Cours de Typage
Master 2 : Langages et Programmation

Delia Kesner et Giovanni Bernardi
IRIF, Université Paris-Diderot

Emails : kesner@irif.fr et bernargi@tcd.ie
URLs : http://www.irif.fr/~kesner et http://www.irif.fr/~gio

Modalités

- Cours/TD Mercredi 10h-11h30.
- TP Mercredi 11h30-13h00.
- Un projet.
- Un examen final.

Plan de la première partie du cours

1. Sémantique opérationnelle d'un mini-langage fonctionnel
2. Types monomorphes
3. Types polymorphes

Notes de cours

Consulter http://www.irif.fr/~kesner regulièrement.

Operational Semantics

Defining an Operational Semantics

- Granurality
- Order of evaluation

Big-step Semantics

Each rule completely evaluates the expression to a value.

$$\frac{}{\langle n, \sigma \rangle \Downarrow n} \qquad \frac{}{\langle X, \sigma \rangle \Downarrow \sigma(X)}$$

$$\frac{\langle a_1, \sigma \rangle \Downarrow n_1 \qquad \langle a_2, \sigma \rangle \Downarrow n_2 \qquad n \text{ is "} n_1 \text{ plus } n_2 \text{"}}{\langle a_1 + a_2, \sigma \rangle \Downarrow n}$$

Properties

- Abstract
- Allows to avoid details
- No specification of evaluation order (e.g. $(1+3)+(5-3)$)
- No specification of control of errors
- No specification of interleaving

Small-step Semantics

Evaluation is given by a sequence of *state changes* of an abstract machine which terminates when the state cannot be reduced further.

$$\frac{\langle a_1, \sigma\rangle \leadsto \langle a_1', \sigma'\rangle}{\langle a_1 + a_2, \sigma\rangle \leadsto \langle a_1' + a_2, \sigma'\rangle} \qquad \frac{\langle a_2, \sigma\rangle \leadsto \langle a_2', \sigma'\rangle}{\langle n_1 + a_2, \sigma\rangle \leadsto \langle n_1 + a_2', \sigma'\rangle}$$

$$\frac{}{\langle X, \sigma\rangle \leadsto \langle \sigma(X), \sigma\rangle} \qquad \frac{n \text{ is "} n_1 \text{ plus } n_2\text{"}}{\langle n_1 + n_2, \sigma\rangle \leadsto \langle n, \sigma\rangle}$$

Properties

- Less abstract
- Specification of order of evaluation
- Control of errors : $\dfrac{n_2 \neq 0}{n_1 / n_2 \rightsquigarrow n}$, where n is "n_1 divided by n_2".
- Interleaving : $\dfrac{\langle c_1, \sigma\rangle \rightsquigarrow \langle c_1', \sigma'\rangle}{\langle c_1 || c_2, \sigma\rangle \rightsquigarrow \langle c_1' || c_2, \sigma'\rangle}$

From Small-step to Multi-step Semantics

The multi-step semantics is given by the relation $t \rightsquigarrow^* t'$ which is the reflexive and transitive closure of $t \rightsquigarrow t'$.

(P1) $t \rightsquigarrow^* t$ for every t

(P2) $t \rightsquigarrow t'$ implies $t \rightsquigarrow^* t'$

(P3) $t \rightsquigarrow^* t'$ and $t' \rightsquigarrow^* t''$ implies $t \rightsquigarrow^* t''$

Normal Forms

- A normal form is a term that cannot be evaluated any further : is a state where the abstract machine is halted (result of the evaluation).

Properties of the small and big step semantics

- The relation $\rightsquigarrow$ is deterministic.
- The relation $\Downarrow$ is deterministic.
- $t \Downarrow v$ iff $t \rightsquigarrow^* v$, where v is a "value".

Big-step versus small-step semantics

- In small-step semantics evaluation stops at errors. In big-step semantics errors occur deeply inside derivation trees.
- The order of evaluation is *explicit* in small-step semantics but *implicit* in big-step semantics.
- Big-step semantics is more abstract, but less precise.
- Small-step semantics allows to make difference between non-termination and "getting stuck".

