Reactive Systems: Modelling, Specification and Verification

EWSCS'07–Lecture 8

- Hennessy-Milner logic and temporal properties
- Hennessy-Milner logic with recursively defined variables
- Temporal properties of reactive systems

Is Hennessy-Milner Logic Powerful Enough?

Modal depth (nesting degree) for Hennessy-Milner formulae:

- $md(tt) = md(ff) = 0$
- $md(F \wedge G) = md(F \vee G) = \max\{md(F), md(G)\}$
- $md([a]F) = md(\langle a \rangle F) = md(F) + 1$

Idea: a formula F can "see" only up to depth $md(F)$.

Theorem (let F be a HM formula and $k = md(F)$)

If the defender has a defending strategy in the strong bisimulation game from s and t up to k rounds then $s \models F$ if and only if $t \models F$.

Conclusion

There is no Hennessy-Milner formula F that can detect a deadlock in an arbitrary LTS.

Is Hennessy-Milner Logic Powerful Enough?

Modal depth (nesting degree) for Hennessy-Milner formulae:

- $md(tt) = md(ff) = 0$
- $md(F \wedge G) = md(F \vee G) = \max\{md(F), md(G)\}$
- $md([a]F) = md(\langle a \rangle F) = md(F) + 1$

Idea: a formula F can "see" only up to depth $md(F)$.

Theorem (let F be a HM formula and $k = md(F)$)

If the defender has a defending strategy in the strong bisimulation game from s and t up to k rounds then $s \models F$ if and only if $t \models F$.

Conclusion

There is no Hennessy-Milner formula F that can detect a deadlock in an arbitrary LTS.

Is Hennessy-Milner Logic Powerful Enough?

Modal depth (nesting degree) for Hennessy-Milner formulae:

- $md(tt) = md(ff) = 0$
- $md(F \wedge G) = md(F \vee G) = \max\{md(F), md(G)\}$
- $md([a]F) = md(\langle a \rangle F) = md(F) + 1$

Idea: a formula F can "see" only up to depth $md(F)$.

Theorem (let F be a HM formula and $k = md(F)$)

If the defender has a defending strategy in the strong bisimulation game from s and t up to k rounds then $s \models F$ if and only if $t \models F$.

Conclusion

There is no Hennessy-Milner formula F that can detect a deadlock in an arbitrary LTS.

Temporal Properties not Expressible in HM Logic

$s \models Inv(F)$ iff all states reachable from s satisfy F
$s \models Pos(F)$ iff there is a reachable state which satisfies F

Fact

Properties $Inv(F)$ and $Pos(F)$ are not expressible in HM logic.

Let $Act = \{a_1, a_2, \ldots, a_n\}$ be a finite set of actions. We define

- $\langle Act \rangle F \stackrel{\text{def}}{=} \langle a_1 \rangle F \vee \langle a_2 \rangle F \vee \ldots \vee \langle a_n \rangle F$
- $[Act] F \stackrel{\text{def}}{=} [a_1] F \wedge [a_2] F \wedge \ldots \wedge [a_n] F$

$Inv(F) \equiv F \wedge [Act]F \wedge [Act][Act]F \wedge [Act][Act][Act]F \wedge \ldots$
$Pos(F) \equiv F \vee \langle Act \rangle F \vee \langle Act \rangle \langle Act \rangle F \vee \langle Act \rangle \langle Act \rangle \langle Act \rangle F \vee \ldots$

Temporal Properties not Expressible in HM Logic

$s \models Inv(F)$ iff all states reachable from s satisfy F
$s \models Pos(F)$ iff there is a reachable state which satisfies F

Fact

Properties $Inv(F)$ and $Pos(F)$ are not expressible in HM logic.

Let $Act = \{a_1, a_2, \ldots, a_n\}$ be a finite set of actions. We define

- $\langle Act \rangle F \stackrel{\text{def}}{=} \langle a_1 \rangle F \vee \langle a_2 \rangle F \vee \ldots \vee \langle a_n \rangle F$
- $[Act] F \stackrel{\text{def}}{=} [a_1] F \wedge [a_2] F \wedge \ldots \wedge [a_n] F$

$Inv(F) \equiv F \wedge [Act]F \wedge [Act][Act]F \wedge [Act][Act][Act]F \wedge \ldots$
$Pos(F) \equiv F \vee \langle Act \rangle F \vee \langle Act \rangle \langle Act \rangle F \vee \langle Act \rangle \langle Act \rangle \langle Act \rangle F \vee \ldots$

Temporal Properties not Expressible in HM Logic

$s \models Inv(F)$ iff all states reachable from s satisfy F
$s \models Pos(F)$ iff there is a reachable state which satisfies F

Fact

Properties $Inv(F)$ and $Pos(F)$ are not expressible in HM logic.

