

Reactive Systems: Modelling, Specification and Verification

EWSCS'07–Lecture 9

- Hennessy-Milner logic with recursively defined variables (continued)
- Game semantics
- Temporal properties of reactive systems

Syntax of Formulae

Formulae are given by the following abstract syntax

$$F ::= X \mid tt \mid ff \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle a \rangle F \mid [a]F$$

where $a \in Act$ and X is a distinguished variable with a definition

- $X \stackrel{\min}{=} F_X$, or $X \stackrel{\max}{=} F_X$

such that F_X is a formula of the logic (can contain X).

How Can We Define Semantics?

For every formula F we define a function $O_F : 2^{Proc} \rightarrow 2^{Proc}$ s.t.

- if S is the set of processes that satisfy X then
- $O_F(S)$ is the set of processes that satisfy F .

Syntax of Formulae

Formulae are given by the following abstract syntax

$$F ::= X \mid tt \mid ff \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle a \rangle F \mid [a]F$$

where $a \in Act$ and X is a distinguished variable with a definition

- $X \stackrel{\min}{=} F_X$, or $X \stackrel{\max}{=} F_X$

such that F_X is a formula of the logic (can contain X).

How Can We Define Semantics?

For every formula F we define a function $O_F : 2^{Proc} \rightarrow 2^{Proc}$ s.t.

- if S is the set of processes that satisfy X then
- $O_F(S)$ is the set of processes that satisfy F .

Definition of $O_F : 2^{Proc} \rightarrow 2^{Proc}$ (let $S \subseteq 2^{Proc}$)

$$O_X(S) = S$$

$$O_{tt}(S) = Proc$$

$$O_{ff}(S) = \emptyset$$

$$O_{F_1 \wedge F_2}(S) = O_{F_1}(S) \cap O_{F_2}(S)$$

$$O_{F_1 \vee F_2}(S) = O_{F_1}(S) \cup O_{F_2}(S)$$

$$O_{\langle a \rangle F}(S) = \langle a \cdot \rangle O_F(S)$$

$$O_{[a]F}(S) = [a \cdot] O_F(S)$$

O_F is monotonic for every formula F

$$S_1 \subseteq S_2 \Rightarrow O_F(S_1) \subseteq O_F(S_2)$$

Proof: Easy (by induction on the structure of F).

Definition of $O_F : 2^{Proc} \rightarrow 2^{Proc}$ (let $S \subseteq 2^{Proc}$)

$$O_X(S) = S$$

$$O_{tt}(S) = Proc$$

$$O_{ff}(S) = \emptyset$$

$$O_{F_1 \wedge F_2}(S) = O_{F_1}(S) \cap O_{F_2}(S)$$

$$O_{F_1 \vee F_2}(S) = O_{F_1}(S) \cup O_{F_2}(S)$$

$$O_{\langle a \rangle F}(S) = \langle a \cdot \rangle O_F(S)$$

$$O_{[a]F}(S) = [a \cdot] O_F(S)$$

O_F is monotonic for every formula F

$$S_1 \subseteq S_2 \Rightarrow O_F(S_1) \subseteq O_F(S_2)$$

Proof: Easy (by induction on the structure of F).

Observation

We know O_F is **monotonic**, so O_F has a unique **greatest and least fixed point**.

Semantics of the Variable X

- If $X \stackrel{\max}{=} F_X$ then

$$\llbracket X \rrbracket = \bigcup \{S \subseteq Proc \mid S \subseteq O_{F_X}(S)\}.$$

- If $X \stackrel{\min}{=} F_X$ then

$$\llbracket X \rrbracket = \bigcap \{S \subseteq Proc \mid O_{F_X}(S) \subseteq S\}.$$

Game Characterization

Intuition: the attacker claims $s \not\models F$, the defender claims $s \models F$.

Configurations of the game are of the form (s, F)

- (s, tt) and (s, ff) have no successors
- (s, X) has one successor (s, F_X)
- $(s, F_1 \wedge F_2)$ has two successors (s, F_1) and (s, F_2)
(selected by the attacker)
- $(s, F_1 \vee F_2)$ has two successors (s, F_1) and (s, F_2)
(selected by the defender)
- $(s, [a]F)$ has successors (s', F) for every s' s.t. $s \xrightarrow{a} s'$
(selected by the attacker)
- $(s, \langle a \rangle F)$ has successors (s', F) for every s' s.t. $s \xrightarrow{a} s'$
(selected by the defender)

Who is the Winner?

