System Validation

Mohammad Mousavi

7. Modal mu-Calculus

Modal μ-Calculus

Mohammad Mousavi

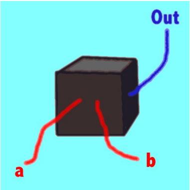
TU/Eindhoven

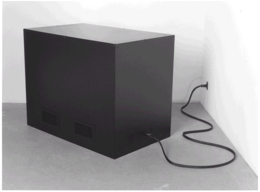
System Validation, 2012-2013
TU Delft

Outline

Specification using Temporal logic

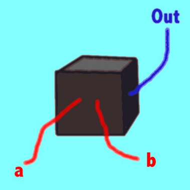
- Fix observable events (interactions with external world)

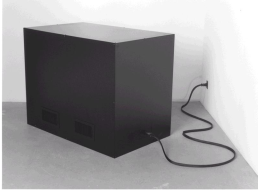




Specification using Temporal logic

- Fix observable events (interactions with external world)

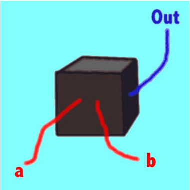


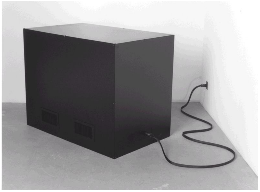


- Describe temporal properties using these

Specification using Temporal logic

- Fix observable events (interactions with external world)





- Describe temporal properties using these
- Verify the correctness of the properties with respect to some labeled transition system

A scientist interacts with its environment

- coffee for taking coffee in
- coin for producing a coin
- pub for producing a publication
- ...

A scientist interacts with its environment

- coffee for taking coffee in
- coin for producing a coin
- pub for producing a publication
- ...

Properties of interest

- the scientist is not willing to drink coffee now

A scientist interacts with its environment

- coffee for taking coffee in
- coin for producing a coin
- pub for producing a publication
- ...

Properties of interest

- the scientist is not willing to drink coffee now
- the scientist is willing to drink both coffee and tea now

A scientist interacts with its environment

- coffee for taking coffee in
- coin for producing a coin
- pub for producing a publication
- ...

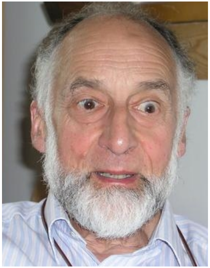
Properties of interest

- the scientist is not willing to drink coffee now
- the scientist is willing to drink both coffee and tea now
- she always produces a publication after drinking coffee

Outline

Hennessy-Milner logic

- Introduced by Hennessy and Milner in 1985



Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. Journal of the ACM, 32(1):137-161, 1985.

Syntax of Hennesy-Milner logic

for $a \in Act$

$$F ::= true \mid false \mid \neg F \mid F \wedge F \mid F \vee F \mid \langle a \rangle F \mid [a]F$$

Syntax of Hennesy-Milner logic

for $a \in Act$

$$F ::= true \mid false \mid \neg F \mid F \wedge F \mid F \vee F \mid \langle a \rangle F \mid [a]F$$

where

- $\langle a \rangle F$ denotes that it is possible to perform action a and thereby (in the next state) satisfy F

Syntax of Hennesy-Milner logic

for $a \in Act$

$$F ::= true \mid false \mid \neg F \mid F \wedge F \mid F \vee F \mid \langle a \rangle F \mid [a]F$$

where

- $\langle a \rangle F$ denotes that it is possible to perform action a and thereby (in the next state) satisfy F
- $[a]F$ denotes that no matter how a process performs action a afterwards necessarily F holds

Syntax of Hennesy-Milner logic

for $a \in Act$

$$F ::= true \mid false \mid \neg F \mid F \wedge F \mid F \vee F \mid \langle a \rangle F \mid [a]F$$

where

- $\langle a \rangle F$ denotes that it is possible to perform action a and thereby (in the next state) satisfy F
- $[a]F$ denotes that no matter how a process performs action a afterwards necessarily F holds
- There is a minimal subset

Syntax of Hennesy-Milner logic

For $A = \{a_1, \cdots, a_n\} \subseteq Act$ with $n \geq 1$

- $\langle A \rangle F$ denotes $\langle a_1 \rangle F \vee \cdots \vee \langle a_n \rangle F$ and $\langle \emptyset \rangle F = false$

Syntax of Hennesy-Milner logic

For $A = \{a_1, \cdots, a_n\} \subseteq \mathit{Act}$ with $n \geq 1$

- $\langle A \rangle F$ denotes $\langle a_1 \rangle F \vee \cdots \vee \langle a_n \rangle F$ and $\langle \emptyset \rangle F = \mathit{false}$
- $[A]F$ denotes $[a_1]F \wedge \cdots \wedge [a_n]F$ and $[\emptyset]F = \mathit{true}$

Syntax of Hennessy-Milner logic

For $A = \{a_1, \cdots, a_n\} \subseteq Act$ with $n \geq 1$

- $\langle A \rangle F$ denotes $\langle a_1 \rangle F \vee \cdots \vee \langle a_n \rangle F$ and $\langle \emptyset \rangle F = \mathit{false}$
- $[A]F$ denotes $[a_1]F \wedge \cdots \wedge [a_n]F$ and $[\emptyset]F = \mathit{true}$

In the book, ‘true’ is also used for ‘Act’.

Examples

- the scientist is not willing to drink coffee now

Examples

- the scientist is not willing to drink coffee now

$$\neg\langle \text{coffee} \rangle \mathit{true} \qquad \text{or} \qquad [\text{coffee}]\mathit{false}$$

Examples

- the scientist is not willing to drink coffee now

$$\neg\langle \text{coffee} \rangle \mathit{true} \qquad \text{or} \qquad [\text{coffee}]\mathit{false}$$

- the scientist is willing to drink both coffee and tea now

Examples

- the scientist is not willing to drink coffee now

$$\neg\langle \text{coffee} \rangle \mathit{true} \qquad \text{or} \qquad [\text{coffee}]\mathit{false}$$

- the scientist is willing to drink both coffee and tea now

$$\langle \text{coffee} \rangle \mathit{true} \wedge \langle \text{tea} \rangle \mathit{true}$$

Examples

- the scientist is not willing to drink coffee now

$$\neg\langle \text{coffee} \rangle \mathit{true} \qquad \text{or} \qquad [\text{coffee}]\mathit{false}$$

- the scientist is willing to drink both coffee and tea now

$$\langle \text{coffee} \rangle \mathit{true} \wedge \langle \text{tea} \rangle \mathit{true}$$

- the scientist is willing to drink coffee, but not tea, now

Examples

- the scientist is not willing to drink coffee now

$$\neg\langle \mathrm{coffee}\rangle \mathit{true} \qquad \text{or} \qquad [\mathrm{coffee}]\mathit{false}$$

- the scientist is willing to drink both coffee and tea now

$$\langle \mathrm{coffee}\rangle \mathit{true} \wedge \langle \mathrm{tea}\rangle \mathit{true}$$

- the scientist is willing to drink coffee, but not tea, now

$$\langle \mathrm{coffee}\rangle \mathit{true} \wedge \neg\langle \mathrm{tea}\rangle \mathit{true}$$

Examples

- the scientist will always produce a publication immediately after having drunk two coffees in a row

Examples

- the scientist will always produce a publication immediately after having drunk two coffees in a row

$$[\text{coffee}][\text{coffee}](\langle \text{pub} \rangle \mathit{true} \wedge [\mathit{Act} \setminus \{\text{pub}\}]\mathit{false})$$