Let $Act = \{a_1, a_2, \ldots, a_n\}$ be a finite set of actions. We define

- $\langle Act \rangle F \stackrel{\text{def}}{=} \langle a_1 \rangle F \vee \langle a_2 \rangle F \vee \ldots \vee \langle a_n \rangle F$
- $[Act] F \stackrel{\text{def}}{=} [a_1] F \wedge [a_2] F \wedge \ldots \wedge [a_n] F$

$Inv(F) \equiv F \wedge [Act]F \wedge [Act][Act]F \wedge [Act][Act][Act]F \wedge \ldots$
$Pos(F) \equiv F \vee \langle Act \rangle F \vee \langle Act \rangle \langle Act \rangle F \vee \langle Act \rangle \langle Act \rangle \langle Act \rangle F \vee \ldots$

Infinite Conjunctions and Disjunctions vs. Recursion

Problems

- infinite formulae are not allowed in HM logic
- infinite formulae are difficult to handle

Why don't we use recursion?

- $Inv(F)$ expressed by $X \stackrel{\text{def}}{=} F \wedge [Act]X$
- $Pos(F)$ expressed by $X \stackrel{\text{def}}{=} F \vee \langle Act \rangle X$

Question: How to define the semantics of such equations?

Infinite Conjunctions and Disjunctions vs. Recursion

Problems

- infinite formulae are not allowed in HM logic
- infinite formulae are difficult to handle

Why don't we use recursion?

- $Inv(F)$ expressed by $X \stackrel{\text{def}}{=} F \wedge [Act]X$
- $Pos(F)$ expressed by $X \stackrel{\text{def}}{=} F \vee \langle Act \rangle X$

Question: How to define the semantics of such equations?

Infinite Conjunctions and Disjunctions vs. Recursion

Problems

- infinite formulae are not allowed in HM logic
- infinite formulae are difficult to handle

Why don't we use recursion?

- $Inv(F)$ expressed by $X \stackrel{\text{def}}{=} F \wedge [Act]X$
- $Pos(F)$ expressed by $X \stackrel{\text{def}}{=} F \vee \langle Act \rangle X$

Question: How to define the semantics of such equations?

Solving Equations is Tricky

Equations over Natural Numbers ($n \in \mathbb{N}$)

$n = 2 * n$	one solution $n = 0$
$n = n + 1$	no solution
$n = 1 * n$	many solutions (every $n \in Nat$ is a solution)

Equations over Sets of Integers ($M \in 2^{\mathbb{N}}$)

$M = (\{7\} \cap M) \cup \{7\}$	one solution $M = \{7\}$
$M = \mathbb{N} \setminus M$	no solution
$M = \{3\} \cup M$	many solutions (every $M \supseteq \{3\}$)

What about Equations over Processes?

$X \stackrel{\text{def}}{=} [a]ff \vee \langle a \rangle X \quad \Rightarrow \quad$ find $S \subseteq 2^{Proc}$ s.t. $S = [\cdot a \cdot]\emptyset \cup \langle \cdot a \cdot \rangle S$

Solving Equations is Tricky

Equations over Natural Numbers ($n \in \mathbb{N}$)

$n = 2 * n$	one solution $n = 0$
$n = n + 1$	no solution
$n = 1 * n$	many solutions (every $n \in Nat$ is a solution)

Equations over Sets of Integers ($M \in 2^{\mathbb{N}}$)

$M = (\{7\} \cap M) \cup \{7\}$	one solution $M = \{7\}$
$M = \mathbb{N} \setminus M$	no solution
$M = \{3\} \cup M$	many solutions (every $M \supseteq \{3\}$)

What about Equations over Processes?

$X \stackrel{\text{def}}{=} [a]ff \vee \langle a \rangle X \quad \Rightarrow \quad$ find $S \subseteq 2^{Proc}$ s.t. $S = [\cdot a \cdot]\emptyset \cup \langle \cdot a \cdot \rangle S$

Solving Equations is Tricky

Equations over Natural Numbers ($n \in \mathbb{N}$)

$n = 2 * n$	one solution $n = 0$
$n = n + 1$	no solution
$n = 1 * n$	many solutions (every $n \in Nat$ is a solution)

Equations over Sets of Integers ($M \in 2^{\mathbb{N}}$)

$M = (\{7\} \cap M) \cup \{7\}$	one solution $M = \{7\}$
$M = \mathbb{N} \setminus M$	no solution
$M = \{3\} \cup M$	many solutions (every $M \supseteq \{3\}$)

What about Equations over Processes?