Play is a maximal sequence of configurations formed according to the rules given on the previous slide.

Finite Play

- The **attacker** is the winner of a finite play if the defender gets stuck or the players reach a configuration (s, ff) .
- The **defender** is the winner of a finite play if the attacker gets stuck or the players reach a configuration (s, tt) .

Infinite Play

- The **attacker** is the winner of an infinite play if X is defined as $X \stackrel{\min}{=} F_X$.
- The **defender** is the winner of an infinite play if X is defined as $X \stackrel{\max}{=} F_X$.

Who is the Winner?

Play is a maximal sequence of configurations formed according to the rules given on the previous slide.

Finite Play

- The **attacker** is the winner of a finite play if the defender gets stuck or the players reach a configuration (s, ff) .
- The **defender** is the winner of a finite play if the attacker gets stuck or the players reach a configuration (s, tt) .

Infinite Play

- The **attacker** is the winner of an infinite play if X is defined as $X \stackrel{\min}{=} F_X$.
- The **defender** is the winner of an infinite play if X is defined as $X \stackrel{\max}{=} F_X$.

Theorem

- $s \models F$ if and only if the defender has a universal winning strategy from (s, F)
- $s \not\models F$ if and only if the attacker has a universal winning strategy from (s, F)

Selection of Temporal Properties

- $Inv(F)$: $X \stackrel{\max}{\equiv} F \wedge [Act]X$
- $Pos(F)$: $X \stackrel{\min}{\equiv} F \vee \langle Act \rangle X$
- $Safe(F)$: $X \stackrel{\max}{\equiv} F \wedge ([Act]ff \vee \langle Act \rangle X)$
- $Even(F)$: $X \stackrel{\min}{\equiv} F \vee (\langle Act \rangle tt \wedge [Act]X)$
- $F U^w G$: $X \stackrel{\max}{\equiv} G \vee (F \wedge [Act]X)$
- $F U^s G$: $X \stackrel{\min}{\equiv} G \vee (F \wedge \langle Act \rangle tt \wedge [Act]X)$

Using until we can express e.g. $Inv(F)$ and $Even(F)$:

$$Inv(F) \equiv F U^w ff$$

$$Even(F) \equiv tt U^s F$$

Selection of Temporal Properties

- $Inv(F)$: $X \stackrel{\max}{\equiv} F \wedge [Act]X$
- $Pos(F)$: $X \stackrel{\min}{\equiv} F \vee \langle Act \rangle X$
- $Safe(F)$: $X \stackrel{\max}{\equiv} F \wedge ([Act]ff \vee \langle Act \rangle X)$
- $Even(F)$: $X \stackrel{\min}{\equiv} F \vee (\langle Act \rangle tt \wedge [Act]X)$
- $F U^w G$: $X \stackrel{\max}{\equiv} G \vee (F \wedge [Act]X)$
- $F U^s G$: $X \stackrel{\min}{\equiv} G \vee (F \wedge \langle Act \rangle tt \wedge [Act]X)$

Using until we can express e.g. $Inv(F)$ and $Even(F)$:

$$Inv(F) \equiv F U^w ff$$

$$Even(F) \equiv tt U^s F$$

Selection of Temporal Properties

- $Inv(F)$: $X \stackrel{\max}{\equiv} F \wedge [Act]X$
- $Pos(F)$: $X \stackrel{\min}{\equiv} F \vee \langle Act \rangle X$
- $Safe(F)$: $X \stackrel{\max}{\equiv} F \wedge ([Act]ff \vee \langle Act \rangle X)$
- $Even(F)$: $X \stackrel{\min}{\equiv} F \vee (\langle Act \rangle tt \wedge [Act]X)$
- $F U^w G$: $X \stackrel{\max}{\equiv} G \vee (F \wedge [Act]X)$
- $F U^s G$: $X \stackrel{\min}{\equiv} G \vee (F \wedge \langle Act \rangle tt \wedge [Act]X)$