Typical formulas

- the process is deadlocked
- the process can execute some action
- *a* must happen next
- *F* holds after one step

Typical formulas

- the process is deadlocked

$$[Act]\mathit{false}$$

- the process can execute some action

- *a* must happen next

- *F* holds after one step

Typical formulas

- the process is deadlocked

$$[Act]\mathit{false}$$

- the process can execute some action

$$\langle Act\rangle \mathit{true}$$

- a must happen next

- F holds after one step

Typical formulas

- the process is deadlocked

$$[Act]\mathit{false}$$

- the process can execute some action

$$\langle Act\rangle \mathit{true}$$

- *a* must happen next

$$\langle a\rangle \mathit{true} \wedge [Act \setminus \{a\}]\mathit{false}$$

$$\langle Act\rangle \mathit{true} \wedge [Act \setminus \{a\}]\mathit{false}$$

- *F* holds after one step

Typical formulas

- the process is deadlocked

$$[Act]\mathit{false}$$

- the process can execute some action

$$\langle Act\rangle \mathit{true}$$

- *a* must happen next

$$\langle a\rangle \mathit{true} \wedge [Act \setminus \{a\}]\mathit{false}$$

$$\langle Act\rangle \mathit{true} \wedge [Act \setminus \{a\}]\mathit{false}$$

- *F* holds after one step

$$[Act]F \wedge \langle Act\rangle \mathit{true}$$

Material for the Semantics

- This set of slides!
- Chapters 5 and 6 of book 'Reactive Systems – Modelling, Specification and Verification' by L. Aceto, A. Ingólfsdóttir, K. Larsen and J. Srba
- Section 6.4 of the book (and possibly Section 15.3)

Outline

Semantics of HML

With each formula associate a set of states where the formula is valid.

$[\![F]\!] \subseteq S$ is defined inductively by

Semantics of HML

With each formula associate a set of states where the formula is valid.

$[\![F]\!] \subseteq S$ is defined inductively by

1. $[\![\mathit{true}]\!] = S$

Semantics of HML

With each formula associate a set of states where the formula is valid.

$[\![F]\!] \subseteq S$ is defined inductively by

1. $[\![true]\!] = S$
2. $[\![false]\!] = \emptyset$

Semantics of HML

With each formula associate a set of states where the formula is valid.

$[\![F]\!] \subseteq S$ is defined inductively by

1. $[\![\mathit{true}]\!] = S$
2. $[\![\mathit{false}]\!] = \emptyset$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$

Semantics of HML

With each formula associate a set of states where the formula is valid.

$[\![F]\!] \subseteq S$ is defined inductively by

1. $[\![\mathit{true}]\!] = S$
2. $[\![\mathit{false}]\!] = \emptyset$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$
4. $[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$

Semantics of HML

With each formula associate a set of states where the formula is valid.

$[\![F]\!] \subseteq S$ is defined inductively by

1. $[\![\mathit{true}]\!] = S$
2. $[\![\mathit{false}]\!] = \emptyset$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$
4. $[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$
5. $[\![\langle a\rangle F]\!] = \langle\cdot a\cdot\rangle[\![F]\!]$
   where $\langle\cdot a\cdot\rangle$ is defined by
   $\langle\cdot a\cdot\rangle T = \{p \in S \mid \exists p'.\, p \xrightarrow{a} p' \text{ and } p' \in T\}$

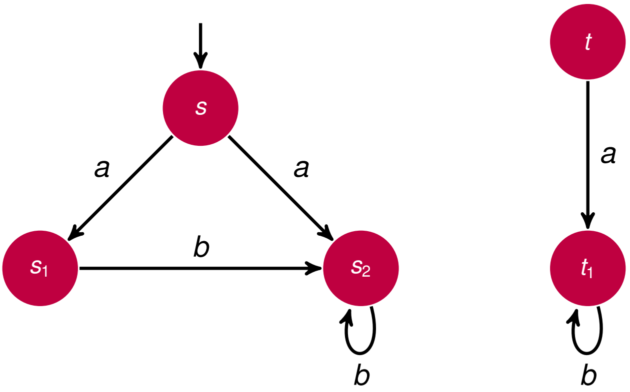
Semantics of HML

With each formula associate a set of states where the formula is valid.

$[\![F]\!] \subseteq S$ is defined inductively by

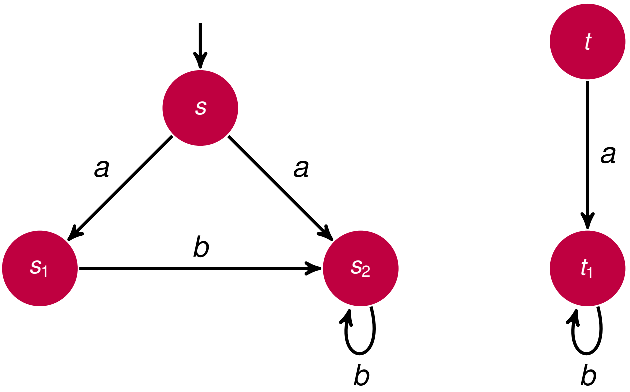
1. $[\![true]\!] = S$
2. $[\![false]\!] = \emptyset$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$
4. $[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$
5. $[\![\langle a \rangle F]\!] = \langle \cdot a \cdot \rangle [\![F]\!]$
   where $\langle \cdot a \cdot \rangle$ is defined by
   $\langle \cdot a \cdot \rangle T = \{p \in S \mid \exists p'.\ p \stackrel{a}{\rightarrow} p' \text{ and } p' \in T\}$
6. $[\![[a]F]\!] = [\cdot a \cdot][\![F]\!]$
   where $[\cdot a \cdot]$ is defined by
   $[\cdot a \cdot]T = \{p \in S \mid \forall p'.\ p \stackrel{a}{\rightarrow} p' \Rightarrow p' \in T\}$

Examples



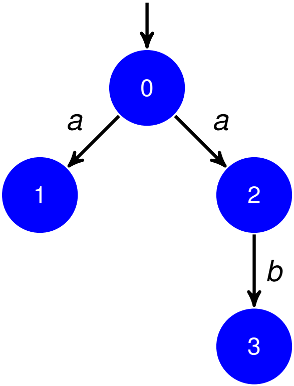
- $\langle \cdot a \cdot \rangle \{s_1, t_1\} = \{s, t\}$

Examples



- $\langle \cdot a \cdot \rangle \{s_1, t_1\} = \{s, t\}$
- $[\cdot a \cdot]\{s_1, t_1\} = \{s_1, s_2, t, t_1\}$

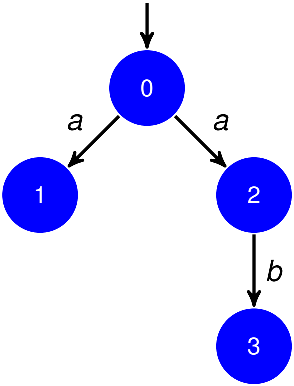
Is the HML formula $\langle a\rangle\langle b\rangle true$ satisfied by the labeled transition system (i.e., by its initial state)?



Subformulas

$true \qquad \langle b\rangle true \qquad \langle a\rangle\langle b\rangle true$

Is the HML formula $\langle a\rangle\langle b\rangle true$ satisfied by the labeled transition system (i.e., by its initial state)?

