LCD (01/04)

Hennessy - Milner's Logic

$$\varphi, \psi ::= \top \mid \bot \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \langle a \rangle \varphi \mid [a] \varphi$$

closely related to bisimilarity (program equivalence)

Hennessy - Milner's Theorem

If P, Q are image-finite processes

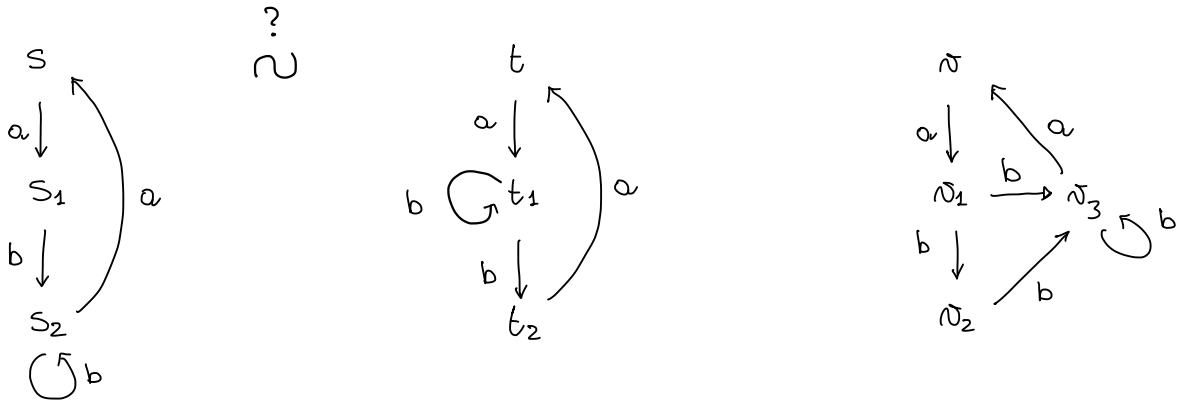
$$P \sim Q \quad \text{iff} \quad \forall \varphi \ (P \models \varphi \leftrightarrow Q \models \varphi)$$

i.e.

① if $P \sim Q$ then $\forall \varphi \ (P \models \varphi$ iff $Q \models \varphi)$ [does not require image-finiteness]

② if $P \not\sim Q$ then $\exists \varphi \ P \models \varphi$ and $Q \not\models \varphi$

Example



$s \models [a][b]\langle b \rangle \top \quad \not\models t \qquad s \not\sim t$

$s \models [a][b]\langle a \rangle \top \quad \not\models v \qquad s \not\sim v$

$t \not\models [a][b][b]\langle a \rangle \top \models v \qquad t \not\sim v$

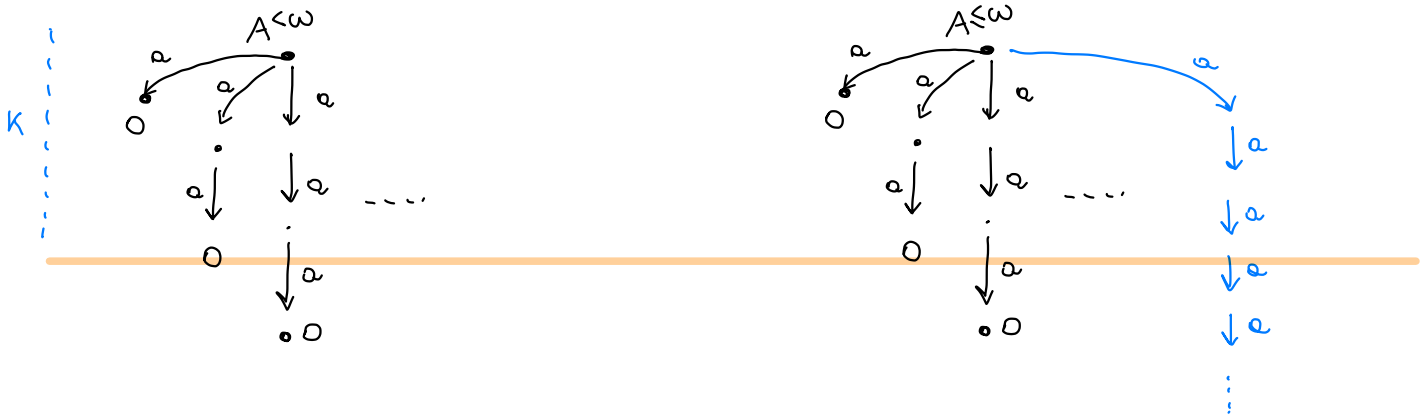
* Counterexample to Hennessy-Milner's theorem for non-image finite processes