A functional langage

$$
\begin{array}{rlll}
M, N ::= & x & (\textit{variable}) & | \\
 & ct & (\textit{constant}) & | \\
 & \langle M, N \rangle & (\textit{pair}) & | \\
 & M\ N & (\textit{application}) & | \\
 & \lambda x.M & (\textit{abstraction}) & | \\
 & \mathbf{let}\ x = M\ \mathbf{in}\ N & (\textit{let}) &
\end{array}
$$

Some constant function symbols : *fst*, *snd*, *fix*, *ifthenelse*, $+$, $*$ ...
Some constants : *true*, *false*, 0, 1, 2, 3, ...

Notations

$$
\begin{array}{lcl}
M_1 M_2 \ldots M_n & \equiv & (\ldots((M_1\ M_2) M_3) \ldots M_{n-1})\ M_n \\
N\ \vec{M} & \equiv & (\ldots(((N\ M_1)\ M_2) M_3) \ldots M_{n-1})\ M_n \\
M + N & \equiv & +\langle M, N\rangle \\
\textbf{if } E \textbf{ then } M \textbf{ else } N & \equiv & \mathit{ifthenelse}\langle E, \langle M, N\rangle\rangle
\end{array}
$$

Free variables

$$
\begin{array}{lcl}
FV(x) & = & \{x\} \\
FV(ct) & = & \emptyset \\
FV(\langle M, N \rangle) & = & FV(M) \cup FV(N) \\
FV(M\ N) & = & FV(M) \cup FV(N) \\
FV(\lambda x.M) & = & FV(M) \setminus \{x\} \\
FV(\textbf{let } x = M \textbf{ in } N) & = & FV(M) \cup FV(N) \setminus \{x\}
\end{array}
$$

A term M is closed iff it has no free variable, i.e. $FV(M) = \emptyset$. For example, $\lambda z.((\lambda x.x\ z)(\lambda y.y))$ is closed but $(\lambda x.x\ z)(\lambda y.y)$ is not.

Bound variables

$$
\begin{array}{lcl}
BV(x) & = & \emptyset \\
BV(ct) & = & \emptyset \\
BV(\langle M, N \rangle) & = & BV(M) \cup BV(N) \\
BV(M\ N) & = & BV(M) \cup BV(N) \\
BV(\lambda x.M) & = & BV(M) \cup \{x\} \\
BV(\textbf{let } x = M \textbf{ in } N) & = & BV(M) \cup BV(N) \cup \{x\}
\end{array}
$$

A variable may be free and bound : $x\ (\lambda x.x)$.

Alpha-conversion

Alpha-conversion is the operation which consists in renaming some bound variables.
Thus for example $x\ (\lambda x.x\ y) =_\alpha x\ (\lambda z.z\ y)$ and
let $x = x'$ **in** $x\ y =_\alpha$ **let** $z = x'$ **in** $z\ y$.

Theorem

For every term t there is a term t' such that

1. $t =_\alpha t'$
2. **Barendregt's Convention** *:*
    - $FV(t') \cap BV(t') = \emptyset$.
    - *All the bound variables of t' are distinct.*

Substitution

The application of a substitution $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ to a term M is defined by induction as follows :

$$
\begin{array}{lcll}
\sigma x_i & = & t_i & \text{If } i \in \{1, \ldots, n\} \\
\sigma y & = & y & \text{If } y \notin \{x_1, \ldots, x_n\} \\
\sigma ct & = & ct & \\
\sigma \langle M, N \rangle & = & \langle \sigma M, \sigma N \rangle & \\
\sigma (M\ N) & = & \sigma M\ \sigma N & \\
\sigma (\lambda x.M) & = & \lambda x.\sigma M & \text{If no capture of variables} \\
\sigma (\mathbf{let}\ x = M\ \mathbf{in}\ N) & = & \mathbf{let}\ x = \sigma M\ \mathbf{in}\ \sigma N & \text{If no capture of variables}
\end{array}
$$

Reduction Rules

$$
\begin{array}{lcl}
(\lambda x.M)\ N & \to & M\{x/N\} \\
\textbf{let } x = N \textbf{ in } M & \to & M\{x/N\} \\
\mathit{fix}\ M & \to & M\ (\mathit{fix}\ M) \\
\mathit{fst}\langle M, N\rangle & \to & M \\
\mathit{snd}\langle M, N\rangle & \to & N \\
\textbf{if } \mathit{true} \textbf{ then } M \textbf{ else } N & \to & M \\
\textbf{if } \mathit{false} \textbf{ then } M \textbf{ else } N & \to & N \\
\textbf{if } 0 \textbf{ then } M \textbf{ else } N & \to & M \\
\textbf{if } n \textbf{ then } M \textbf{ else } N & \to & N, \quad n \neq 0
\end{array}
$$

WARNING ! : The reduction relation $\to$ is non-deterministic.