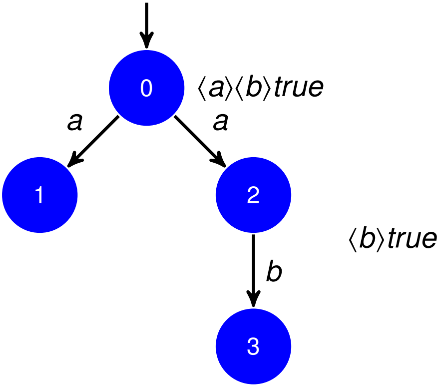


$\langle b\rangle true$

Subformulas

$true \qquad \langle b\rangle true \qquad \langle a\rangle\langle b\rangle true$

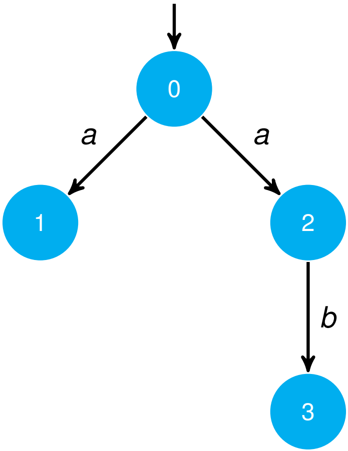
Is the HML formula $\langle a\rangle\langle b\rangle true$ satisfied by the labeled transition system (i.e., by its initial state)?



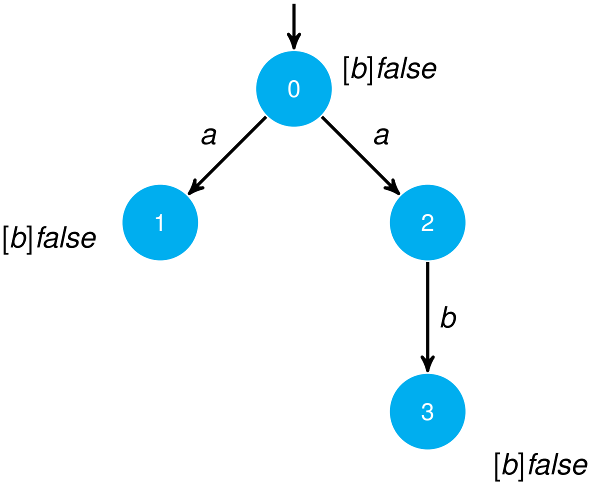
Subformulas

$true \qquad \langle b\rangle true \qquad \langle a\rangle\langle b\rangle true$

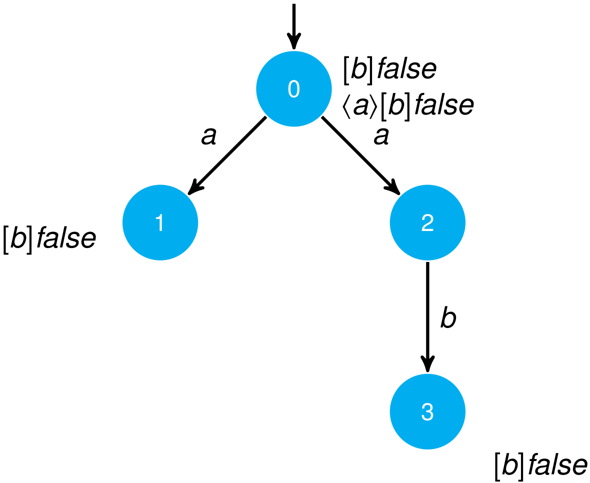
Is the HML formula $\langle a \rangle [b] \mathit{false}$ satisfied?



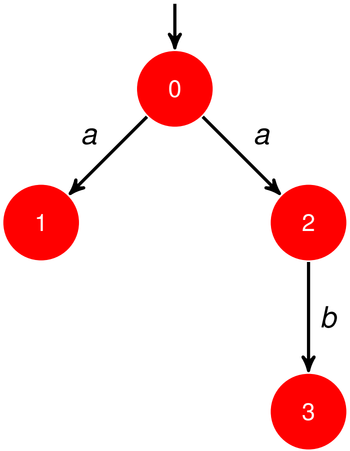
Is the HML formula $\langle a \rangle [b] \mathit{false}$ satisfied?



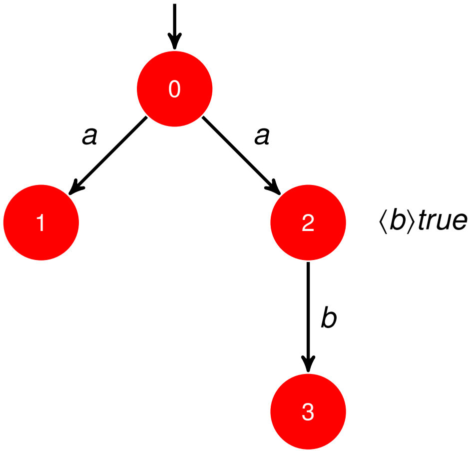
Is the HML formula $\langle a \rangle [b] \mathit{false}$ satisfied?



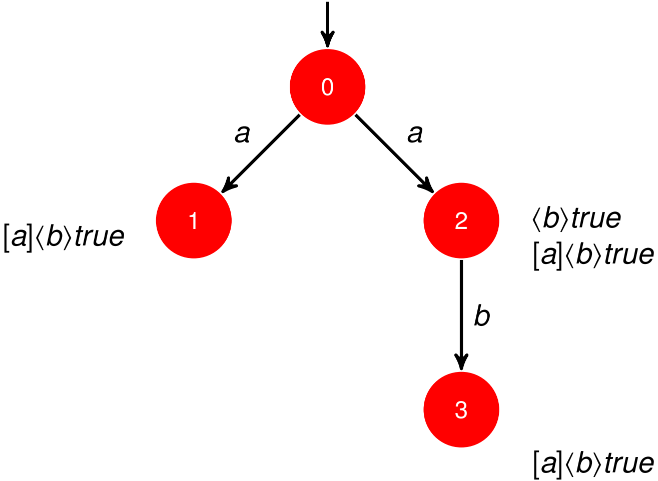
Is the HML formula $[a]\langle b\rangle \mathit{true}$ satisfied?



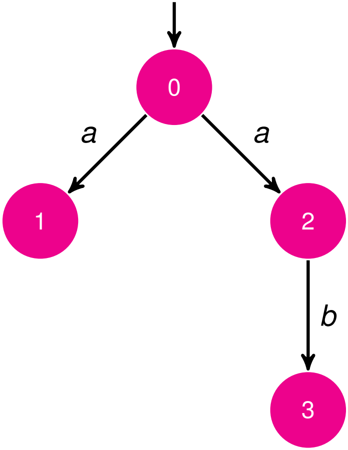
Is the HML formula $[a]\langle b\rangle true$ satisfied?



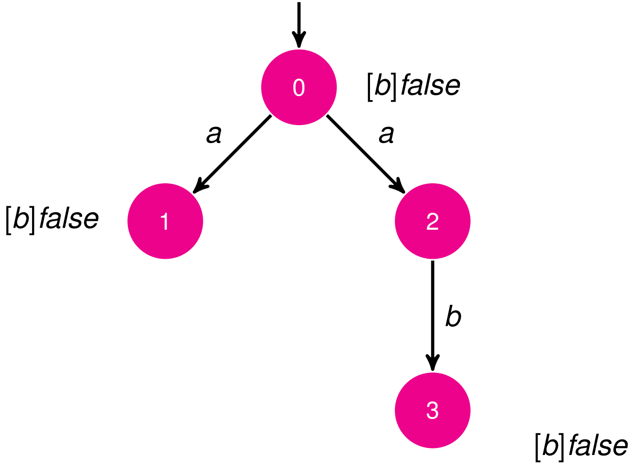
Is the HML formula $[a]\langle b\rangle true$ satisfied?