we want

$P \not\sim Q \qquad \forall \varphi \quad (P \models \varphi \text{ iff } Q \models \varphi)$

$A^{<\omega} = \sum_{i \in \mathbb{N}} a^i$ $\qquad$ $A^{\leq \omega} = A^{<\omega} + A^{\omega}$

$a^i = \underbrace{a.a. \ldots a.}_{i \text{ times}} 0$ $\qquad$ $A^{\omega} = a.A^{\omega}$



We have $A^{<\omega} \not\sim A^{\leq\omega}$

$\forall \varphi \in HML \qquad A^{<\omega} \models \varphi \quad \text{iff} \quad A^{\leq\omega} \models \varphi$

(EXERCISE FOR THE EXAM)

Hennessy Milner Logic with Recursion

$r = 0 \qquad p \not\sim q \qquad p \models [a]\langle a\rangle T \qquad \not\models q$

$r = a.0 \qquad\qquad [a][a]\langle a\rangle T$

$\vdots$

$r = \underbrace{a \ldots a.}_{i} 0 \qquad\qquad \underbrace{[a] \ldots [a]}_{i} \langle a\rangle T$

$\vdots$

distinguishing property

$$Inv(\langle a\rangle T) = \bigwedge_{i \geq 1} \underbrace{[a] \ldots [a]}_{i} \langle a\rangle T$$

$$Pos([a]F) = \bigvee_{i \geq 1} \underbrace{\langle a \rangle \ldots \langle a \rangle}_{i} [a]F$$

↕ infinite formulae

We want to use recursion

for expanding $Inv(\langle a \rangle T)$

$$X = \langle a \rangle T \wedge [a] X$$

↑ solution? unique? canonical solution?

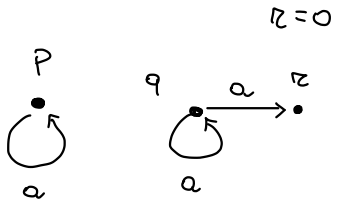
$$[\![X]\!] = [\![\langle a \rangle T \wedge [a] X]\!]$$

$$= \langle a \rangle [\![T]\!] \cap [a] [\![X]\!]$$

$$= \langle a \rangle Proc \cap [a] [\![X]\!]$$

$S \subseteq Proc$

$$S = \langle a \rangle Proc \cap [a] S$$

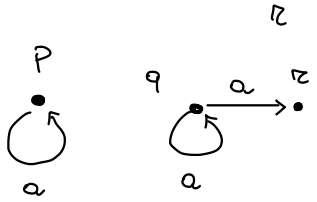


which solution ?

$S = \emptyset$ $\qquad \emptyset = \underbrace{\langle a \rangle Proc}_{\text{processes which can input } a} \cap \underbrace{[a]\emptyset}_{\text{processes which cannot do } a}$

$S = \{p\}$ $\qquad \{p\} = \underbrace{\langle a \rangle Proc}_{\{p, q\}} \cap \underbrace{[a]\{p\}}_{\{p, r\}}$

↑ greatest solution



* $Pos([a]F)$

$$Y = [a]F \vee \langle a \rangle Y$$

$$S = [a]\emptyset \cup \langle a \rangle S$$

$S = Proc$ is a solution

$$Proc = \underbrace{[a]\emptyset}_{\text{can't do } a} \cup \underbrace{\langle a \rangle Proc}_{\text{can do } a}$$

I want $S = \{q, r\}$ least solution.