Using until we can express e.g. $Inv(F)$ and $Even(F)$:

$$Inv(F) \equiv F U^w ff$$

$$Even(F) \equiv tt U^s F$$

Selection of Temporal Properties

- $Inv(F)$: $X \stackrel{\max}{\equiv} F \wedge [Act]X$
- $Pos(F)$: $X \stackrel{\min}{\equiv} F \vee \langle Act \rangle X$
- $Safe(F)$: $X \stackrel{\max}{\equiv} F \wedge ([Act]ff \vee \langle Act \rangle X)$
- $Even(F)$: $X \stackrel{\min}{\equiv} F \vee (\langle Act \rangle tt \wedge [Act]X)$
- $F \mathcal{U}^w G$: $X \stackrel{\max}{\equiv} G \vee (F \wedge [Act]X)$
- $F \mathcal{U}^s G$: $X \stackrel{\min}{\equiv} G \vee (F \wedge \langle Act \rangle tt \wedge [Act]X)$

Using until we can express e.g. $Inv(F)$ and $Even(F)$:

$$Inv(F) \equiv F \mathcal{U}^w ff$$

$$Even(F) \equiv tt \mathcal{U}^s F$$

Examples of More Advanced Recursive Formulae

Nested Definitions of Recursive Variables

$$X \stackrel{\min}{=} Y \vee \langle \text{Act} \rangle X \qquad Y \stackrel{\max}{=} \langle a \rangle tt \wedge \langle \text{Act} \rangle Y$$

Solution: Compute first $\llbracket Y \rrbracket$ and then $\llbracket X \rrbracket$.

Mutually Recursive Definitions

$$X \stackrel{\max}{=} [a]Y \qquad Y \stackrel{\max}{=} \langle a \rangle X$$

Solution: Consider the complete lattice $(2^{\text{Proc}} \times 2^{\text{Proc}}, \sqsubseteq)$ where $(S_1, S_2) \sqsubseteq (S'_1, S'_2)$ iff $S_1 \subseteq S'_1$ and $S_2 \subseteq S'_2$.

Theorem (Characteristic Property for Finite-State Processes)

Let s be a process with finitely many reachable states. There exists a property X_s s.t. for all processes t : $s \sim t$ if and only if $t \in \llbracket X_s \rrbracket$.

Examples of More Advanced Recursive Formulae

Nested Definitions of Recursive Variables

$$X \stackrel{\min}{=} Y \vee \langle \text{Act} \rangle X \qquad Y \stackrel{\max}{=} \langle a \rangle tt \wedge \langle \text{Act} \rangle Y$$

Solution: Compute first $\llbracket Y \rrbracket$ and then $\llbracket X \rrbracket$.

Mutually Recursive Definitions

$$X \stackrel{\max}{=} [a]Y \qquad Y \stackrel{\max}{=} \langle a \rangle X$$

Solution: Consider the complete lattice $(2^{\text{Proc}} \times 2^{\text{Proc}}, \sqsubseteq)$ where $(S_1, S_2) \sqsubseteq (S'_1, S'_2)$ iff $S_1 \subseteq S'_1$ and $S_2 \subseteq S'_2$.

Theorem (Characteristic Property for Finite-State Processes)

Let s be a process with finitely many reachable states. There exists a property X_s s.t. for all processes t : $s \sim t$ if and only if $t \in \llbracket X_s \rrbracket$.

Examples of More Advanced Recursive Formulae

Nested Definitions of Recursive Variables

$$X \stackrel{\min}{=} Y \vee \langle \text{Act} \rangle X \qquad Y \stackrel{\max}{=} \langle a \rangle tt \wedge \langle \text{Act} \rangle Y$$

Solution: Compute first $\llbracket Y \rrbracket$ and then $\llbracket X \rrbracket$.

Mutually Recursive Definitions

$$X \stackrel{\max}{=} [a]Y \qquad Y \stackrel{\max}{=} \langle a \rangle X$$

Solution: Consider the complete lattice $(2^{\text{Proc}} \times 2^{\text{Proc}}, \sqsubseteq)$ where $(S_1, S_2) \sqsubseteq (S'_1, S'_2)$ iff $S_1 \subseteq S'_1$ and $S_2 \subseteq S'_2$.

Theorem (Characteristic Property for Finite-State Processes)

Let s be a process with finitely many reachable states. There exists a property X_s s.t. for all processes t : $s \sim t$ if and only if $t \in \llbracket X_s \rrbracket$.