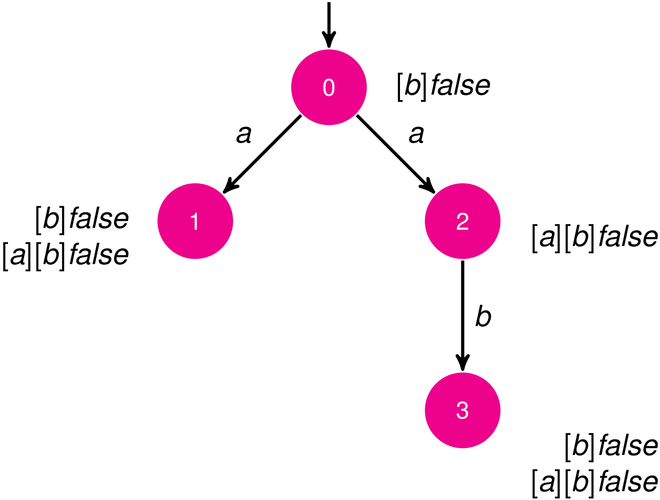
Is the HML formula $[a][b]\mathit{false}$ satisfied?



Is the HML formula $[a][b]\mathit{false}$ satisfied?



Is the HML formula $[a][b]\mathit{false}$ satisfied?



Outline

Temporal logic

Hennessy-Milner logic

Semantics of HML

Recursion

Semantics of Recursion

Limitations of HML

Limited expressiveness of HML

Using Hennessy-Milner Logic we can only describe properties of behaviors with a finite depth.

Modal depth

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

Temporal Properties not Expressible in HML

- $Inv(F)$ iff all reachable states satisfy F

$$Inv(F) = F \wedge [Act]F \wedge [Act][Act]F \wedge [Act][Act][Act]F \wedge \ldots$$

- $Pos(F)$ iff there is a reachable state which satisfies F

$$Pos(F) = F \vee \langle Act\rangle F \vee \langle Act\rangle\langle Act\rangle F \vee \langle Act\rangle\langle Act\rangle\langle Act\rangle F \vee \ldots$$

Temporal Properties not Expressible in HML

- $Inv(F)$ iff all reachable states satisfy F

$$Inv(F) = F \wedge [Act]F \wedge [Act][Act]F \wedge [Act][Act][Act]F \wedge \ldots$$

- $Pos(F)$ iff there is a reachable state which satisfies F

$$Pos(F) = F \vee \langle Act\rangle F \vee \langle Act\rangle\langle Act\rangle F \vee \langle Act\rangle\langle Act\rangle\langle Act\rangle F \vee \ldots$$

Problems

- infinite formulae are not allowed in HML
- infinite formulae are difficult to handle

Why not to use recursion?

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

Why not to use recursion?

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

Recursion on natural numbers

$$n \ : \ n \stackrel{\text{def}}{=} n^2$$

$$n \ : \ n \stackrel{\text{def}}{=} n + 1$$

$$n \ : \ n \stackrel{\text{def}}{=} 1 \times n$$

HML with one recursively defined variable

Syntax of Formulae

Formulae are given by the following abstract syntax

$$F ::= X \mid \mathit{true} \mid \mathit{false} \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle a \rangle F \mid [a]F$$

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

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

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

HML with one recursively defined variable

Syntax of Formulae

Formulae are given by the following abstract syntax

$$F ::= X \mid \mathit{true} \mid \mathit{false} \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle a \rangle F \mid [a]F$$

where $a \in \mathit{Act}$ 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 (which can contain X).

Alternative syntax:

$$\begin{array}{lll} F & ::= & X \mid \mathit{true} \mid \mathit{false} \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle a \rangle F \mid [a]F \\ & \mid & \mu X.F \mid \nu X.F \end{array}$$

Example:

$$X \stackrel{\text{min}}{=} X$$

Any set of states S satisfies the set-equation $X = X$. The least such set is $\emptyset$.

Example:

$$X \overset{\text{min}}{=} X$$

Any set of states S satisfies the set-equation $X = X$. The least such set is $\emptyset$.

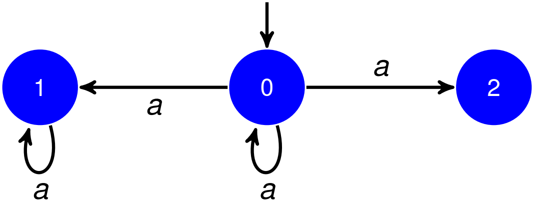
Example:

$$X \overset{\text{max}}{=} X$$

Any set of states S satisfies the set-equation $X = X$. The greatest such set is S.

Eventually 'a' will be disabled:

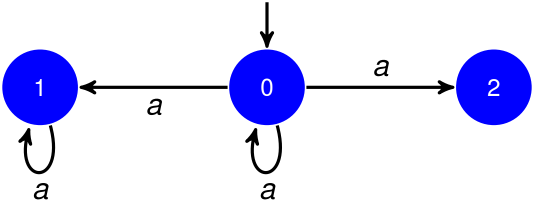
$$X \stackrel{?}{=} [a]\mathit{false} \vee \langle Act \rangle X$$



The property is valid for the labeled transition system

Eventually '*a*' will be disabled:

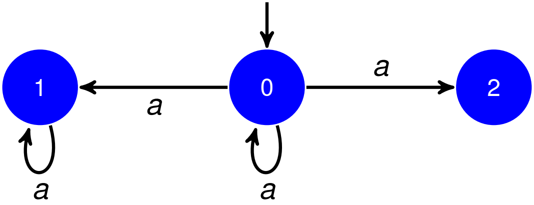
$$X \stackrel{?}{=} [a]\mathit{false} \vee \langle Act \rangle X$$



The property is valid for the labeled transition system
Solutions of this equation are the sets: $\{0,2\}$ and $\{0,1,2\}$

Eventually 'a' will be disabled:

$$X \stackrel{?}{=} [a]\mathit{false} \vee \langle \mathit{Act} \rangle X$$

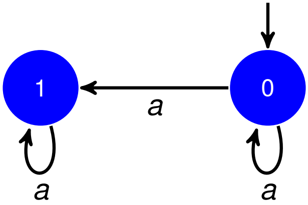


The property is valid for the labeled transition system
Solutions of this equation are the sets: $\{0, 2\}$ and $\{0, 1, 2\}$
We intended to describe the least solution!

$$X \stackrel{\min}{=} [a]\mathit{false} \vee \langle \mathit{Act} \rangle X$$

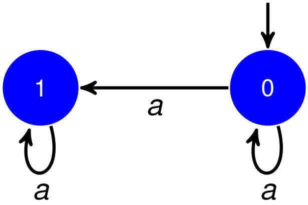
Example: A state can be reached where *a* cannot be executed:

$$X \stackrel{\min}{=} [a]\mathit{false} \vee \langle Act \rangle X$$



Example: A state can be reached where *a* cannot be executed:

$$X \stackrel{\min}{=} [a]\mathit{false} \vee \langle \mathit{Act} \rangle X$$

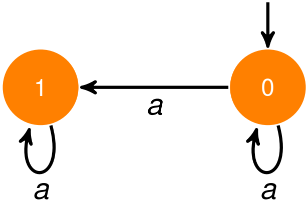


The unique least solution for this equation is the set of states $\emptyset$

Hence the property is not valid for the labeled transition system

Example: In every reachable state an *a*-transition is possible

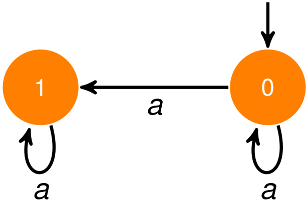
$$X \stackrel{?}{=} \langle a \rangle \mathit{true} \wedge [\mathit{Act}]X$$



