Esercizi Semantica Denotazionale

Si considerino alcune varianti di HOFL senza ricorsione, che indicheremo genericamente con PL. Per ciascuna di esse si chiede di dare una semantica denotazione nella categoria degli insiemi, ovvero:

1. Dare le regole di tipaggio per i costrutti aggiunti ad HOFL
2. identificare una segnatura Σ per il metalinguaggio con i tipi computazionali ML_M
3. dare una traduzione CBV di PL in $ML_M(\Sigma)$, che estende quella vista a lezione (quindi basta dare la traduzione per i nuovi costrutti)
4. dare una interpretazione di $ML_M(\Sigma)$ in **Set**. Questa interpretazione puo' essere data in modo indiretto, cioe' mediante traduzione dei costrutti per i tipi computazionali e dei simboli in Σ in un metalinguaggio $ML(\Sigma')$ senza tipi computazionali (p.e. il lambda-calcolo tipato, che sappiamo come interpretare in qualunque CCC con somme), eventualmente esteso con simboli aggiuntivi, specificati dalla segnatura Σ', e la cui interpretazione va data in modo diretto (piu' o meno informale). Alternativamente si puo' dare l'interpretazione dei costrutti per i tipi computazionali e dei simboli in Σ in modo diretto (piu' o meno informale).

HOFL non-deterministico. Si consideri la variante $HOFL_{or}$ di HOFL, caratterizzata dall'aggiunta della scelta non-deterministica (e rimozione delle definizioni ricorsive, che rende possibile una interpretazione nella categoria **Set**):

$$\begin{array}{rcl} n \in \mathsf{N} & ::= & \text{i numeri interi} \\ x \in \mathsf{Var} & ::= & \ldots \\ t \in \mathsf{Exp} & ::= & n \mid t_1 + t_2 \mid \mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid x \mid \lambda x{:}\tau.t \mid t_1\ t_2 \mid t_1\ \mathsf{or}\ t_2 \\ \tau \in \mathsf{Type} & ::= & \mathsf{int} \mid \tau_1 \to \tau_2 \end{array}$$

1. regole di tipaggio ...
2. segnatura Σ per ML_M (oltre alla parte relativa agli interi)
   $or{:}\forall X.MX \to MX \to MX$
3. traduzione
   $(t_1\ \mathsf{or}\ t_2)^e \stackrel{\Delta}{=} or[\tau^e](t_1^e, t_2^e)$, dove τ e' il tipo di t_1 e t_2.
   Piu' formalmente la traduzione di $\Gamma \vdash_{PL} t{:}\tau$ dovrebbe essere definita per induzione sulla (unica) derivazione di $\Gamma \vdash_{PL} t{:}\tau$.
4. interpretazione
   $MX \stackrel{\Delta}{=} \mathcal{P}_{fin}(X)$, l'insieme dei sottoinsiemi finiti di X
   $[e]_M \stackrel{\Delta}{=} \{e\}$, cioe' un soto-insieme singoletto
   $\mathsf{let}_M\, x \Leftarrow e_1\, \mathsf{in}\, e_2 \stackrel{\Delta}{=} \cup\{e_2 | x \in e_1\}$
   $or(e_1, e_2) \stackrel{\Delta}{=} e_1 \cup e_2$
5. si puo' far vedere che questa interpretazione di $ML_M(\Sigma)$ convalida la seguente proprieta' equazionale
   $\mathsf{let}_M\, x \Leftarrow or(e_1, e_2)\, \mathsf{in}\, e = or(\mathsf{let}_M\, x \Leftarrow e_1\, \mathsf{in}\, e, \mathsf{let}_M\, x \Leftarrow e_2\, \mathsf{in}\, e)$

HOFL con stato. Si consideri la variante $HOFL_{loc}$ di HOFL, caratterizzata dall'aggiunta delle locazioni (contenenti interi, come in IMP):

$$\begin{array}{rcl} l \in \mathsf{Loc} & ::= & \text{locazioni} \\ n \in \mathsf{N} & ::= & \text{i numeri interi} \\ x \in \mathsf{Var} & ::= & \ldots \\ t \in \mathsf{Exp} & ::= & n \mid t_1 + t_2 \mid \mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid x \mid \lambda x{:}\tau.t \mid t_1\ t_2 \mid l \mid t_1{:=}\,t_2 \mid\ !t \\ \tau \in \mathsf{Type} & ::= & \mathsf{int} \mid \mathsf{loc} \mid \tau_1 \to \tau_2 \end{array}$$

dove i nuovi costrutti hanno la seguente semantica informale

- $t_1{:=}\,t_2$, se t_1 e' una locazione l e t_2 ha valore n, allora assegna n ad l e restituisci l

- $!t$, se t e' una locazione l, allora restituisci il contenuto n di l

1. segnatura Σ per ML_M

   L: 0 locazioni
   $ifeq: \forall X.L \to L \to X \to X \to X$ test uguaglianza per locazioni

   $get: L \to MN, \quad set: L \to N \to ML$

   oppure

   $get: MS, \quad set: S \to MS$ dove $S = L \to N$

2. interpretazione

   $MX \stackrel{\Delta}{=} S \to (X \times S)$ dove $S \stackrel{\Delta}{=} L \to N$
   ...

