

# SEAL: Self Adaptive Agents Specific Language

Marco Tinacci

10 maggio 2012

## Indice

<b>1</b>	<b>Syntax</b>	<b>1</b>
1.1	Syntactic sugar . . . . .	1
<b>2</b>	<b>Semantic</b>	<b>2</b>
<b>3</b>	<b>Examples</b>	<b>3</b>

## 1 Syntax

$S ::= m \mid S \parallel_A S$	(System)
$\gamma ::= \beta \Rightarrow \delta \mid \gamma, \gamma$	(Rule)
$\beta ::= \exists m x : (\beta) \mid e \bowtie e \mid \mathbf{tt} \mid \beta \vee \beta \mid \neg \beta$	(Condition)
$\delta ::= \langle e, a \rangle \alpha \mid \delta \oplus \delta$	(Distribution)
$\alpha ::= x = e; \mid \mathbf{noaction}; \mid \alpha \alpha$	(Action)

Sono state omesse le descrizioni di alcuni simboli per alleggerire la lettura della sintassi:

- $m$ : riferimento alla definizione di un modulo,
- $A$ : insieme di azioni di sincronizzazione,
- $e$ : espressione,
- $\bowtie$ : operatore di confronto,
- $x$ : dichiarazione di un identificatore,
- $a$ : label azione.

### 1.1 Syntactic sugar

Inseriamo un costrutto tale da poter inserire una condizione di abilitazione sugli elementi del supporto della distribuzione:

$$\beta \Rightarrow \langle e, a, \beta' \rangle \alpha \oplus \delta \equiv \beta \wedge \beta' \Rightarrow \langle e, a \rangle \alpha \oplus \delta, \beta \wedge \neg \beta' \Rightarrow \delta$$

## 2 Semantic

Andiamo a dare un'introduzione informale alla semantica del linguaggio descritto:

- *System*: un sistema può essere definito tramite un singolo modulo ( $m$  è un riferimento alla definizione di un modulo) o attraverso la composizione parallela di più sistemi su di un insieme di azioni di sincronizzazione  $A \subseteq Act$ ;
- *Rule*: un insieme di regole definisce il comportamento di un modulo, se vale la condizione  $\beta$  si può passare alla valutazione della distribuzione  $\delta$ ;
- *Condition*: descrive una condizione fornendo anche un operatore di quantificatore esistenziale sui moduli;
- *Distribution*: descrive una distribuzione probabilistica di azioni  $\alpha$ , dove ogni azione è accompagnata da un'espressione  $e$ , che ne descrive il peso, e un'azione  $a$ ;
- *Action*: un azione descrive un aggiornamento dello stato, che può consistere nell'assegnamento di una, nessuna, o più variabili.

Definiamo la semantica del linguaggio in termini di *Markov Decision Processes*. Alla definizione di ogni modulo sarà assegnata una *MDP* della forma:

$$(\Sigma, Act, \rightarrow_\rho, \sigma_0)$$

dove

- $\Sigma \triangleq \{\sigma \mid \sigma : \mathbb{VAR} \rightarrow \mathbb{VAL}\}$  è l'insieme degli *stati* rappresentati da funzioni che mappano variabili in valori,
- $Act$  l'insieme delle azioni,
- $\rightarrow_\rho \subseteq \Sigma \times Act \times Dist(U)$  è la relazione di *avanzamento* di stato,
- $\sigma_0 \in \Sigma$  è lo *stato iniziale*,
- $\rho \subseteq \beta \times Act \times 2^{Dist(U)}$  è la *struttura statica* del *MDP*,
- $U \triangleq \{u \mid u : \Sigma \rightarrow \Sigma\}$  è l'insieme delle funzioni *update* di aggiornamento di stato.

$$\frac{(g, a, \mathcal{D}) \in \rho}{\sigma \xrightarrow{a}_\rho d(\sigma)} \sigma \models g, d \in \mathcal{D} \quad (\text{Update})$$

$$\begin{array}{c}
\frac{S_1 \xrightarrow{a} \Pi_1 \quad S_2 \xrightarrow{a}_\rho \Pi_2}{S_1 \parallel_A S_2 \xrightarrow{a} \Pi_1 \parallel_A \Pi_2} a \in A \quad (\text{Sync}) \\
\\
\frac{S_1 \xrightarrow{a} \Pi_1}{S_1 \parallel_A S_2 \xrightarrow{a}_\rho \Pi_1 \parallel_A S_2} a \notin A \quad (\text{Async 1}) \\
\\
\frac{S_2 \xrightarrow{a} \Pi_2}{S_1 \parallel_A S_2 \xrightarrow{a}_\rho S_1 \parallel_A \Pi_2} a \notin A \quad (\text{Async 2})
\end{array}$$

### 3 Examples

Esempio di un modulo di robot che esegue una *random walk* su una griglia escludendo dalla scelta probabilistica le direzioni adiacenti occupate:

$$\begin{aligned}
m_1 &\triangleq \mathbf{tt} \Rightarrow \\
&< 1, north, \neg \exists m_1 v : v.x = x \wedge v.y = y + 1 > y = y + 1; \oplus \\
&< 1, south, \neg \exists m_1 v : v.x = x \wedge v.y = y - 1 > y = y - 1; \oplus \\
&< 1, east, \neg \exists m_1 v : v.x = x + 1 \wedge v.y = y > x = x + 1; \oplus \\
&< 1, west, \neg \exists m_1 v : v.x = x - 1 \wedge v.y = y > x = x - 1; \oplus \\
&< 1, stay, \mathbf{tt} > \mathbf{noaction};
\end{aligned}$$

Esempio di un modulo di robot analogo al precedente con la differenza che la scelta della mossa viene fatta in modo nondeterministico:

$$\begin{aligned}
m_2 &\triangleq \neg \exists m_1 v : (v.x = x \wedge v.y = y + 1) \Rightarrow < 1, north > y = y + 1; \\
&\neg \exists m_1 v : (v.x = x \wedge v.y = y - 1) \Rightarrow < 1, south > y = y - 1; \\
&\neg \exists m_1 v : (v.x = x + 1 \wedge v.y = y) \Rightarrow < 1, east > x = x + 1; \\
&\neg \exists m_1 v : (v.x = x - 1 \wedge v.y = y) \Rightarrow < 1, west > x = x - 1; \\
&\mathbf{tt} \Rightarrow < 1, stay > \mathbf{noaction};
\end{aligned}$$