Solutions: $\emptyset$, $\{1\}$, and $\{0, 1\}$

Example: In every reachable state an *a*-transition is possible

$$X \stackrel{?}{=} \langle a \rangle true \wedge [Act]X$$



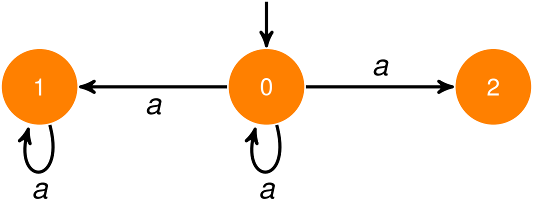
Solutions: $\emptyset$, $\{1\}$, and $\{0, 1\}$

We intended to describe the greatest solution!

$$X \stackrel{\max}{=} \langle a \rangle true \wedge [Act]X$$

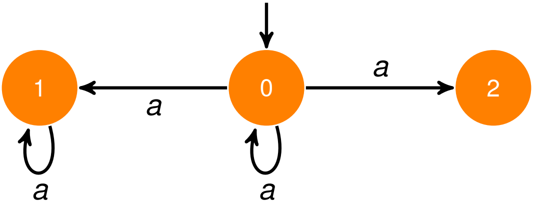
Example: In every reachable state an *a*-transition is possible

$$X \stackrel{\max}{=} \langle a \rangle true \wedge [Act]X$$



Example: In every reachable state an *a*-transition is possible

$$X \stackrel{\max}{=} \langle a \rangle true \wedge [Act]X$$

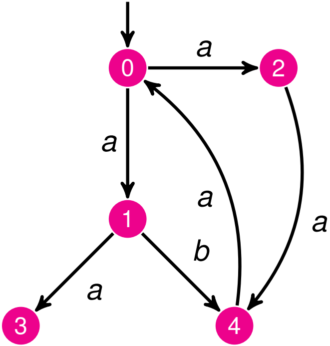


The greatest solution for this equation is the set of states $\{1\}$

Thus property is not valid for the labeled transition system

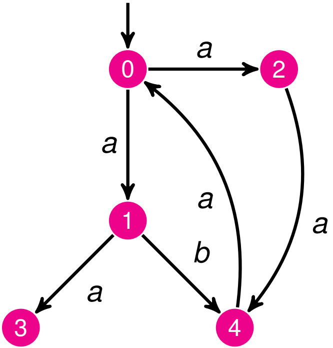
$$X \stackrel{\min}{=} \langle b \rangle \mathit{true} \vee \langle Act \rangle X$$

There is a path to a state where a *b* is possible



$$X \stackrel{\min}{=} \langle b \rangle true \vee \langle Act \rangle X$$

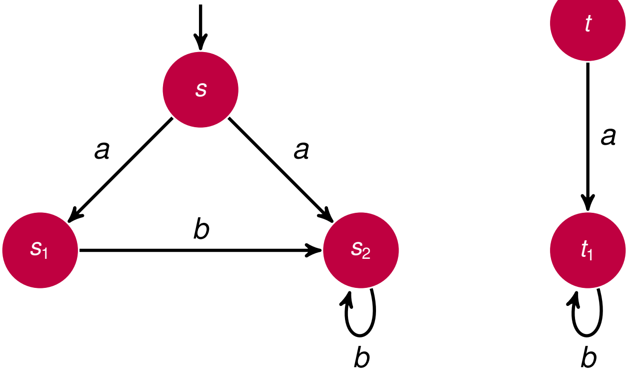
There is a path to a state where a b is possible



The least solution is the set of states $\{0, 1, 2, 4\}$; thus, property is valid for the labeled transition system

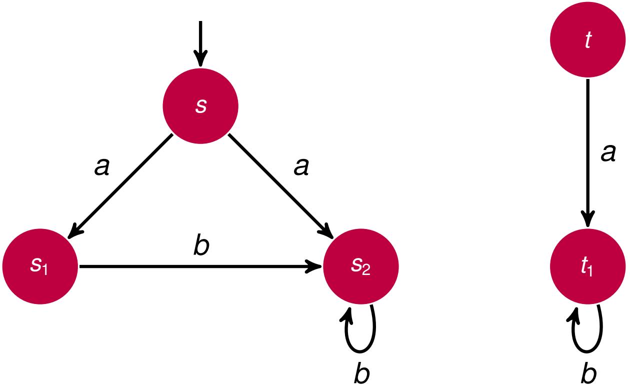
$$X \stackrel{\max}{=} \langle b \rangle true \wedge [b]X$$

initially and after each b, one can take a b transition



$$X \stackrel{\max}{=} \langle b \rangle true \wedge [b]X$$

initially and after each b, one can take a b transition



The greatest solution is the set of states $\{s_1, s_2, t_1\}$.

Formulas for the properties that cannot be expressed in HML

- the scientist never drinks beer

$$X \stackrel{\max}{=} [\text{beer}]\mathit{false} \wedge [\mathit{Act}]X$$

Formulas for the properties that cannot be expressed in HML

- the scientist never drinks beer

$$X \stackrel{\max}{=} [\mathrm{beer}]\mathit{false} \land [\mathit{Act}]X$$

- the scientist always produces a publication after drinking coffee

$$X \stackrel{\max}{=} [\mathrm{coffee}](\langle \mathrm{pub} \rangle \mathit{true} \land [\mathit{Act} \setminus \{\mathrm{pub}\}]\mathit{false}) \land [\mathit{Act}]X$$

Formulas for the properties that cannot be expressed in HML

- the scientist never drinks beer

$$X \stackrel{\max}{=} [\text{beer}]\mathit{false} \wedge [\mathit{Act}]X$$

- the scientist always produces a publication after drinking coffee

$$X \stackrel{\max}{=} [\text{coffee}](\langle\text{pub}\rangle\mathit{true} \wedge [\mathit{Act} \setminus \{\text{pub}\}]\mathit{false}) \wedge [\mathit{Act}]X$$

- $\mathit{Inv}(F)$

$$X \stackrel{\max}{=} F \wedge [\mathit{Act}]X$$

Formulas for the properties that cannot be expressed in HML

- the scientist never drinks beer

$$X \stackrel{\max}{=} [\text{beer}]\mathit{false} \wedge [\mathit{Act}]X$$

- the scientist always produces a publication after drinking coffee

$$X \stackrel{\max}{=} [\text{coffee}](\langle\text{pub}\rangle\mathit{true} \wedge [\mathit{Act} \setminus \{\text{pub}\}]\mathit{false}) \wedge [\mathit{Act}]X$$

- $\mathit{Inv}(F)$

$$X \stackrel{\max}{=} F \wedge [\mathit{Act}]X$$

- $\mathit{Pos}(F)$

$$X \stackrel{\min}{=} F \vee \langle\mathit{Act}\rangle X$$

Outline

Semantics of Recursion (one variable)

- With each formula associate a set of states for which it is satisfied

$$[\![F]\!] \subseteq S$$

Semantics of Recursion (one variable)

- With each formula associate a set of states for which it is satisfied

$$[\![F]\!] \subseteq S$$

- How to deal with recursion variable X?