3. proprieta' equazionali validate da interpretazione di $ML_M(\Sigma)$

   $\mathsf{let}_M\, n \Leftarrow get(l)\, \mathsf{in}\, set(l, n) = get(l)$
   $\mathsf{let}_M\, n' \Leftarrow set(l, n)\, \mathsf{in}\, get(l) = set(l, n)$

HOFL con eccezioni. Si consideri la variante $HOFL_{exc}$ di HOFL, che aggiunge dei costrutti tipo CAML per la gestione delle eccezioni

$$
\begin{array}{rcl}
exn \in \mathsf{Exn} & ::= & \text{nomi di eccezione} \\
n \in \mathsf{N} & ::= & \text{i numeri interi} \\
x \in \mathsf{Var} & ::= & \ldots \\
t \in \mathsf{Exp} & ::= & n \mid t_1 + t_2 \mid \mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid x \mid \lambda x{:}\tau.t \mid t_1\ t_2 \mid \\
 & & exn \mid \mathsf{raise}(t) \mid t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2 \\
\tau \in \mathsf{Type} & ::= & \mathsf{int} \mid \mathsf{exc} \mid \tau_1 \to \tau_2
\end{array}
$$

dove i nuovi costrutti hanno la seguente semantica informale

- $\mathsf{raise}(t)$, se t si valuta in una eccezione exn, allora solleva l'eccezione exn
- t_0 handle t_1 with t_2, se t_0 non solleva eccezioni, si comporta come t_0; se invece t_0 solleva una eccezione exn e t_1 si valuta in exn, allora si valuta t_2

1. segnatura Σ per ML_M

   E: 0 eccezioni
   $ifeq: \forall X.E \to E \to X \to X \to X$ test uguaglianza per eccezioni

   $raise: \forall X.E \to MX$
   $handle: \forall X.MX \to (E \to MX) \to MX$

2. interpretazione

   $MX \stackrel{\Delta}{=} (X + E)$
   $[e]_M \stackrel{\Delta}{=} in_0(e)$
   $\mathsf{let}_M\, x \Leftarrow e_1\, \mathsf{in}\, e_2 \stackrel{\Delta}{=}$ case e_1 of $in_0(x) \Rightarrow e_2 \mid in_1(exn) \Rightarrow in_1(exn)$
   $raise(e) \stackrel{\Delta}{=} in_1(e)$
   $handle(e, h) \stackrel{\Delta}{=}$ case e of $in_0(x) \Rightarrow in_0(x) \mid in_1(exn) \Rightarrow h(exn)$

3. proprieta' equazionali validate da interpretazione di $ML_M(\Sigma)$

   $\mathsf{let}_M\, x \Leftarrow raise(exn)\, \mathsf{in}\, e = raise(exn)$
   $handle(raise(exn), h) = h(exn)$
   $handle([e]_M, h) = [e]_M$

Altre varianti. Semantica denotazionale per $HOFL_{exc,or}$ o $HOFL_{loc,or}$. SUGGERIMENTO: usare le seguenti interpretazioni dei tipi computazionali $MX = \mathcal{P}_{fin}(X + E)$ e $MX = S \to \mathcal{P}_{fin}(X \times S)$, rispettivamente.

Compitino su Semantica Denotazionale

Sia PL l'estensione di HOFL con stato ed eccezioni, cioe'

$$
\begin{array}{rcl}
n \in \mathsf{N} & ::= & \text{i numeri interi} \\
x \in \mathsf{Var} & ::= & \ldots \\
t \in \mathsf{Exp} & ::= & n \mid t_1 + t_2 \mid \mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid x \mid \lambda x{:}\tau.t \mid t_1\ t_2 \mid \\
 & & t_1 := t_2 \mid\ !t \mid \mathsf{raise}(t) \mid t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2 \\
\tau \in \mathsf{Type} & ::= & \mathsf{int} \mid \mathsf{loc} \mid \mathsf{exc} \mid \tau_1 \to \tau_2
\end{array}
$$

I nuovi costrutti hanno la seguente semantica informale:

