Paradigma Funzionale

Paradigma Imperativo: Programma = transizione di stato
Paradigma Funzionale: Programma = valutazione di un’espressione
La maggior parte dei linguaggi imperativi ha costrutti applicativi.

Esempio: consideriamo la seguente grammatica LF

Prog	::=	Decs **eval** Exp.
Decs	::=	Dec | Dec Decs.
Dec	::=	CDec | FDec.
CDec	::=	Type Id = Exp;.
FDec	::=	Type Id(PDecs) {Exp}.
PDecs	::=	PDec | PDec PDecs.
PDec	::=	Type Id;
Type	::=	**int** | **bool** | Types □ Type
Types	::=	Type | Type □ Types
Exp	::=	Id | Id(Exps) | Exp + Exp | … **if** (Exp) {Exp} **else** {Exp}.
Exps	::=	Exp | Exp, Exps.

Confronto con LW

I costrutti di LF sono in gran parte simili a quelli di LW.
Rispetto a LW, una (grossa) differenza è che oltre ai tipi base, compaiono dei tipi funzionali o *higher-order*.
Quindi LF è un linguaggio per manipolare valori che possono essere base ($[\![\mathbf{int}]\!] = \mathbb{Z}$, $[\![\mathbf{bool}]\!] = \mathbb{B}$) o funzionali (definiti induttivamente da $[\![t_1 \times \ldots \times t_n \rightarrow t]\!] = [[\![t_1]\!] \times \ldots \times [\![t_n]\!] \rightarrow_P [\![t]\!]]$).
Nel seguito useremo Value $= \bigcup_{t \in L_{Type}(LF)} [\![t]\!]$.
I tipi higher-order sono cittadini di prima classe, cioè possono comparire in tutti i contesti in cui può comparire un tipo base (eg come argomenti e risultati di funzioni).
Principale differenza con linguaggi funzionali “reali”: manca il polimorfismo e i tipi sono dichiarati invece che dedotti.
La semantica di LF sarà molto simile a quella di LW, ma, non essendo necessaria la nozione di stato, tutto si semplificherà.

Semantica Statica

Ignoriamo il trattamento di errore (cioè definiamo la semantica statica come funzione parziale)

$Env^s = [L_{Id}(LF) \rightarrow_P L_{Type}(LF)]_{Fin}$

$[\![\_]\!]^s_D : L_{Decs}(LF) \rightarrow Env^s \times Env^s \rightarrow_P Env^s$

$[\![\_]\!]^s_E : L_{Exps}(LF) \rightarrow Env^s \rightarrow_P L_{Type}(LF)^+$

$[\![\_]\!]^s_P : L_{Prog}(LF) \rightarrow_P L_{Type}(LF)$

$$\frac{[\![dl]\!]^s_D([],[]) = \rho \qquad [\![e]\!]^s_E(\rho) = t}{[\![dl\ \mathbf{eval}\ e]\!]^s_P = t}$$

$$\frac{}{types(t\ x;) = t} \qquad \frac{types(pl) = tl}{types(t\ x;\ pl) = t\ tl}$$

$$\frac{[\![d]\!]^s_D(\rho_g, \rho_l) = \rho_l' \qquad [\![dl]\!]^s_D(\rho_g, \rho_l') = \rho}{[\![d\ dl]\!]^s_D(\rho_g, \rho_l) = \rho}$$

$$\frac{[\![e]\!]^s_E(\rho_g[\rho_l]) = t}{[\![t\ x = e;]\!]^s_D(\rho_g, \rho_l) = \rho_l[t/x]} \quad x \notin Dom(\rho_l)$$

$$\frac{[\![pl]\!]^s_{PD}(\rho_g[\rho_l],[lt \mapsto t/f]) = \rho_l' \qquad [\![e]\!]^s_E(\rho_g[\rho_l][\rho_l']) = t}{[\![t\ f(pl)\{e\}]\!]^s_D(\rho_g, \rho_l) = \rho_l[lt \mapsto t/f]} \quad \begin{array}{l} f \notin Dom(\rho_l) \\ lt = types(pl) \end{array}$$

$$\frac{}{[\![t\ x;]\!]^s_{PD}(\rho_g, \rho_l) = \rho_l[t/x]} \quad x \notin Dom(\rho_l)$$