Semantics of Recursion (one variable)

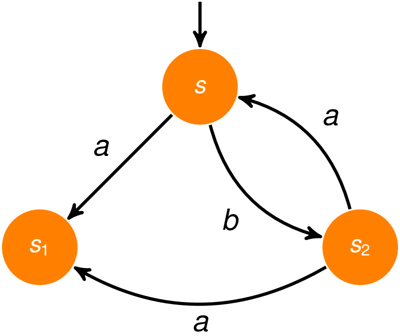
- With each formula associate a set of states for which it is satisfied
$$[\![F]\!] \subseteq S$$
- How to deal with recursion variable X?
- Make an assumption on states satisfied by X. For every formula F we define a function $O_F : 2^S \to 2^S$ s.t.
  - if S is the set of processes that satisfy X
  - then $O_F(S)$ is the set of processes that satisfy F.

Definition of $O_F : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$

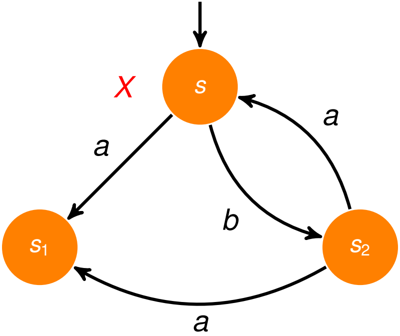
For $S \subseteq \mathcal{S}$

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

Example

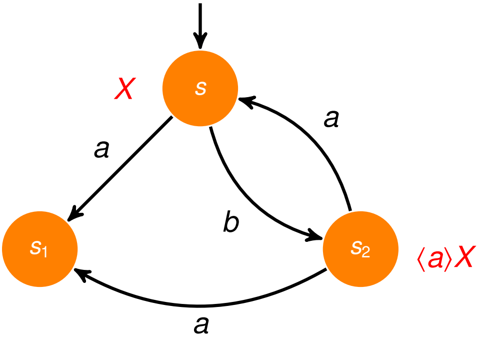


Example



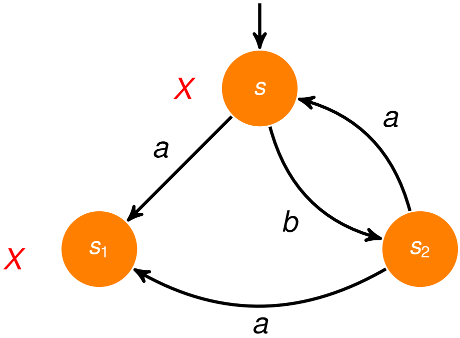
1. $O_{\langle a \rangle X}(\{s\}) = \langle \cdot a \cdot \rangle O_X(\{s\}) = \langle \cdot a \cdot \rangle \{s\} = \{s_2\}$

Example



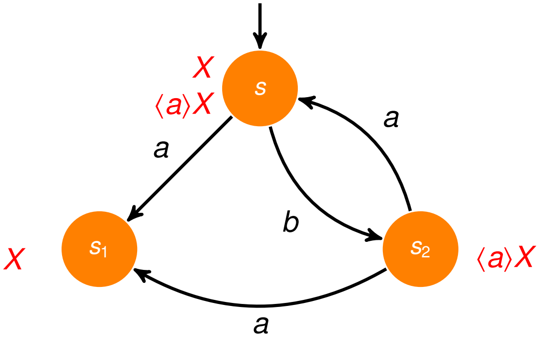
1. $O_{\langle a \rangle X}(\{s\}) = \langle \cdot a \cdot \rangle O_X(\{s\}) = \langle \cdot a \cdot \rangle \{s\} = \{s_2\}$

Example



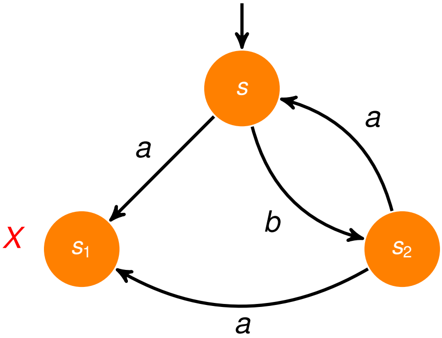
1. $O_{\langle a \rangle X}(\{s\}) = \langle \cdot a \cdot \rangle O_X(\{s\}) = \langle \cdot a \cdot \rangle \{s\} = \{s_2\}$
2. $O_{\langle a \rangle X}(\{s, s_1\}) = \langle \cdot a \cdot \rangle O_X(\{s, s_1\}) = \langle \cdot a \cdot \rangle \{s, s_1\} = \{s, s_2\}$

Example



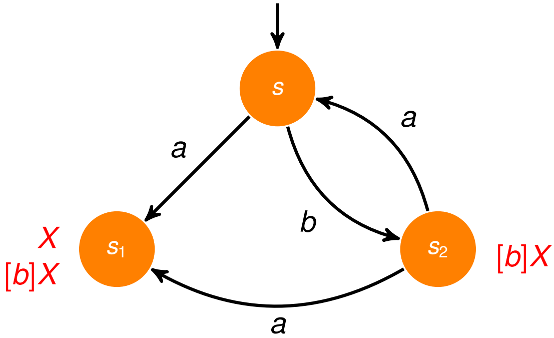
1. $O_{\langle a \rangle X}(\{s\}) = \langle \cdot a \cdot \rangle O_X(\{s\}) = \langle \cdot a \cdot \rangle \{s\} = \{s_2\}$
2. $O_{\langle a \rangle X}(\{s, s_1\}) = \langle \cdot a \cdot \rangle O_X(\{s, s_1\}) = \langle \cdot a \cdot \rangle \{s, s_1\} = \{s, s_2\}$

Example



1. $O_{\langle a\rangle X}(\{s\}) = \langle\cdot a\cdot\rangle O_X(\{s\}) = \langle\cdot a\cdot\rangle\{s\} = \{s_2\}$
2. $O_{\langle a\rangle X}(\{s, s_1\}) = \langle\cdot a\cdot\rangle O_X(\{s, s_1\}) = \langle\cdot a\cdot\rangle\{s, s_1\} = \{s, s_2\}$
3. $O_{[b]X}(\{s_1\}) = [\cdot b\cdot] O_X(\{s_1\}) = [\cdot b\cdot]\{s_1\} = \{s_1, s_2\}$

Example



1. $O_{\langle a\rangle X}(\{s\}) = \langle\cdot a\cdot\rangle O_X(\{s\}) = \langle\cdot a\cdot\rangle\{s\} = \{s_2\}$
2. $O_{\langle a\rangle X}(\{s, s_1\}) = \langle\cdot a\cdot\rangle O_X(\{s, s_1\}) = \langle\cdot a\cdot\rangle\{s, s_1\} = \{s, s_2\}$
3. $O_{[b]X}(\{s_1\}) = [\cdot b\cdot] O_X(\{s_1\}) = [\cdot b\cdot]\{s_1\} = \{s_1, s_2\}$

Observation

Semantics of formula F

Observation

Semantics of formula F

1. $[\![true]\!] = S$
2. $[\![false]\!] = \emptyset$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$
4. $[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$
5. $[\![\langle a \rangle F]\!] = \langle \cdot a \cdot \rangle [\![F]\!]$ where $\langle \cdot a \cdot \rangle : 2^S \to 2^S$ is defined by

$$\langle \cdot a \cdot \rangle S = \{p \in S \mid \exists p'.\, p \xrightarrow{a} p' \text{ and } p' \in S\}$$

6. $[\![[a]F]\!] = [\cdot a \cdot][\![F]\!]$ where $[\cdot a \cdot] : 2^S \to 2^S$ is defined by

$$[\cdot a \cdot] S = \{p \in S \mid \forall p'.\, p \xrightarrow{a} p' \Rightarrow p' \in S\}$$

