Probabilistic Embedding: Experiments with Tuple-based Probabilistic Languages

Stefano Mariani, Andrea Omicini
{s.mariani, andrea.omicini}@unibo.it
Dipartimento di Informatica - Scienza e Ingegneria
Università di Bologna

Embedding --------> Modular Embedding ------> Probabilistic Embedding

Shapiro (1991)
Embedding
(Sequential Languages)

Languages L, L',
program sets P, P'
observable behaviours B, B',
observation criteria O, O' to hold
O: P ---> B, O': P' ---> B'
L embeds L' iff
exist C: P' ---> P, D: B ---> B'
s.t. for every program W in L'
D(O[C(W)]) = O'[W]

De Boer - Palamidessi (1994)
Modular Embedding
(Concurrent Languages)

Embedding
+
Independent observation:
for every Bi in B
D(Bi) = {d(b)|b in Bi}
+
C compositionality:
C(W1||W2) = C(W1)||C(W2)
C(W1+W2) = C(W1)+C(W2)
+
Deadlock invariance:
for every Bi in B, b in Bi
tm'[D(b)] = tm(b)

Probabilistic Modular Embedding (PME)
(Probabilistic Tuple-based Languages)

Modular Embedding
+
Probabilistic observation:

$$\Theta[W] = \left\{ (\rho, W[\bar{\mu}]) \mid \quad (W, \langle\sigma\rangle) \longrightarrow^{*} (\rho, W[\bar{\mu}]) \right\}$$

+
Probabilistic termination:

$$\Phi[W] = \left\{ (\rho_{\perp}, \tau) \mid \quad (W, \langle\sigma\rangle) \longrightarrow^{*}_{\perp} (\rho_{\perp}, \tau) \right\}$$

+
Probabilistic aggregation functions:

$$\bar{\nu} : W \times \langle\sigma\rangle \mapsto \rho \ \text{ where } \ \rho = \prod_{j=0}^{n} \{ p_j \mid (p_j, \mu_{\bar{\ell}}) \in \Theta[W = \bar{\ell}.\emptyset] \}$$

$$\nu^{+} : W \times \langle\sigma\rangle \mapsto \rho \ \text{ where } \ \rho = \sum_{j=0}^{n} \{ p_j \mid (p_j, \mu_{\ell^{+}}) \in \Theta[W = \ell^{+}.\emptyset] \}$$

upLINDA $\succeq_p$ uLINDA / pLINDA $\succeq_p$ LINDA

uLINDA vs. LINDA

P = uin(T).0 + uin(T).urd(T').0
Q = in(T).0 + in(T).rd(T').0
S = < tl, tl, tr >

o[P] = (success, < tl, tr >) |
(deadlock, < tl, tl >)
o[Q] = (success, < tl, tr >) |
(deadlock, < tl, tl >)

Phi[P] = (2/3, success) | (1/3, deadlock)
Phi[Q] = (-, success) | (-, deadlock)

$$C_{Linda} = \begin{cases} \texttt{out} & \longmapsto \ \texttt{uout} \\ \texttt{rd} & \longmapsto \ \texttt{urd} \\ \texttt{in} & \longmapsto \ \texttt{uin} \end{cases}$$

$$\text{uLINDA} \succeq_p \text{LINDA}, \ \text{LINDA} \not\succeq_p \text{uLINDA}$$
$$\Longrightarrow \quad \text{uLINDA} \not\equiv_o \text{LINDA}$$

pLINDA vs. LINDA

P = 2/3 in(T).0 + 1/3 in(T).rd(T').0
Q = in(T).0 + in(T).rd(T').0
S = < tl, tl, tr >

o[P] = (success, < tl, tr >) |
(deadlock, < tl, tl >)
o[Q] = (success, < tl, tr >) |
(deadlock, < tl, tl >)

Phi[P] = (2/3, success) | (1/3, deadlock)
Phi[Q] = (-, success) | (-, deadlock)

$$C_{Linda} = \begin{cases} \texttt{out} & \longmapsto \ \texttt{pout} \\ \texttt{rd} & \longmapsto \ \texttt{prd} \\ \texttt{in} & \longmapsto \ \texttt{pin} \end{cases}$$

$$\text{pLINDA} \succeq_p \text{LINDA}$$

upLINDA vs. uLINDA vs. pLINDA

Reactive Process = uLINDA
= pLINDA

upLINDA = uLINDA + pLINDA
= Generative Process

$$\begin{array}{ccc} \text{LINDA} & \longleftarrow & \text{pLINDA} \\ \uparrow & & \uparrow \\ \text{uLINDA} & \longleftarrow & \text{upLINDA} \end{array}$$

Shapiro, E.: Separating Concurrent Languages with Categories of Language Embeddings. 23rd ACM Symposium on Theory of Computing.

De Boer, F.S., Palamidessi, C.: Embedding as a Tool for Language Comparison. Information and Computation (1994).

Bengt, J., Larsen, J.K., Yi, W.: Probabilistic Extensions of Process Algebras. Handbook of Process Algebra (2001).

28th ACM Symposium on Applied Computing (Coimbra, Portugal - March 18-22, 2013)