- $t_1 := t_2$, se t_1 e' una locazione l e t_2 ha valore n, allora assegna n ad l e restituisci l
- $!t$, se t e' una locazione l, allora restituisci
- $\mathsf{raise}(t)$, se t si valuta in una eccezione exn, allora solleva l'eccezione exn
- t_0 handle t_1 with t_2, se t_0 non solleva eccezioni, si comporta come t_0; se invece t_0 solleva una eccezione exn e t_1 si valuta in exn, allora si valuta t_2

Si considerino due possibili semantiche CBV per tale linguaggio:

1. nella prima (analoga a CAML) il sollevamento di una eccezione non comporta la perdita dello stato, ovvero le asserzioni per la valutazione delle espressioni sono della forma forma $t, \mu \to \mathsf{val}\ v, \mu'$ e $t, \mu \to \mathsf{exc}\ exn, \mu'$;
2. nella seconda (analoga alla semantica con errore) il sollevamento di una eccezione comporta la perdita dello stato corrent, ovvero le asserzioni per la valutazione delle espressioni sono della forma forma $t, \mu \to \mathsf{val}\ v, \mu'$ e $t, \mu \to \mathsf{exc}\ exn$.

Si richiedi di:

1. Dare le regole per valutare $\mathsf{raise}(t)$ e t_0 handle t_1 with t_2 nelle due versioni di semantica operazione.
2. Identificare una segnatura Σ per il metalinguaggio con i tipi computazionali ML_M, che vada bene per entrambe le traduzioni CBV di PL in $ML_M(\Sigma)$.
3. Dare due interpretazioni di $ML_M(\Sigma)$ in **Set**, corrispondenti alle due versioni di semantica denotazionale.
4. Dare due termini ben tipati $\Gamma \vdash_{PL} t_i{:}\tau$ di PL che risultino equivalenti in una semantica denotazionale, ma non nell'altra.

$$PL \xrightarrow{\_^e} ML_M(\Sigma) \underset{[\![\_]\!]_2}{\overset{[\![\_]\!]_1}{\rightrightarrows}} \mathbf{Set}$$

Soluzione Compitino su Semantica Denotazionale

1. regole per valutare $\mathsf{raise}(t)$ e t_0 handle t_1 with t_2 nelle due versioni di semantica operazione

   (a) $$\frac{t,\mu_0 \to \mathsf{val}\ exn,\mu_1}{\mathsf{raise}(t),\mu_0 \to \mathsf{exc}\ exn,\mu_1} \qquad \frac{t,\mu_0 \to \mathsf{exc}\ exn,\mu_1}{\mathsf{raise}(t),\mu_0 \to \mathsf{exc}\ exn,\mu_1}$$
   $$\frac{t_0,\mu_0 \to \mathsf{val}\ v,\mu_1}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to \mathsf{val}\ v,\mu_1} \qquad \frac{t_0,\mu_0 \to \mathsf{exc}\ exn,\mu_1 \quad t_1,\mu_1 \to \mathsf{val}\ exn,\mu_2 \quad t_2,\mu_2 \to r}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to r}$$
   $$\frac{\begin{array}{c} t_0,\mu_0 \to \mathsf{exc}\ exn,\mu_1 \\ t_1,\mu_1 \to \mathsf{val}\ exn',\mu_2 \end{array}}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to \mathsf{exc}\ exn,\mu_2}\ exn' \not\equiv exn \qquad \frac{\begin{array}{c} t_0,\mu_0 \to \mathsf{exc}\ exn,\mu_1 \\ t_1,\mu_1 \to \mathsf{exc}\ exn',\mu_2 \end{array}}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to \mathsf{exc}\ exn,\mu_2}$$

   (b) $$\frac{t,\mu_0 \to \mathsf{val}\ exn,\mu_1}{\mathsf{raise}(t),\mu_0 \to \mathsf{exc}\ exn} \qquad \frac{t,\mu_0 \to \mathsf{exc}\ exn}{\mathsf{raise}(t),\mu_0 \to \mathsf{exc}\ exn}$$
   $$\frac{t_0,\mu_0 \to \mathsf{val}\ v,\mu_1}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to \mathsf{val}\ v,\mu_1} \qquad \frac{t_0,\mu_0 \to \mathsf{exc}\ exn \quad t_1,\mu_0 \to \mathsf{val}\ exn,\mu_2 \quad t_2,\mu_2 \to r}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to r}$$
   $$\frac{\begin{array}{c} t_0,\mu_0 \to \mathsf{exc}\ exn \\ t_1,\mu_0 \to \mathsf{val}\ exn',\mu_2 \end{array}}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to \mathsf{exc}\ exn}\ exn' \not\equiv exn \qquad \frac{\begin{array}{c} t_0,\mu_0 \to \mathsf{exc}\ exn \\ t_1,\mu_0 \to \mathsf{exc}\ exn' \end{array}}{t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2,\mu_0 \to \mathsf{exc}\ exn}$$

