LING4400: Lecture Notes 4
Predicates and operators

Contents

So far we looked at a general framework for logic and functions on propositions (truth values).

Today we'll look at some functions that define a fairly rich kind of logic which is widely used.

4.1 Predicates

Functions that map one or more entities to truth values are called **predicates**.

They have a variety of types: $\langle e,t\rangle, \langle e,\langle e,t\rangle\rangle, \langle e,\langle e,\langle e,t\rangle\rangle\rangle, \ldots$

Their function tables depend on the number of inputs they have:

1. **Properties** or **unary predicates** (with one argument) look like this:

$$[\![\mathsf{Coastal}]\!]^M = \begin{array}{|ll|}\hline \text{input} & \text{output} \\ \hline \textbf{Laos}: & \textbf{False} \\ \textbf{Mali}: & \textbf{False} \\ \textbf{Togo}: & \textbf{True} \\ \hline \end{array}$$

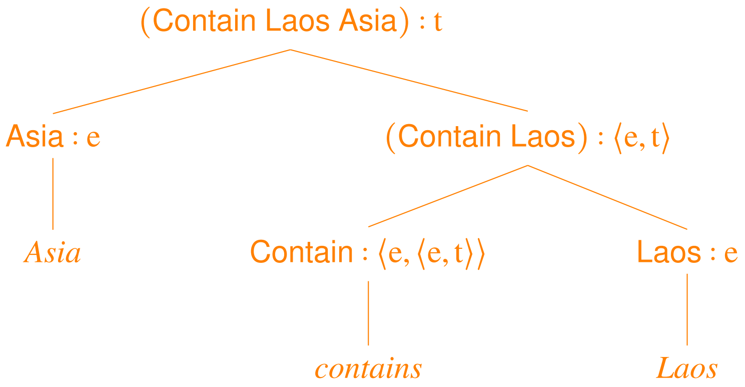
   They can also be called **characteristic functions** or **sets**.

2. **Relations** or **binary predicates** (with two arguments) look like this:

$$[\![\mathsf{Contain}]\!]^M = \begin{array}{|ll|}\hline \text{input} & \text{output} \\ \hline \vdots & \vdots \\ \textbf{Laos}: & \begin{array}{|ll|}\hline \text{input} & \text{output} \\ \hline \textbf{Africa}: & \textbf{False} \\ \textbf{Asia}: & \textbf{True} \\ \vdots & \vdots \\ \hline \end{array} \\ \textbf{Mali}: & \begin{array}{|ll|}\hline \text{input} & \text{output} \\ \hline \textbf{Africa}: & \textbf{True} \\ \textbf{Asia}: & \textbf{False} \\ \vdots & \vdots \\ \hline \end{array} \\ \textbf{Togo}: & \begin{array}{|ll|}\hline \text{input} & \text{output} \\ \hline \textbf{Africa}: & \textbf{True} \\ \textbf{Asia}: & \textbf{False} \\ \vdots & \vdots \\ \hline \end{array} \\ \hline \end{array}$$

We sometimes order the arguments backward to make function applications align with syntax.

Composing from the bottom up, direct objects or preposition complements compose first:



This is an **isomorphism**: a mapping (from syntax to semantics) preserving relations (structure).

Later we'll see this helps when phrases are conjoined: *Africa* [*contains Mali and contains Togo*].

Some common relations of type $\langle e, \langle e, t \rangle\rangle$ (actually some of them are 'polymorphic' $\langle \alpha, \langle \alpha, t \rangle\rangle$):

1. **equality** (infix):

$$[\![\mathsf{Equal}\ \varphi\ \psi]\!]^M = [\![\varphi = \psi]\!]^M \text{ holds if and only if } [\![\varphi]\!]^M = [\![\psi]\!]^M$$

   for example:

$$[\![\mathsf{Equal\ Mali\ Togo}]\!]^M = [\![\mathsf{Mali} = \mathsf{Togo}]\!]^M = \textbf{False}$$