$X \stackrel{\text{def}}{=} [a]ff \vee \langle a \rangle X \quad \Rightarrow \quad$ find $S \subseteq 2^{Proc}$ s.t. $S = [\cdot a \cdot]\emptyset \cup \langle \cdot a \cdot \rangle S$

Monotonic Functions

Monotonic Function and Fixed Points

A function $f : 2^{Proc} \longrightarrow 2^{Proc}$ is called monotonic iff

$$X \subseteq Y \quad \Rightarrow \quad f(X) \subseteq f(Y)$$

for all $X, Y \in 2^{Proc}$.

A set $X \in 2^{Proc}$ is called a fixed point of f iff $X = f(X)$.

Questions

Is the function $f(X) = X \cup \{s, t\}$ monotonic? What about $g(X) = Proc \setminus X$? Do these functions have fixed points?

Tarski's Fixed Point Theorem

Theorem (Tarski)

Let $f : 2^{Proc} \longrightarrow 2^{Proc}$ be a monotonic function.
Then f has a unique largest fixed point z_{max} and a unique least fixed point z_{min} given by:

$$z_{max} \stackrel{\text{def}}{=} \bigcup \{X \in 2^{Proc} \mid X \subseteq f(X)\}$$

$$z_{min} \stackrel{\text{def}}{=} \bigcap \{X \in 2^{Proc} \mid f(X) \subseteq X\}$$

Computing Min and Max Fixed Points on Finite Sets

Let $f : 2^{Proc} \longrightarrow 2^{Proc}$ be monotonic.
Let $f^1(X) \stackrel{\text{def}}{=} f(X)$ and $f^n(X) \stackrel{\text{def}}{=} f(f^{n-1}(X))$ for $n > 1$, i.e.,

$$f^n(X) = \underbrace{f(f(\ldots f}_{n \text{ times}}(X)\ldots)).$$

Theorem

If 2^{Proc} is a finite set then there exist integers $M, m > 0$ such that

- $z_{max} = f^M(Proc)$
- $z_{min} = f^m(\emptyset)$

Idea (for z_{min}): The following sequence stabilizes for any finite 2^{Proc}

$$\emptyset \subseteq f(\emptyset) \subseteq f(f(\emptyset)) \subseteq f(f(f(\emptyset))) \subseteq \cdots$$

Computing Min and Max Fixed Points on Finite Sets

Let $f : 2^{Proc} \longrightarrow 2^{Proc}$ be monotonic.
Let $f^1(X) \stackrel{\text{def}}{=} f(X)$ and $f^n(X) \stackrel{\text{def}}{=} f(f^{n-1}(X))$ for $n > 1$, i.e.,

$$f^n(X) = \underbrace{f(f(\ldots f}_{n \text{ times}}(X)\ldots)).$$

Theorem

If 2^{Proc} is a finite set then there exist integers $M, m > 0$ such that

- $z_{max} = f^M(Proc)$
- $z_{min} = f^m(\emptyset)$

Idea (for z_{min}): The following sequence stabilizes for any finite 2^{Proc}

$$\emptyset \subseteq f(\emptyset) \subseteq f(f(\emptyset)) \subseteq f(f(f(\emptyset))) \subseteq \cdots$$

HML with One Recursively Defined Variable

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 to Define Semantics?

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

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

HML with One Recursively Defined Variable

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 to Define Semantics?

For every formula F we define a function $O_F : 2^{Proc} \longrightarrow 2^{Proc}$ 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 $O_F : 2^{Proc} \longrightarrow 2^{Proc}$ (let $S \subseteq 2^{Proc}$)

$$
\begin{aligned}
O_X(S) &= S \\
O_{tt}(S) &= Proc \\
O_{ff}(S) &= \emptyset \\
O_{F_1 \wedge F_2}(S) &= O_{F_1}(S) \cap O_{F_2}(S) \\
O_{F_1 \vee F_2}(S) &= O_{F_1}(S) \cup O_{F_2}(S) \\
O_{\langle a \rangle F}(S) &= \langle \cdot a \cdot \rangle O_F(S) \\
O_{[a]F}(S) &= [\cdot a \cdot] O_F(S)
\end{aligned}
$$

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 (structural induction on the structure of F).

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

$$
\begin{aligned}
O_X(S) &= S \\
O_{tt}(S) &= Proc \\
O_{ff}(S) &= \emptyset \\
O_{F_1 \wedge F_2}(S) &= O_{F_1}(S) \cap O_{F_2}(S) \\
O_{F_1 \vee F_2}(S) &= O_{F_1}(S) \cup O_{F_2}(S) \\
O_{\langle a \rangle F}(S) &= \langle \cdot a \cdot \rangle O_F(S) \\
O_{[a]F}(S) &= [\cdot a \cdot] O_F(S)
\end{aligned}
$$

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 (structural induction on the structure of F).