Syntax

$Inv(\langle a \rangle T)$ $\quad X \overset{max}{=} \langle a \rangle T \wedge [a] X$

$\nu X. \langle a \rangle T \wedge [a] X$

$Pos([a]F)$ $\quad Y \overset{min}{=} [a]F \vee \langle a \rangle Y$

$\mu Y. [a]F \vee \langle a \rangle Y$

↑ μ-calculus

Other properties

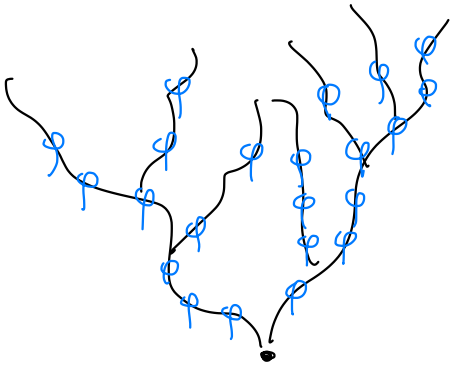
Given φ

* $Inv(\varphi) = \nu X. (\varphi \wedge [Act] X)$

↑

$[a_1 \dots a_m]\varphi$

$= [a_1]\varphi \wedge \dots \wedge [a_m]\varphi$

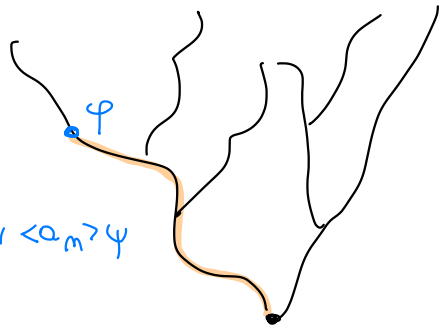


* $Pos(\varphi) = \mu X. (\varphi \vee \langle Act \rangle X)$

↑

$\langle a_1 \dots a_m \rangle \varphi$

$= \langle a_1 \rangle \varphi \vee \dots \vee \langle a_m \rangle \varphi$



NoDeadlock $\equiv$ $Inv(\langle Act \rangle T)$

* Safe(φ) ≡ there is a complete trace where φ always holds

$P = P_0 \to P_1 \to P_2 \to - - - -$ infinite

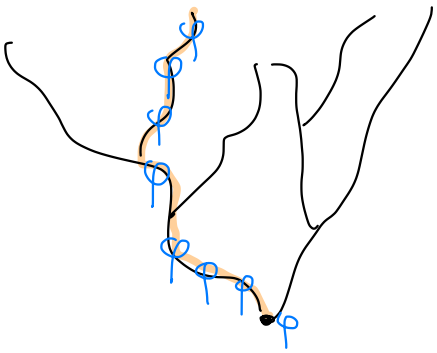
$\searrow \dots \to P_m \not\to$

$$Safe(\varphi) \equiv \nu X \, (\varphi \wedge (\langle Act \rangle X \vee [Act] F))$$

φ holds now

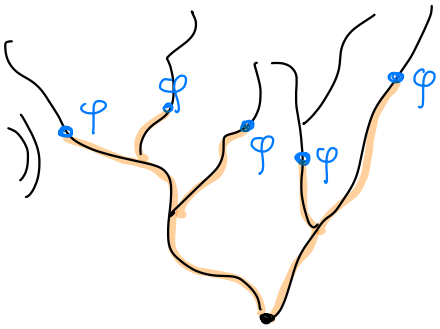
I can move to a state where X holds

computation terminated



* Even (φ) = in every complete computation (trace) I reach a state where φ holds

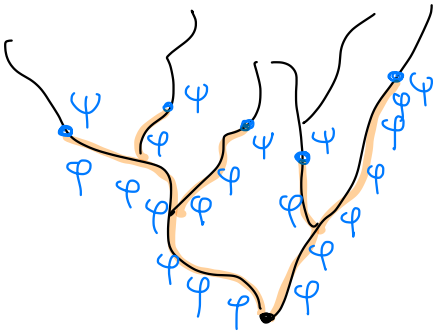
$$Even(\varphi) \equiv \mu X. \, (\varphi \vee ([Act] X \wedge \langle Act \rangle T))$$