2. **inequality** (infix):

$$[\![\mathsf{Not}\ (\mathsf{Equal}\ \varphi\ \psi)]\!]^M = [\![\varphi \neq \psi]\!]^M \text{ holds if and only if } [\![\varphi]\!]^M \neq [\![\psi]\!]^M$$

   for example:

$$[\![\mathsf{Not\ (Equal\ Mali\ Togo)}]\!]^M = [\![\mathsf{Mali} \neq \mathsf{Togo}]\!]^M = \textbf{True}$$

3. **less than** (infix, of numbers):

$$[\![\mathsf{LessThan}\ \varphi\ \psi]\!]^M = [\![\varphi < \psi]\!]^M \text{ holds if and only if } [\![\varphi]\!]^M < [\![\psi]\!]^M$$

   for example:

$$[\![\mathsf{LessThan}\ 2\ 3]\!]^M = [\![2 < 3]\!]^M = \textbf{True}$$

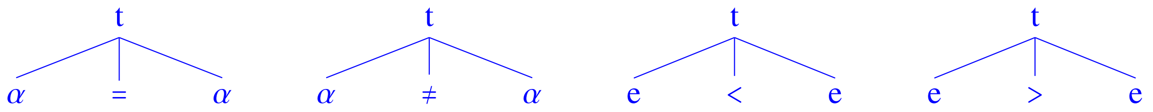
4. **greater than** (infix, of numbers):

$$[\![\mathsf{GreaterThan}\ \varphi\ \psi]\!]^M = [\![\varphi > \psi]\!]^M \text{ holds if and only if } [\![\varphi]\!]^M > [\![\psi]\!]^M$$

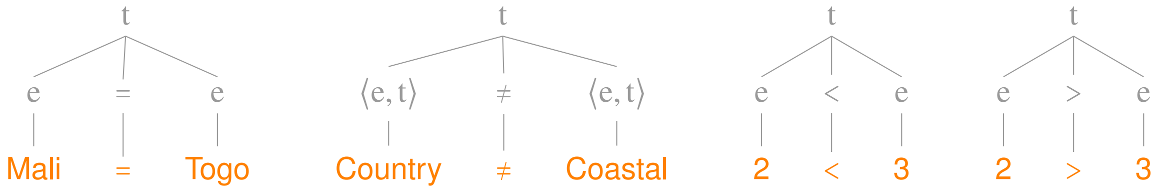
   for example:

$$[\![\mathsf{GreaterThan}\ 2\ 3]\!]^M = [\![2 > 3]\!]^M = \textbf{False}$$

Again, we can draw trees for these expressions in infix notation using flattened rules:



For example:



4.2 Operator functions

Logical expressions can also contain **operators** of type $\langle e, \langle e, e\rangle\rangle$:

1. **addition** (infix, of numbers):

$$[\![\text{Sum } \varphi\ \psi]\!]^M = [\![\varphi + \psi]\!]^M = [\![\varphi]\!]^M + [\![\psi]\!]^M$$

   for example:

$$[\![\text{Sum 2 3}]\!]^M = [\![2 + 3]\!]^M = \mathbf{5}$$

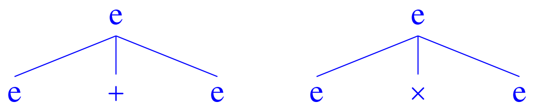
2. **multiplication** (infix, of numbers):

$$[\![\text{Prod } \varphi\ \psi]\!]^M = [\![\varphi \times \psi]\!]^M = [\![\varphi]\!]^M \times [\![\psi]\!]^M$$

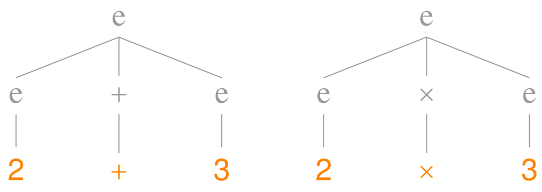
   for example:

$$[\![\text{Prod 2 3}]\!]^M = [\![2 \times 3]\!]^M = \mathbf{6}$$

Again, we can draw trees for these expressions in infix notation using flattened rules:



For example:


e
e
+
e
2
+
3
e
e
×
e
2
×
3