Semantics

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

$$[\![X]\!] = \bigcup \{S \subseteq Proc \mid S \subseteq O_{F_X}(S)\}.$$

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

$$[\![X]\!] = \bigcap \{S \subseteq Proc \mid O_{F_X}(S) \subseteq S\}.$$

Selection of Temporal Properties

- $Inv(F)$: $X \stackrel{\max}{=} F \wedge [Act]X$
- $Pos(F)$: $X \stackrel{\min}{=} F \vee \langle Act \rangle X$

- $Safe(F)$: $X \stackrel{\max}{=} F \wedge ([Act]ff \vee \langle Act \rangle X)$
- $Even(F)$: $X \stackrel{\min}{=} F \vee (\langle Act \rangle tt \wedge [Act]X)$

- $F\ \mathcal{U}^w\ G$: $X \stackrel{\max}{=} G \vee (F \wedge [Act]X)$
- $F\ \mathcal{U}^s\ G$: $X \stackrel{\min}{=} G \vee (F \wedge \langle Act \rangle tt \wedge [Act]X)$

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

$$Inv(F) \equiv F\ \mathcal{U}^w\ ff \qquad\qquad Even(F) \equiv tt\ \mathcal{U}^s\ F$$

Selection of Temporal Properties

- $\mathit{Inv}(F)$: $X \stackrel{\max}{=} F \wedge [\mathit{Act}]X$
- $\mathit{Pos}(F)$: $X \stackrel{\min}{=} F \vee \langle \mathit{Act} \rangle X$

- $\mathit{Safe}(F)$: $X \stackrel{\max}{=} F \wedge ([\mathit{Act}]\mathit{ff} \vee \langle \mathit{Act} \rangle X)$
- $\mathit{Even}(F)$: $X \stackrel{\min}{=} F \vee (\langle \mathit{Act} \rangle \mathit{tt} \wedge [\mathit{Act}]X)$

- $F \; \mathcal{U}^w \; G$: $X \stackrel{\max}{=} G \vee (F \wedge [\mathit{Act}]X)$
- $F \; \mathcal{U}^s \; G$: $X \stackrel{\min}{=} G \vee (F \wedge \langle \mathit{Act} \rangle \mathit{tt} \wedge [\mathit{Act}]X)$

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

$$\mathit{Inv}(F) \equiv F \; \mathcal{U}^w \; \mathit{ff} \qquad\qquad \mathit{Even}(F) \equiv \mathit{tt} \; \mathcal{U}^s \; F$$

Selection of Temporal Properties

- $Inv(F)$: $X \stackrel{\max}{=} F \wedge [Act]X$
- $Pos(F)$: $X \stackrel{\min}{=} F \vee \langle Act \rangle X$

- $Safe(F)$: $X \stackrel{\max}{=} F \wedge ([Act]ff \vee \langle Act \rangle X)$
- $Even(F)$: $X \stackrel{\min}{=} F \vee (\langle Act \rangle tt \wedge [Act]X)$

- $F\ \mathcal{U}^w\ G$: $X \stackrel{\max}{=} G \vee (F \wedge [Act]X)$
- $F\ \mathcal{U}^s\ G$: $X \stackrel{\min}{=} G \vee (F \wedge \langle Act \rangle tt \wedge [Act]X)$

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

$$Inv(F) \equiv F\ \mathcal{U}^w\ ff \qquad\qquad Even(F) \equiv tt\ \mathcal{U}^s\ F$$

Selection of Temporal Properties

- $Inv(F)$: $X \stackrel{\max}{=} F \wedge [Act]X$
- $Pos(F)$: $X \stackrel{\min}{=} F \vee \langle Act \rangle X$

- $Safe(F)$: $X \stackrel{\max}{=} F \wedge ([Act]ff \vee \langle Act \rangle X)$
- $Even(F)$: $X \stackrel{\min}{=} F \vee (\langle Act \rangle tt \wedge [Act]X)$

- $F\ \mathcal{U}^w\ G$: $X \stackrel{\max}{=} G \vee (F \wedge [Act]X)$
- $F\ \mathcal{U}^s\ G$: $X \stackrel{\min}{=} G \vee (F \wedge \langle Act \rangle tt \wedge [Act]X)$

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

$$Inv(F) \equiv F\ \mathcal{U}^w\ ff \qquad\qquad Even(F) \equiv tt\ \mathcal{U}^s\ F$$