Observation

Semantics of formula F

1. $[\![true]\!] = S$
2. $[\![false]\!] = \varnothing$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$
4. $[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$
5. $[\![\langle a \rangle F]\!] = \langle \cdot a \cdot \rangle [\![F]\!]$ where $\langle \cdot a \cdot \rangle : 2^S \to 2^S$ is defined by

$$\langle \cdot a \cdot \rangle S = \{p \in S \mid \exists p'.\, p \xrightarrow{a} p' \text{ and } p' \in S\}$$

6. $[\![[a]F]\!] = [\cdot a \cdot][\![F]\!]$ where $[\cdot a \cdot] : 2^S \to 2^S$ is defined by

$$[\cdot a \cdot] S = \{p \in S \mid \forall p'.\, p \xrightarrow{a} p' \Rightarrow p' \in S\}$$

7. If $X \stackrel{\min}{=} F_X$ then $[\![X]\!] = \bigcap\{S \subseteq S \mid S = O_{F_X}(S)\}$

Observation

Semantics of formula F

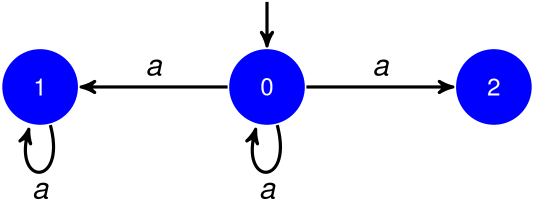
1. $[\![true]\!] = S$
2. $[\![false]\!] = \varnothing$
3. $[\![F \wedge G]\!] = [\![F]\!] \cap [\![G]\!]$
4. $[\![F \vee G]\!] = [\![F]\!] \cup [\![G]\!]$
5. $[\![\langle a \rangle F]\!] = \langle \cdot a \cdot \rangle [\![F]\!]$ where $\langle \cdot a \cdot \rangle : 2^S \to 2^S$ is defined by
$$\langle \cdot a \cdot \rangle S = \{p \in S \mid \exists p'.\, p \xrightarrow{a} p' \text{ and } p' \in S\}$$
6. $[\![[a]F]\!] = [\cdot a \cdot][\![F]\!]$ where $[\cdot a \cdot] : 2^S \to 2^S$ is defined by
$$[\cdot a \cdot] S = \{p \in S \mid \forall p'.\, p \xrightarrow{a} p' \Rightarrow p' \in S\}$$
7. If $X \stackrel{\min}{=} F_X$ then $[\![X]\!] = \bigcap\{S \subseteq S \mid S = O_{F_X}(S)\}$
8. If $X \stackrel{\max}{=} F_X$ then $[\![X]\!] = \bigcup\{S \subseteq S \mid S = O_{F_X}(S)\}$

Let S be a finite set.

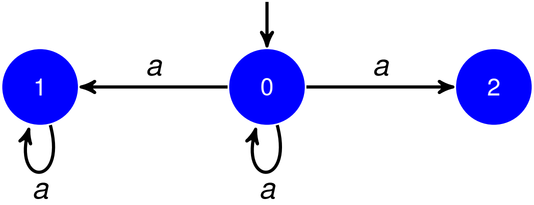
$$\frac{\text{Computing the solution of } X \stackrel{\min}{=} F_X}{\text{There exists a natural number } m > 0 \text{ such that } [\![X]\!] = {O_{F_X}}^m(\emptyset)}$$

$$\frac{\text{Computing the solution of } X \stackrel{\max}{=} F_X}{\text{There exist a natural number } M > 0 \text{ such that } [\![X]\!] = {O_{F_X}}^M(S)}$$

Example: $X \stackrel{\min}{=} [a]\mathit{false} \vee \langle \mathit{Act} \rangle X$

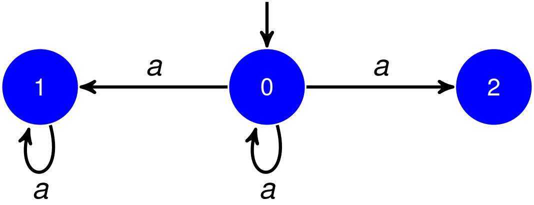


Example: $X \stackrel{\min}{=} [a]\mathit{false} \vee \langle Act \rangle X$



$$
\begin{aligned}
O_{F_X}(S) &= O_{[a]\mathit{false}}(S) \cup O_{\langle Act \rangle X}(S) \\
&= [\cdot a \cdot] O_{\mathit{false}}(S) \cup \langle \cdot Act \cdot \rangle O_X(S) \\
&= [\cdot a \cdot] \varnothing \cup \langle \cdot Act \cdot \rangle S \\
&= \{2\} \cup \langle \cdot Act \cdot \rangle S
\end{aligned}
$$

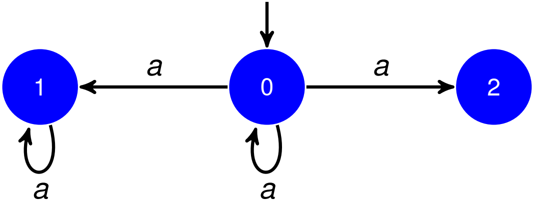
Example: $X \overset{min}{=} [a]\mathit{false} \vee \langle Act \rangle X$



$$\begin{aligned} O_{F_X}(S) &= O_{[a]\mathit{false}}(S) \cup O_{\langle Act \rangle X}(S) \\ &= [\cdot a \cdot] O_{\mathit{false}}(S) \cup \langle \cdot Act \cdot \rangle O_X(S) \\ &= [\cdot a \cdot] \emptyset \cup \langle \cdot Act \cdot \rangle S \\ &= \{2\} \cup \langle \cdot Act \cdot \rangle S \end{aligned}$$

1. $O_{F_X}(\emptyset) = \{2\} \cup \langle \cdot Act \cdot \rangle \emptyset = \{2\} \cup \emptyset = \{2\}$

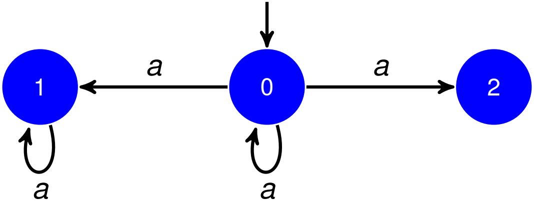
Example: $X \stackrel{\min}{=} [a]\mathit{false} \vee \langle Act \rangle X$



$$
\begin{aligned}
O_{F_X}(S) &= O_{[a]\mathit{false}}(S) \cup O_{\langle Act\rangle X}(S)\\
&= [\cdot a\cdot]O_{\mathit{false}}(S) \cup \langle\cdot Act\cdot\rangle O_X(S)\\
&= [\cdot a\cdot]\varnothing \cup \langle\cdot Act\cdot\rangle S\\
&= \{2\} \cup \langle\cdot Act\cdot\rangle S
\end{aligned}
$$

1. $O_{F_X}(\varnothing) = \{2\} \cup \langle\cdot Act\cdot\rangle\varnothing = \{2\} \cup \varnothing = \{2\}$
2. $O_{F_X}(\{2\}) = \{2\} \cup \langle\cdot Act\cdot\rangle\{2\} = \{2\} \cup \{0\} = \{0, 2\}$

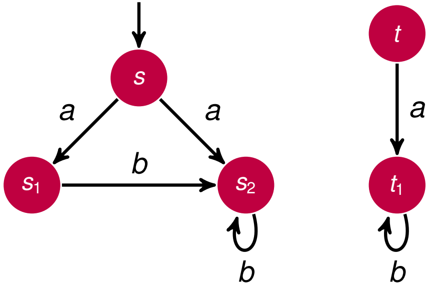
Example: $X \stackrel{min}{=} [a]\mathit{false} \vee \langle Act \rangle X$



