*

\[
\begin{align*}
\text{drive: } &<e,t> \quad \text{fast: } <<e,t>,<e,t>> \\
\text{Bill: } &e \quad \text{fast(drive): } <e,t> \\
\text{fast(drive)(bill): } &t
\end{align*}
\]

*Mary works in Saarbrücken*

\[
\begin{align*}
\text{mary: } &e \quad \text{work: } <e,t> \quad \text{in: } <<e,<t,t>>, sb: e \\
\text{work(mary): } &t \quad \text{in(sb): } <t,t> \\
\text{in(sb)(work(mary)): } &t
\end{align*}
\]

More examples

- *Blond is a hair colour.*

- *Santa Claus has all the attributes of a sadist.*
Type-theoretic semantics [1]

- Let $U$ be a non-empty set of entities.
- The **domain of possible denotations** $D_\tau$ for every type $\tau$ is given by:
  - $D_e = U$
  - $D_I = \{0, 1\}$
  - $D_{<\sigma, \tau>}$ is the set of all functions from $D_\sigma$ to $D_\tau$

Type-theoretic semantics [2]

- A **model structure** for a type theoretic language:
  $M = \langle U, V \rangle$, where
  - $U$ (or $U_M$) is a non-empty domain of individuals
  - $V$ (or $V_M$) is an interpretation function, which assigns to every member of $\text{Con}_\tau$ an element of $D_\tau$.
- **Variable assignment** $g$ assigns every variable of type $\tau$ a member of $D_\tau$. 
Type-theoretic semantics [3]

Interpretation (with respect to model structure $M$ and variable assignment $g$):

$[[\alpha]]^M_g = V_M(\alpha)$, if $\alpha$ constant

$[[\alpha]]^M_g = g(\alpha)$, if $\alpha$ variable

$[[\alpha(\beta)]]^M_g = [[\alpha]]^M_g([[\beta]]^M_g)$

$[[\neg \varphi]]^M_g = 1$ iff $[[\varphi]]^M_g = 0$

$[[\varphi \land \psi]]^M_g = 1$ iff $[[\varphi]]^M_g = 1$ and $[[\psi]]^M_g = 1$, etc.

If $v \in \text{Var}_\tau$, $[[\exists v \varphi]]^M_g = 1$ iff there is $a \in D_\tau$ such that $[[\varphi]]^M_g[v/a] = 1$

If $v \in \text{Var}_\tau$, $[[\forall v \varphi]]^M_g = 1$ iff for all $a \in D_\tau : [[\varphi]]^M_g[v/a] = 1$

$[[\alpha = \beta]]^M_g = 1$ iff $[[\alpha]]^M_g = [[\beta]]^M_g$

Type theory

- The definition of the syntax and semantics of type theory is a straightforward extension of FOL.
- Words like "satisfies", "valid", "satisfiable", "entailment" carry over almost verbatim from FOL.
- Type theory is sometimes called "higher-order logic":
  - first-order logic allows quantification over individual variables (type $e$)
  - second-order logic allows quantification over variables of type $<\sigma, \tau>$ where $\sigma$ and $\tau$ are atomic
  - ....
Currying

- All functional types are interpreted as one-place functions.
- How do we deal with functions/relations with multiple arguments?
- Currying ("Schönfinkeln"):
  - simulate term $P(a,b)$ as the term $P(a)(b)$
  - simulate type $<e \times e, t>$ as the type $<e, <e, t>>$.

Summary

- First-order logic is nice, but its expressive power has limits that are not acceptable in NL semantics:
  - modification
  - modification of modifiers
  - higher-order properties
- Type theory is a generalisation of first-order logic that allows us to represent the semantics of all these expressions.