2. segnatura Σ per il metalinguaggio con i tipi computazionali ML_M, segnatura per gli interi ed in piu'

   $L, E{:}\,0$ locazioni ed eccezioni
   $ifeq{:}\,\forall X.L \to L \to X \to X \to X$ test uguaglianza per locazioni (non usato nella traduzione)
   $ifeq{:}\,\forall X.E \to E \to X \to X \to X$ test uguaglianza per eccezione
   $get{:}\,L \to MN$
   $set{:}\,L \to N \to ML$
   $raise{:}\,\forall X.E \to MX$
   $handle{:}\,\forall X.MX \to (E \to MX) \to MX$

   la traduzione CBV $\_^e{:}\,PL \to ML_M(\Sigma)$ per i costrutti aggiunti a HOFL e'

   $(!t)^e \triangleq \mathsf{let}_M\, l \Leftarrow t^e\ \mathsf{in}\, get(l)$
   $(t_1{:=}\,t_2)^e \triangleq \mathsf{let}_M\, l \Leftarrow t_1^e,\ n \Leftarrow t_2^e\ \mathsf{in}\, set(l,n)$
   $(\mathsf{raise}(t))^e \triangleq \mathsf{let}_M\, exn \Leftarrow t^e\ \mathsf{in}\, raise(exn)$
   $(t_0\ \mathsf{handle}\ t_1\ \mathsf{with}\ t_2)^e \triangleq handle(t_0^e, \lambda exn{:}\,E.\mathsf{let}_M\, exn' \Leftarrow t_1^e\ \mathsf{in}\, ifeq(exn, exn', t_2^e, raise(exn)))$

3. interpretazioni di $ML_M(\Sigma)$ in **Set**, corrispondenti alle due versioni di semantica denotazionale

   (a) $MX \triangleq S \to ((X+E)\times S)$ dove $S = L \to N$
   $[e]_M \triangleq \lambda s{:}\,S.(in_0(e), s)$
   $\mathsf{let}_M\, x \Leftarrow e_1\ \mathsf{in}\, e_2 \triangleq \lambda s{:}\,S.\mathsf{let}\ (u,s') = e_1\ s\ \mathsf{in}\ \mathsf{case}\ u\ \mathsf{of}\ in_0(x) \Rightarrow e_2\ s' \mid in_1(exn) \Rightarrow (in_1(exn), s')$
   $get(l) \triangleq \lambda s{:}\,S.(in_0(s\ l), s)$
   $set(l,n) \triangleq \lambda s{:}\,S.(in_0(l), s[l{:=}\,n])$
   $raise(exn) \triangleq \lambda s{:}\,S.(in_1(exn), s)$
   $handle(e,h) \triangleq \lambda s{:}\,S.\mathsf{let}\ (u,s') = e\ s\ \mathsf{in}\ \mathsf{case}\ u\ \mathsf{of}\ in_0(x) \Rightarrow (in_0(x), s') \mid in_1(exn) \Rightarrow h\ exn\ s'$

   (b) $MX \triangleq S \to ((X \times S)+E)$ dove $S = L \to N$
   $[e]_M \triangleq \lambda s{:}\,S.in_0(e,s)$
   $\mathsf{let}_M\, x \Leftarrow e_1\ \mathsf{in}\, e_2 \triangleq \lambda s{:}\,S.\mathsf{case}\ e_1\ s\ \mathsf{of}\ in_0(x,s) \Rightarrow e_2\ s' \mid in_1(exn) \Rightarrow in_1(exn)$
   $get(l) \triangleq \lambda s{:}\,S.in_0(s\ l, s)$
   $set(l,n) \triangleq \lambda s{:}\,S.in_0(l, s[l{:=}\,n])$
   $raise(exn) \triangleq \lambda s{:}\,S.in_1(exn)$
   $handle(e,h) \triangleq \lambda s{:}\,S.\mathsf{case}\ e\ s\ \mathsf{of}\ in_0(x,s') \Rightarrow in_0(x,s') \mid in_1(exn) \Rightarrow h\ exn\ s$

4. termini ben tipati $\Gamma \vdash_{PL} t_i{:}\,\tau$ di PL che risulta equivalenti in una semantica denotazionale, ma non nell'altra

   $l{:}\,\mathsf{loc}, exn{:}\,\mathsf{exc} \vdash_{PL} (\lambda x{:}\,\mathsf{loc}.\mathsf{raise}(exn))\ (l{:=}\,0) = \mathsf{raise}(exn){:}\,\tau$ e' vera nella senatica con $MX = S \to ((X\times S)+E)$, ma non vale nell'altra semantica.