Call-by-value lambda-calculus (big-step semantics)

$$(\textbf{Values})\ \ V ::= ct \mid \langle V, V\rangle \mid \lambda x.M \mid \mathit{fix}\ \ M$$

Meaningless expressions such as $(\langle 1, 1\rangle\ 3)$ or $(\mathit{true}\ 3)$ are not considered as values.

$$\frac{V \text{ is a value}}{V\ \ \Downarrow_v\ \ V} \qquad \frac{M_1 \Downarrow_v V_1 \qquad M_2 \Downarrow_v V_2}{\langle M_1, M_2\rangle \Downarrow_v \langle V_1, V_2\rangle}$$

$$\frac{M \Downarrow_v \lambda x.L \qquad N \Downarrow_v W \qquad L\{x/W\} \Downarrow_v V}{M\ N \Downarrow_v V}$$

$$\frac{N \Downarrow_v V \qquad L\{x/V\} \Downarrow_v W}{\textbf{let}\ x = N\ \textbf{in}\ L \Downarrow_v W}$$

$$\frac{M \Downarrow_v \textit{fix } L \quad N \Downarrow_v W \quad (L\ (\textit{fix } L))\ W \Downarrow_v V}{M\ N \Downarrow_v V}$$

$$\frac{M \Downarrow_v \textit{fst} \quad N \Downarrow_v \langle V_1, V_2 \rangle}{M\ N \Downarrow_v V_1} \qquad \frac{M \Downarrow_v \textit{snd} \quad N \Downarrow_v \langle V_1, V_2 \rangle}{M\ N \Downarrow_v V_2}$$

$$\frac{M \Downarrow_v \textit{true} \quad N \Downarrow_v V}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_v V} \qquad \frac{M \Downarrow_v \textit{false} \quad L \Downarrow_v V}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_v V}$$

$$\frac{M \Downarrow_v 0 \quad N \Downarrow_v V}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_v V} \qquad \frac{M \Downarrow_v n \quad n \neq 0 \quad L \Downarrow_v V}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_v V}$$

Particular case : closed pure lambda-terms

$$(\textbf{Values})\ V ::= \lambda x.M$$

$$\frac{}{V \ \Downarrow_v \ V} \qquad \frac{M \Downarrow_v \lambda x.L \quad N \Downarrow_v W \quad L\{x/W\} \Downarrow_v V}{M\ N \Downarrow_v V}$$

An example

$M = \lambda f.\lambda x.\langle x, f\ x\rangle$ and $N = \lambda y.y$.

$$\frac{M\ N \Downarrow_v \lambda x.\langle x, N\ x\rangle \quad 1 \Downarrow_v 1 \quad \langle 1, N\ 1\rangle \Downarrow_v \langle 1, 1\rangle}{M\ N\ 1 \Downarrow_v \langle 1, 1\rangle}$$

$$\frac{M \Downarrow_v M \quad N \Downarrow_v N \quad \lambda x.\langle x, f\ x\rangle\{f/N\} \Downarrow_v \lambda x.\langle x, N\ x\rangle}{M\ N \Downarrow_v \lambda x.\langle x, N\ x\rangle}$$

$$\frac{1 \Downarrow_v 1 \quad \dfrac{N \Downarrow_v N \quad 1 \Downarrow_v 1 \quad y\{y/1\} \Downarrow_v 1}{N\ 1 \Downarrow_v 1}}{\langle 1, N\ 1\rangle \Downarrow_v \langle 1, 1\rangle}$$