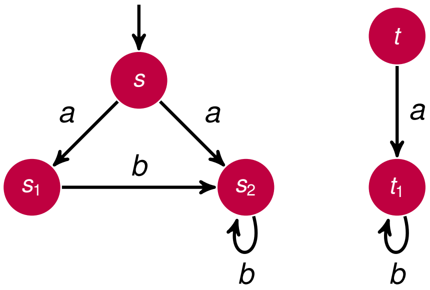
$$\begin{aligned}
O_{F_X}(S) &= O_{[a]\mathit{false}}(S) \cup O_{\langle Act \rangle X}(S) \\
&= [\cdot a \cdot] O_{\mathit{false}}(S) \cup \langle \cdot Act \cdot \rangle O_X(S) \\
&= [\cdot a \cdot] \emptyset \cup \langle \cdot Act \cdot \rangle S \\
&= \{2\} \cup \langle \cdot Act \cdot \rangle S
\end{aligned}$$

1. $O_{F_X}(\emptyset) = \{2\} \cup \langle \cdot Act \cdot \rangle \emptyset = \{2\} \cup \emptyset = \{2\}$
2. $O_{F_X}(\{2\}) = \{2\} \cup \langle \cdot Act \cdot \rangle \{2\} = \{2\} \cup \{0\} = \{0, 2\}$
3. $O_{F_X}(\{0, 2\}) = \{2\} \cup \langle \cdot Act \cdot \rangle \{0, 2\} = \{2\} \cup \{0\} = \{0, 2\}$

Example: $X \stackrel{\max}{=} \langle b \rangle \mathit{true} \wedge [b]X$

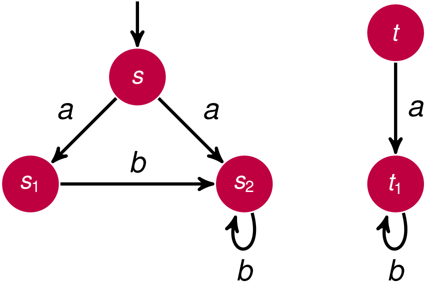


Example: $X \overset{\max}{=} \langle b \rangle true \wedge [b]X$

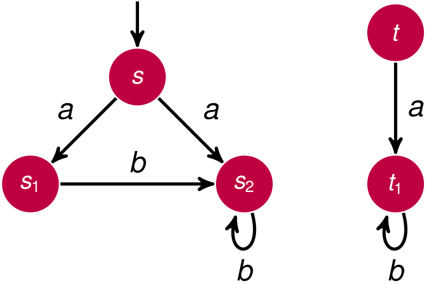


$$
\begin{aligned}
O_{F_X}(S) &= O_{\langle b \rangle true}(S) \cap O_{[b]X}(S) \\
&= \langle \cdot b \cdot \rangle O_{true}(S) \cap [\cdot b \cdot] O_X(S) \\
&= \langle \cdot b \cdot \rangle S \cap [\cdot b \cdot] S \\
&= \{s_1, s_2, t_1\} \cap [\cdot b \cdot] S
\end{aligned}
$$

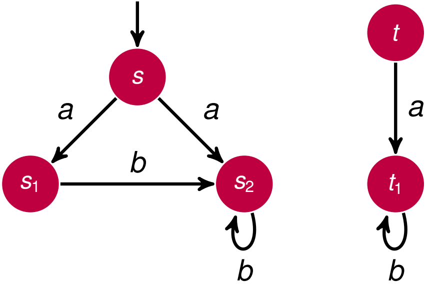
Example: $X \stackrel{\max}{=} \langle b \rangle true \wedge [b]X$



Example: $X \overset{max}{=} \langle b \rangle true \wedge [b]X$

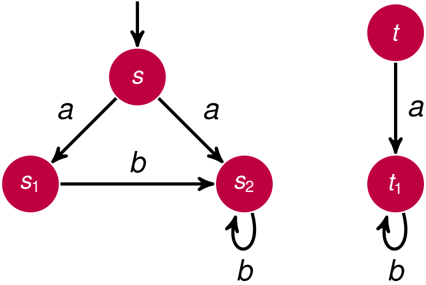


Example: $X \stackrel{\max}{=} \langle b \rangle true \wedge [b]X$



1. $O_{F_X}(S) = \{s_1, s_2, t_1\} \cap [\cdot b \cdot]S = \{s_1, s_2, t_1\} \cap \{s, s_1, s_2, t, t_1\} = \{s_1, s_2, t_1\}$

Example: $X \stackrel{\max}{=} \langle b \rangle true \wedge [b]X$



1. $O_{F_X}(S) = \{s_1, s_2, t_1\} \cap [\cdot b \cdot]S = \{s_1, s_2, t_1\} \cap \{s, s_1, s_2, t, t_1\} = \{s_1, s_2, t_1\}$
2. $O_{F_X}(\{s_1, s_2, t_1\}) = \{s_1, s_2, t_1\} \cap [\cdot b \cdot]\{s_1, s_2, t_1\} = \{s_1, s_2, t_1\} \cap \{s, s_1, s_2, t, t_1\} = \{s_1, s_2, t_1\}$

Some temporal properties

- *Safe*(F): for some execution F holds everywhere

$$X \stackrel{\max}{=} F \wedge ([Act]\mathit{false} \vee \langle Act \rangle X)$$

Some temporal properties

- *Safe*(F): for some execution F holds everywhere

$$X \stackrel{\max}{=} F \wedge ([Act]\mathit{false} \vee \langle Act \rangle X)$$

- *Even*(F): eventually F will hold (in every execution)

$$X \stackrel{\min}{=} F \vee (\langle Act \rangle \mathit{true} \wedge [Act]X)$$

Some temporal properties

- $\mathit{Safe}(F)$: for some execution F holds everywhere

$$X \stackrel{\max}{=} F \wedge ([\mathit{Act}]\mathit{false} \vee \langle \mathit{Act} \rangle X)$$

- $\mathit{Even}(F)$: eventually F will hold (in every execution)

$$X \stackrel{\min}{=} F \vee (\langle \mathit{Act} \rangle \mathit{true} \wedge [\mathit{Act}]X)$$

- $F\,\mathcal{U}^w\,G$: F holds in all states until a state is reached where G holds

$$X \stackrel{\max}{=} G \vee (F \wedge [\mathit{Act}]X)$$

Some temporal properties

- *Safe*(F): for some execution F holds everywhere

$$X \stackrel{\max}{=} F \wedge ([Act]\mathit{false} \vee \langle Act \rangle X)$$

- *Even*(F): eventually F will hold (in every execution)

$$X \stackrel{\min}{=} F \vee (\langle Act \rangle \mathit{true} \wedge [Act]X)$$

- $F \, \mathcal{U}^w \, G$: F holds in all states until a state is reached where G holds

$$X \stackrel{\max}{=} G \vee (F \wedge [Act]X)$$

- $F \, \mathcal{U}^s \, G$: sooner or later G holds and until then F holds in all states traversed

$$X \stackrel{\min}{=} G \vee (F \wedge \langle Act \rangle \mathit{true} \wedge [Act]X)$$

Some temporal properties

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

$\mathit{Inv}(F)$ and $F\, \mathcal{U}^w\, \mathit{false}$ are logically equivalent

$\mathit{Even}(F)$ and $\mathit{true}\, \mathcal{U}^s\, F$ are logically equivalent