Mathematical Logic and Linguistics

Glyn Morrill & Oriol Valentín

Department of Computer Science
Universitat Politècnica de Catalunya
morrill@cs.upc.edu & oriol.valentin@gmail.com

BGSMath Course
Class 5

Semantic types

Recall the following operations on sets:

(1) a. Functional exponentiation:
X^Y = the set of all total functions from Y to X

b. Cartesian product: $X \times Y = \{\langle x, y\rangle |\ x \in X \ \&\ y \in Y\}$

c. Disjoint union: $X \uplus Y = (\{1\} \times X) \cup (\{2\} \times Y)$

d. n-th Cross product, $n \in \mathcal{N}$:

$$X^0 = \{0\}$$
$$X^{1+n} = X \times (X^n)$$

The set $\mathcal{T}$ of *semantic types* of the semantic representation language is defined on the basis of a set δ of *basic semantic types* as follows:

(2) $\mathcal{T} ::= \delta \mid \top \mid \mathcal{T}+\mathcal{T} \mid \mathcal{T}\&\mathcal{T} \mid \mathcal{T}\rightarrow\mathcal{T} \mid \mathbf{M}\mathcal{T} \mid \mathbf{L}\mathcal{T} \mid \mathcal{T}^{+}$

The set $\mathcal{T}$ of *semantic types* of the semantic representation language is defined on the basis of a set δ of *basic semantic types* as follows:

(4) $\mathcal{T} ::= \delta \mid \top \mid \mathcal{T}+\mathcal{T} \mid \mathcal{T}\&\mathcal{T} \mid \mathcal{T}\rightarrow\mathcal{T} \mid \mathbf{M}\mathcal{T} \mid \mathbf{L}\mathcal{T} \mid \mathcal{T}^{+}$

A *semantic frame* comprises a family $\{D_\tau\}_{\tau\in\delta}$ of non-empty *basic type domains* and a nonempty set W of worlds. This induces a nonempty *type domain* D_τ for each type τ as follows:

(5)
$$
\begin{array}{rcl}
D_\top & = & \{\emptyset\} \\
D_{\tau_1+\tau_2} & = & D_{\tau_1} \uplus D_{\tau_2} \\
D_{\tau_1\&\tau_2} & = & D_{\tau_1} \times D_{\tau_2} \\
D_{\tau_1\rightarrow\tau_2} & = & D_{\tau_2}^{D_{\tau_1}} \\
D_{\mathbf{M}\tau} & = & W \times D_\tau \\
D_{\mathbf{L}\tau} & = & D_\tau^W \\
D_{\tau^+} & = & \bigcup_{n>0}(D_\tau)^n
\end{array}
$$

Semantic Representation Language

The sets Φ_τ of *terms* of type τ for each semantic type τ are defined on the basis of sets C_τ of constants of type τ and denumerably infinite sets V_τ of variables of type τ for each type τ as follows:

(6)

Φ_τ	$::=$	C_τ	constants
Φ_τ	$::=$	V_τ	variables
$\Phi_\top$	$::=$	0	
Φ_τ	$::=$	$\Phi_{\tau_1+\tau_2}\text{->}V_{\tau_1}.\Phi_\tau;\ V_{\tau_2}.\Phi_\tau$	case statement
$\Phi_{\tau+\tau'}$	$::=$	$\iota_1\Phi_\tau$	first injection
$\Phi_{\tau'+\tau}$	$::=$	$\iota_2\Phi_\tau$	second injection
Φ_τ	$::=$	$\pi_1\Phi_{\tau\&\tau'}$	first projection
Φ_τ	$::=$	$\pi_2\Phi_{\tau'\&\tau}$	second projection
$\Phi_{\tau\&\tau'}$	$::=$	$(\Phi_\tau, \Phi_{\tau'})$	ordered pair formation
Φ_τ	$::=$	$(\Phi_{\tau'\to\tau}\ \Phi_{\tau'})$	functional application
$\Phi_{\tau\to\tau'}$	$::=$	$\lambda V_\tau \Phi_{\tau'}$	functional abstraction
Φ_τ	$::=$	${}^\vee\Phi_{\mathbf{L}\tau}$	extensionalization
$\Phi_{\mathbf{L}\tau}$	$::=$	${}^\wedge\Phi_\tau$	intensionalization
Φ_τ	$::=$	${}^\cup\Phi_{\mathbf{M}\tau}$	projection
$\Phi_{\mathbf{M}\tau}$	$::=$	${}^\cap\Phi_\tau$	injection
Φ_{τ^+}	$::=$	$[\Phi_\tau] \mid [\Phi_\tau \vert \Phi_{\tau^+}]$	non-empty list construction

