Introduction to Logic 15

Last time:

- Interpretations
- Formalization

This time:

- Interpretations and Assignments
- The Meaning of Terms
- The Meaning of Formulas
  - Atomic Formulas
  - Compound Formulas

Interpretations and Assignments

- We have introduced the notion of an **interpretation**

$$\mathcal{I} = (D, I)$$

The interpretation $\mathcal{I}$ fixes two things:

1. the domain of interpretation D; and
2. the interpretation function I for constants, function symbols and predicate symbols.

- A **variable assignment** g associates individual variables with elements of the domain of intepretations.
- There are general rules for calculating the meaning of compound expressions:
  1. terms
  2. formulas

Terms

Definition *Let $\mathcal{I} = (D, I)$ be an interpretation and g a variable assignment function. The meaning $[\![t]\!]$ of a term t (w.r.t. $\mathcal{I}$ and g) is given by:*

1. *if t is an individual constant a, then* $[\![t]\!] = I(a)$;
2. *if t is an individual variable x, then* $[\![t]\!] = g(x)$;
3. *if t is a functional term $f(t_1, t_2, \ldots, t_n)$, then* $[\![t]\!] = I(f)([\![t_1]\!], [\![t_2]\!], \ldots, [\![t_n]\!])$

Example (The Integer domain)

Domain:

$D = \{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots\}$

Interpretation Function:

Notation	interpreted as	Denotation
$I(z)$	$=$	0
$I(p)$	$=$	*predecessor*
$I(s)$	$=$	*successor*

- So now:

$$
\begin{aligned}
[\![s(p(z))]\!] &= \\
I(s)([\![p(z)]\!]) &= \\
I(s)(I(p)([\![z]\!])) &= \\
I(s)(I(p)(I(z))) &= \\
\mathit{successor}(\mathit{predecessor}(0)) &= \\
\mathit{successor}(-1) &= \\
0
\end{aligned}
$$

Formulas

- It remains to provide an interpretation for formulas of the predicate calculus.

  i.e. given an interpretation $\mathcal{I} = (D, I)$ and variable assignment g, we must provide:

  1. a way of calulating the truth values of atomic formulas

     $$P(t_1, \ldots, t_n)$$

  2. rules for determining the truth values of compound formulas:
     - connectives: $\wedge$, $\vee$, $\neg$, $\rightarrow$ and $\leftrightarrow$
     - quantifiers: $\forall$ and $\exists$

Atomic Formulas

- Atomic formulas are built from predicate symbols and terms.
- In general, if P is a predicate symbol of arity n, and $t_1, \ldots, t_n$ are n terms, then $P(t_1, \ldots, t_n)$ is an atomic formula.
- $P(t_1, \ldots, t_n)$ can be understood as stating that the individuals picked out by $t_1, \ldots, t_n$ stand in the n-place relation denoted by P.

Definition *Let $\mathcal{I} = (D, I)$ be an interpretation, and g a variable assignment. An atomic formula $P(t_1, \ldots, t_n)$ is* **true** *(w.r.t. $\mathcal{I}$ and g) if*

$$\langle [\![t_1]\!], \ldots, [\![t_n]\!] \rangle \in I(P)$$

and otherwise it is **false** .

Example (Integers)

$D = \{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots\}$

Notation	interpreted as	Denotation
$I(z)$	$=$	0
$I(p)$	$=$	*predecessor*
$I(s)$	$=$	*successor*
$I(L)$	$=$	*less-than*

- Consider:

$$L(p(z), z)$$

- This is **true** (w.r.t. this interpretation) since:

$$\langle [\![p(z)]\!], [\![z]\!] \rangle = \langle -1, 0 \rangle$$

and

$$\langle -1, 0 \rangle \in \textit{less-than}$$

Compound Formulas – Connectives –

- We now have a way of assigning truth values to atomic formulas.
- Atomic formulas may be combined with the connectives to build compound formulas.
  - Must give rules for calculating truth values for such formulas...
  - ... this is easy! We already know what the connectives mean.
- The rules for compound formulas built with connectives just follow those for propositional logic.

Definition *For arbitrary formulas A and B, then with respect to a given interpretation $\mathcal{I}$ and variable assignment g, we have:*

1. $\neg A$ *is* **true** *iff* A *is* **false**
2. $(A \wedge B)$ *is* **true** *iff* A *is* **true** *and* B *is* **true**
3. $(A \vee B)$ *is* **true** *iff* A *is* **true** *or* B *is* **true**
4. $(A \rightarrow B)$ *is* **true** *iff* A *is* **false** *and* B *is* **true**
5. $(A \leftrightarrow B)$ *is* **true** *iff* A *is* **true** *and* B *is* **true** *, or* A *is* **false** *and* B *is* **false**

Compound Formulas
– Quantification –

- Compound formulas can also be formed using the quantifiers $\forall$ and $\exists$.
- In general, if A is a formula, and v is a variable then:
  - $\forall v.A$ is a formula;

    $$\forall x.(L(x, z) \rightarrow L(p(x), z))$$

  - $\exists v.A$ is a formula.

    $$\exists x.L(x, z)$$

- How can we formalize the interpretation of such statements?

Definition *Let $\mathcal{I} = (D, I)$ be an interpretation and g a variable assignment. Then w.r.t. $\mathcal{I}$ and g we have that:*

1. $\forall v.A$ *is* **true** *iff* A *is* **true** *whatever the value of the variable* v*;*
2. $\exists v.A$ *is* **true** *iff* A *is* **true** *for at least one possible value of the variable* v*;*

Example (Integers once again)

- Consider the statement

$$\forall x.(L(x, z) \rightarrow L(p(x), z))$$

- From the definition, this is **true** as long as

$$L(x, z) \rightarrow L(p(x), z)$$

is **true** for whatever value we pick for x.

Summary

- An interpretation for a first-order language fixes the domain of interpretation and the meaning of basic expressions.
- General rules can be given for calculating the meaning of compound expressions (both terms and formulas).
- Terms denote individuals.
- Atomic formulas state that relations hold between individuals.
- The rules for calculating the truth values of boolean combinations of formulas are just as for propositional logic.
- Quantified formulas are interpreted as statements about all (some) possible values of the quantified variable.