Call-by-value lambda calculus (small-step semantics)

$$\frac{M \leadsto_v M'}{M\ N \leadsto_v M'\ N} \qquad \frac{N \leadsto_v N'}{V\ N \leadsto_v V\ N'}$$

$$\frac{}{(\lambda x.M)\ V \leadsto_v M\{x/V\}} \qquad \frac{}{(\mathit{fix}\ M)\ V \leadsto_v (M\ (\mathit{fix}\ M))\ V}$$

$$\frac{N \leadsto_v N'}{\textbf{let}\ x = N\ \textbf{in}\ L \leadsto_v \textbf{let}\ x = N'\ \textbf{in}\ L} \qquad \frac{}{\textbf{let}\ x = V\ \textbf{in}\ L \leadsto_v L\{x/V\}}$$

$$\frac{M \leadsto_v M'}{\langle M, N\rangle \leadsto_v \langle M', N\rangle} \qquad \frac{N \leadsto_v N'}{\langle V, N\rangle \leadsto_v \langle V, N'\rangle}$$

$$\frac{}{\mathit{fst}\ \langle V_1, V_2\rangle \leadsto_v V_1} \qquad \frac{}{\mathit{snd}\ \langle V_1, V_2\rangle \leadsto_v V_2}$$

$$\frac{M \rightsquigarrow_v M'}{\textbf{if } M \textbf{ then } N \textbf{ else } L \rightsquigarrow_v \textbf{if } M' \textbf{ then } N \textbf{ else } L}$$

$$\frac{}{\textbf{if } \mathit{true} \textbf{ then } N \textbf{ else } L \rightsquigarrow_v N} \qquad \frac{}{\textbf{if } \mathit{false} \textbf{ then } N \textbf{ else } L \rightsquigarrow_v L}$$

$$\frac{}{\textbf{if } 0 \textbf{ then } N \textbf{ else } L \rightsquigarrow_v N} \qquad \frac{n \neq 0}{\textbf{if } n \textbf{ then } N \textbf{ else } L \rightsquigarrow_v L}$$

The same example

$M = \lambda f.\lambda x.\langle x, f\ x\rangle$ and $N = \lambda y.y$.

$$\begin{array}{ll} M\ N\ 1 & \rightsquigarrow_v \\ (\lambda x.\langle x, N\ x\rangle)\ 1 & \rightsquigarrow_v \\ \langle 1, N\ 1\rangle & \rightsquigarrow_v \\ \langle 1, 1\rangle & \end{array}$$

Call-by-name lambda-calculus (big-step semantics)

(Lazy Forms) $P ::= ct \mid \langle M, N \rangle \mid \lambda x.M \mid \mathit{fix}\ M$

$$\frac{M \Downarrow_n \lambda x.L \qquad L\{x/N\} \Downarrow_n P}{M\ N \Downarrow_n P} \qquad \frac{P \text{ is a lazy form}}{P\ \Downarrow_n\ P}$$

$$\frac{L\{x/N\} \Downarrow_n P}{\textbf{let } x = N \textbf{ in } L \Downarrow_n P} \qquad \frac{M \Downarrow_n \mathit{fix}\ L \qquad (L\ (\mathit{fix}\ L))\ N \Downarrow_n P}{M\ N \Downarrow_n P}$$

$$\frac{M \Downarrow_n \langle M_1, M_2 \rangle \qquad M_1 \Downarrow_n P_1}{\mathit{fst}\ M \Downarrow_n P_1} \qquad \frac{M \Downarrow_n \langle M_1, M_2 \rangle \qquad M_2 \Downarrow_n P_2}{\mathit{snd}\ M \Downarrow_n P_2}$$

$$\frac{M \Downarrow_n \mathit{true} \qquad N \Downarrow_n P}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_n P} \qquad \frac{M \Downarrow_n \mathit{false} \qquad L \Downarrow_n P}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_n P}$$

$$\frac{M \Downarrow_n 0 \qquad N \Downarrow_n P}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_n P} \qquad \frac{M \Downarrow_n n \qquad n \neq 0 \qquad L \Downarrow_n P}{\textbf{if } M \textbf{ then } N \textbf{ else } L \Downarrow_n P}$$

Particular case : closed pure lambda-terms

$$(\textbf{Lazy Forms})\ P ::= \lambda x.M$$

$$\frac{}{P \Downarrow_n P} \qquad \frac{M \Downarrow_n \lambda x.L \quad L\{x/N\} \Downarrow_n P}{M\ N \Downarrow_n P}$$

An example

Let $M = \lambda f.\lambda x.\langle x, (f\ x)\rangle$

$$\frac{\mathit{fix}\ M \Downarrow_n \mathit{fix}\ M \qquad M\ (\mathit{fix}\ M)\ 1 \Downarrow_n \langle 1, \mathit{fix}\ M\ 1\rangle}{\mathit{fix}\ M\ 1 \Downarrow_n \langle 1, \mathit{fix}\ M\ 1\rangle}$$