Given a semantic frame, a *valuation f* mapping each constant of type τ into an element of D_τ, an assignment g mapping each variable of type τ into an element of D_τ, and a world $i \in W$, each term ϕ of type τ receives an interpretation $[\phi]^{g,i} \in D_\tau$ as shown below;

Given a semantic frame, a *valuation* f mapping each constant of type τ into an element of D_τ, an assignment g mapping each variable of type τ into an element of D_τ, and a world $i \in W$, each term ϕ of type τ receives an interpretation $[\phi]^{g,i} \in D_\tau$ as shown below; the *update* $g[x := m]$ is $(g - \{(x, g(x))\}) \cup \{(x, m)\}$, i.e. the function which sends x to m and agrees with g elsewhere.

$$
\begin{array}{rcl}
[a]^{g,i} & = & f(a) \text{ for constant } a \in C_\tau \\
[x]^{g,i} & = & g(x) \text{ for variable } x \in V_\tau \\
[0]^{g,i} & = & \emptyset \\
[\phi\text{->}x.\psi; y.\chi]^{g,i} & = & \begin{cases} [\psi]^{g[x:=\mathbf{snd}([\phi]^{g,i})],i} & \text{if } \mathbf{fst}([\phi]^{g,i}) = 1 \\ [\chi]^{g[y:=\mathbf{snd}([\phi]^{g,i})],i} & \text{if } \mathbf{fst}([\phi]^{g,i}) = 2 \end{cases} \\
[\iota_1\phi]^{g,i} & = & \langle 1, [\phi]^{g,i} \rangle \\
[\iota_2\phi]^{g,i} & = & \langle 2, [\phi]^{g,i} \rangle \\
[\pi_1\phi]^{g,i} & = & \mathbf{fst}([\phi]^{g,i}) \\
[\pi_2\phi]^{g,i} & = & \mathbf{snd}([\phi]^{g,i}) \\
[(\phi, \psi)]^{g,i} & = & \langle [\phi]^{g,i}, [\psi]^{g,i} \rangle \\
[(\phi\, \psi)]^{g,i} & = & [\phi]^{g,i}([\psi]^{g,i}) \\
[\lambda x \phi]^{g,i} & = & d \mapsto [\phi]^{g[x:=d],i} \\
[{}^\vee\phi]^{g,i} & = & [\phi]^{g,i}(i) \\
[{}^\wedge\phi]^{g,i} & = & j \mapsto [\phi]^{g,j} \\
[{}^\cup\phi]^{g,i} & = & \mathbf{snd}([\phi]^{g,i}) \\
[{}^\cap\phi]^{g,i} & = & \langle i, [\phi]^{g,i} \rangle \\
[[\phi]]^{g,i} & = & \langle [\phi]^{g,i}, 0 \rangle \\
[[\phi|\psi]]^{g,i} & = & \langle [\phi]^{g,i}, [\psi]^{g,i} \rangle
\end{array}
$$

In $x.\phi$, $\lambda x\phi$ or $^\wedge\phi$, ϕ is the *scope* of $x.$, λx or $^\wedge$.

In $x.\phi$, $\lambda x\phi$ or $^\wedge\phi$, ϕ is the *scope* of $x.$, λx or $^\wedge$.

An occurrence of a variable x in a term is called *free* if and only if it does not fall within the scope of any $x.$ or λx; otherwise it is *bound* (by the closest $x.$ or λx within the scope of which it falls).

In $x.\phi$, $\lambda x \phi$ or $^\wedge\phi$, ϕ is the *scope* of $x.$, λx or $^\wedge$.

An occurrence of a variable x in a term is called *free* if and only if it does not fall within the scope of any $x.$ or λx; otherwise it is *bound* (by the closest $x.$ or λx within the scope of which it falls).

The result $\phi\{\psi_1/x_1, \ldots, \psi_n/x_n\}$ of substituting terms $\psi_1, \ldots, \psi_n$ for variables $x_1, \ldots, x_n$ of the same types respectively in a term ϕ is the result of simultaneously replacing by ψ_i every free occurrence of x_i in ϕ.

In $x.\phi$, $\lambda x\phi$ or $^\wedge\phi$, ϕ is the *scope* of $x.$, λx or $^\wedge$.

An occurrence of a variable x in a term is called *free* if and only if it does not fall within the scope of any $x.$ or λx; otherwise it is *bound* (by the closest $x.$ or λx within the scope of which it falls).

The result $\phi\{\psi_1/x_1, \ldots, \psi_n/x_n\}$ of substituting terms $\psi_1, \ldots, \psi_n$ for variables $x_1, \ldots, x_n$ of the same types respectively in a term ϕ is the result of simultaneously replacing by ψ_i every free occurrence of x_i in ϕ.