$[\![\_]\!]^s_{PD} : L_{PDecs}(LF) \rightarrow Env^s \times Env^s \rightarrow_P Env^s$

Semantica Statica 2

$$\frac{}{[\![x]\!]^s_E(\square) = t} \quad \square(x) = t$$

$$\frac{[\![f]\!]^s_E(\square) = lt \,\square\, t \qquad [\![es]\!]^s_E(\square) = lt}{[\![f(es)]\!]^s_D(\square) = t}$$

$$\frac{[\![b]\!]^s_E(\square) = \mathbf{bool} \quad [\![e]\!]^s_E(\square) = t \quad [\![e']\!]^s_E(\square) = t}{[\![\mathbf{if}\ (b)\ \{e\}\ \mathbf{else}\ \{e'\}]\!]^s_E(\square) = t}$$

$$\frac{[\![e]\!]^s_E(\square) = \mathbf{int} \qquad [\![e']\!]^s_E(\square) = \mathbf{int}}{[\![e + e']\!]^s_E(\square) = \mathbf{int}}$$

Esercizio proposto: aggiungere altri costrutti base al linguaggio, ad esempio costrutti per la funzione identica, composizione di funzioni, tipi prodotto, *pairing* (se f: A $\square$ B e g: C $\square$ D, allora (f,g): A $\square$ C $\square$ B $\square$ D è definita da (f,g)(a,c) = (f(a),g(b)))...

Semantica Dinamica
(Denotazionale)

Le dichiarazioni modificano l'ambiente: $[\![\_]\!]_{Decs}: L_{Decs}(LF) \to Env \to Env$
dove $Env = [L_{Id}(LF) \rightharpoonup_P Value]_{Fin}$

Le espressioni producono un valore: $[\![\_]\!]_{Exps}: L_{Exps}(LF) \to Env \to Value^{+}$

$$\frac{[\![dl]\!]_D([\,]) = r \qquad [\![e]\!]_E(r) = v}{[\![dl\ \mathbf{eval}\ e]\!]_P = v} \qquad\qquad \frac{[\![e]\!]_E(r) = v}{[\![t\ x = e;]\!]_D(r) = r[v/x]}$$

$$\frac{[\![d]\!]_D(r) = r' \qquad [\![dl]\!]_D(r') = r''}{[\![d\ dl]\!]_D(r) = r''} \qquad\qquad \frac{}{[\![RT\ f(T\ x)\ \{e\}]\!]_D(r) = r[F/f]}$$

Dove $F : [\![T]\!] \to [\![RT]\!]$ è definita induttivamente da tutte le regole della semantica più le seguenti due:

Rispetto a LW è sparito lo stato

Il valore del parametro attuale è associato direttamente al nome del parametro formale

Caso base

$$\frac{[\![e]\!](r[\square/f, v/x]) = a}{F(v) = a}$$

Regola ad hoc per la chiamata di f

$$\frac{[\![e]\!](r[\square/f]) = v \qquad F(v) = a}{[\![f(e)]\!](r[\square/f]) = a}$$

Semantica Dinamica
(2)

$$\frac{[\![e]\!]_E(r) = \mathbb{v} \qquad [\![es]\!]_E(r) = \mathbb{l}\mathbb{v}}{[\![e\ es]\!]_E(r) = \mathbb{v}\ \mathbb{l}\mathbb{v}}$$

$$\frac{}{[\![x]\!]_E(r) = r(x)}$$

$$\frac{[\![es]\!]_E(r) = \mathbb{l}\mathbb{v}}{[\![f(es)]\!]_E(r) = r(f)(\mathbb{l}\mathbb{v})}$$

$$\frac{[\![e_1]\!]_E(r) = \mathbb{v}_1 \qquad [\![e_2]\!]_E(r) = \mathbb{v}_2}{[\![e_1 + e_2]\!]_E(r) = \mathbb{a}} \quad \mathbb{v}_1 + \mathbb{v}_2 = \mathbb{a}$$