Let $M_f = \mathit{fix}\ M$.

$$\frac{\dfrac{M \Downarrow_n M \qquad (\lambda x.\langle x, f\ x\rangle)\{f/M_f\} \Downarrow_n \lambda x.\langle x, M_f\ x\rangle}{M\ M_f \Downarrow_n \lambda x.\langle x, M_f\ x\rangle} \qquad \dfrac{}{\langle x, M_f\ x\rangle\{x/1\} \Downarrow_n \langle 1, M_f\ 1\rangle}}{M\ (M_f)\ 1 \Downarrow_n \langle 1, M_f\ 1\rangle}$$

Exercice

Try to compute $\mathit{fix}\ M\ 1 \Downarrow_v$?

Call-by-name lambda calculus (small-step semantics)

$$\frac{M \rightsquigarrow_n M'}{M\ N \rightsquigarrow_n M'\ N}$$

$$\frac{}{(\lambda x.M)\ N \rightsquigarrow_n M\{x/N\}} \qquad \frac{}{(\mathit{fix}\ M)\ N \rightsquigarrow_n (M\ (\mathit{fix}\ M))\ N}$$

$$\frac{}{\mathbf{let}\ x = M\ \mathbf{in}\ L \rightsquigarrow_n L\{x/M\}}$$

$$\frac{M \rightsquigarrow_n M'}{\mathit{fst}\ M \rightsquigarrow_n \mathit{fst}\ M'} \qquad \frac{}{\mathit{fst}\ \langle M, N\rangle \rightsquigarrow_n M}$$

$$\frac{M \rightsquigarrow_n M'}{\mathit{snd}\ M \rightsquigarrow_n \mathit{snd}\ M'} \qquad \frac{}{\mathit{snd}\ \langle M, N\rangle \rightsquigarrow_n N}$$

$$\frac{M \rightsquigarrow_n M'}{\textbf{if } M \textbf{ then } N \textbf{ else } L \rightsquigarrow_n \textbf{if } M' \textbf{ then } N \textbf{ else } L}$$

$$\frac{}{\textbf{if } \mathit{true} \textbf{ then } N \textbf{ else } L \rightsquigarrow_n N} \qquad \frac{}{\textbf{if } \mathit{false} \textbf{ then } N \textbf{ else } L \rightsquigarrow_n L}$$

$$\frac{}{\textbf{if } 0 \textbf{ then } N \textbf{ else } L \rightsquigarrow_n N} \qquad \frac{n \neq 0}{\textbf{if } n \textbf{ then } N \textbf{ else } L \rightsquigarrow_n L}$$

The same example

$M = \lambda f.\lambda x.\langle x, (f\ x)\rangle$.

$$
\begin{array}{ll}
\mathit{fix}\ M\ 1 & \rightsquigarrow_n \\
M\ (\mathit{fix}\ M)\ 1 & \rightsquigarrow_n \\
(\lambda x.\langle x, (\mathit{fix}\ M\ x)\rangle)\ 1 & \rightsquigarrow_n \\
\langle 1, (\mathit{fix}\ M\ 1)\rangle &
\end{array}
$$

Coherence of results

- If $M \Downarrow_v N$, then N is a value.
- If $M \Downarrow_n N$, then N is a lazy form.

Deterministic properties

- If $M \Downarrow_v V$ and $M \Downarrow_v V'$, then $V = V'$.
- If $M \Downarrow_n P$ and $M \Downarrow_n P'$, then $P = P'$.
- If $M \rightsquigarrow_v N$ and $M \rightsquigarrow_v N'$, then $N = N'$.
- If $M \rightsquigarrow_n N$ and $M \rightsquigarrow_n N'$, then $N = N'$.

Relating big and small-steps semantics

- If $M \Downarrow_v V$, then $M \rightsquigarrow_v^* V$.
- If $M \Downarrow_n P$, then $M \rightsquigarrow_n^* P$.
- If $M \rightsquigarrow_v^* N$ and N is a value, then $M \Downarrow_v N$.
- If $M \rightsquigarrow_n^* N$ and N is a lazy form, then $M \Downarrow_n N$.