We say that ψ is *free for x in ϕ* if and only if no variable in ψ becomes bound in $\phi\{\psi/x\}$.

We say that a term is *modally closed* if and only if every occurrence of $^{\vee}$ occurs within the scope of an $^{\wedge}$.

We say that a term is *modally closed* if and only if every occurrence of $^\vee$ occurs within the scope of an $^\wedge$.

A modally closed term is denotationally invariant across worlds.

We say that a term is *modally closed* if and only if every occurrence of $^\vee$ occurs within the scope of an $^\wedge$.

A modally closed term is denotationally invariant across worlds.

We say that a term ψ is *modally free for x in* ϕ if and only if either ψ is modally closed, or no free occurrence of *x* in ϕ is within the scope of an $^\wedge$.

We say that a term is *modally closed* if and only if every occurrence of $^\vee$ occurs within the scope of an $^\wedge$.

A modally closed term is denotationally invariant across worlds.

We say that a term ψ is *modally free for x in* ϕ if and only if either ψ is modally closed, or no free occurrence of *x* in ϕ is within the scope of an $^\wedge$.

There are the following laws of conversion.

$$\phi\text{->}y.\psi; z.\chi = \phi\text{->}x.(\psi\{x/y\}); z.\chi$$
if x is not free in ψ and is free for y in ψ
$$\phi\text{->}y.\psi; z.\chi = \phi\text{->}y.\psi; x.(\chi\{x/z\})$$
if x is not free in χ and is free for z in χ
$$\lambda y\phi = \lambda x(\phi\{x/y\})$$
if x is not free in ϕ and is free for y in ϕ

α-conversion

$$\iota_1\phi\text{->}y.\psi; z.\chi = \psi\{\phi/y\}$$
if ϕ is free for y in ψ and modally free for y in ψ
$$\iota_2\phi\text{->}y.\psi; z.\chi = \chi\{\phi/z\}$$
if ϕ is free for z in χ and modally free for z in χ
$$\pi_1(\phi, \psi) = \phi$$
$$\pi_2(\phi, \psi) = \psi$$
$$(\lambda x\phi\,\psi) = \phi\{\psi/x\}$$
if ψ is free for x in ϕ, and modally free for x in ϕ
$$^{\vee\wedge}\phi = \phi$$
$$^{\cup\cap}\phi = \phi$$

β-conversion

$$(\pi_1\phi, \pi_2\phi) = \phi$$
$$\lambda x(\phi\, x) = \phi$$
if x is not free in ϕ
$$^{\wedge\vee}\phi = \phi$$
if ϕ is modally closed
$$^{\cap\cup}\phi = \phi$$

η-conversion

For completeness, the so-called commuting conversions for the case statement are thus:

$$
\begin{aligned}
\phi{-}{>}x.\iota_1\psi; y.\iota_1\chi &= \iota_1(\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.\iota_2\psi; y.\iota_2\chi &= \iota_2(\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.\pi_1\psi; y.\iota_1\chi &= \pi_1(\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.\pi_2\psi; y.\iota_2\chi &= \pi_2(\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.(\delta,\psi); y.(\delta,\chi) &= (\delta,\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.(\psi,\delta); y.(\chi,\delta) &= (\phi{-}{>}x.\psi; y.\chi,\delta)\\
\phi{-}{>}x.(\delta\,\psi); y.(\delta\,\chi) &= (\delta\,\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.(\psi\,\delta); y.(\chi\,\delta) &= (\phi{-}{>}x.\psi; y.\chi\,\delta)
\end{aligned}
$$

$$\phi{-}{>}x.\lambda z\psi; y.\lambda z\chi = \lambda z(\phi{-}{>}x.\psi; y.\chi)$$

if z is not free in ϕ

$$\phi{-}{>}x.{}^{\vee}\psi; y.{}^{\vee}\chi = {}^{\vee}(\phi{-}{>}x.\psi; y.\chi)$$

$$\phi{-}{>}x.{}^{\wedge}\psi; y.{}^{\wedge}\chi = {}^{\wedge}(\phi{-}{>}x.\psi; y.\chi)$$

if ϕ is modally closed

$$
\begin{aligned}
\phi{-}{>}x.{}^{\cup}\psi; y.{}^{\cup}\chi &= {}^{\cup}(\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.{}^{\cap}\psi; y.{}^{\cap}\chi &= {}^{\cap}(\phi{-}{>}x.\psi; y.\chi)\\
\phi{-}{>}x.[\delta|\psi]; y.[\delta|\chi] &= [\delta|\phi{-}{>}x.\psi; y.\chi]\\
\phi{-}{>}x.[\psi|\delta]; y.[\chi|\delta] &= [\phi{-}{>}x.\psi; y.\chi|\delta]
\end{aligned}
$$