* Until (strong)

$\varphi \, \mathcal{U} \, \psi$

$\langle backup \rangle T \, \mathcal{U} \, \langle official \rangle T$



$$\varphi \, \mathcal{U} \, \psi = \mu X. \, (\psi \vee (\varphi \wedge [Act] X \wedge \langle Act \rangle T))$$

Formal view: μ-calculus

$$\varphi, \psi ::= T \mid F \mid \psi \wedge \varphi \mid \varphi \vee \psi \mid \langle a \rangle \varphi \mid [a] \varphi \mid X \mid \mu X. \varphi \mid \nu X. \varphi$$

$\eta : Var \to 2^{Proc}$

$X \mapsto \eta(X) \subseteq Proc$

↑ processes which satisfy X

$[\![\varphi]\!]_\eta \subseteq Proc$

⋮ same as for HML

$[\![\varphi \wedge \psi]\!]_\eta = [\![\varphi]\!]_\eta \cap [\![\psi]\!]_\eta$

⋮

$[\![X]\!]_\eta = \eta(X)$

$[\![\mu X.\ \varphi]\!]_\eta = lfp(f_\varphi^\eta)$ $\qquad X \overset{min}{=} \ldots\ldots X \ldots$ (φ)

$[\![\nu X.\ \varphi]\!]_\eta = gfp(f_\varphi^\eta)$ $\qquad$ if $X \mapsto S \subseteq Proc$

$[\![\varphi]\!]_{\eta[X \mapsto S]}$

$f_\varphi^\eta : 2^{Proc} \to 2^{Proc}$

$S \longmapsto [\![\varphi]\!]_{\eta[X \mapsto S]}$

NOTE: $(2^{Proc}, \subseteq)$ powerset lattice ; f_φ^η is monotone } Knaster-Tarski

* with finite state processes

$\nu X.\varphi \qquad f_\varphi$

$Proc \supseteq f_\varphi(Proc) \supseteq f_\varphi^2(Proc) \supseteq \ldots \supseteq f^k(Proc) = f^{k+1}(Proc)$

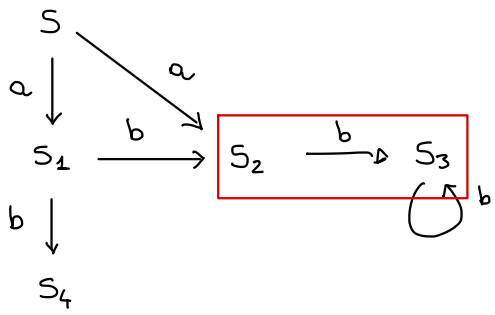
$= [\![\nu X.\varphi]\!]$

$\mu X.\varphi$

$\emptyset \subseteq f_\varphi(\emptyset) \subseteq f_\varphi^2(\emptyset) \subseteq \ldots \subseteq f^k(\emptyset) = f^{k+1}(\emptyset)$

$= [\![\mu X.\varphi]\!]$

Example :

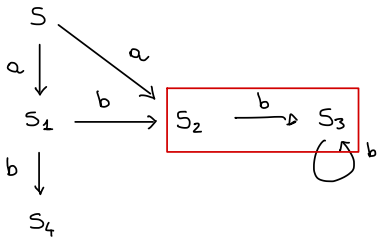


$\varphi = Inv(\langle b \rangle T) = \nu X.\ \langle b \rangle T \wedge [Act] X$

$f_\varphi(S) = [\![\langle b \rangle T \wedge [Act] X]\!]_{\eta[X \to S]} = \underbrace{\langle b \rangle Proc}_{\{s_1, s_2, s_3\}} \cap [Act] S$

$f_\varphi^0(Proc) = Proc$

$f_\varphi(Proc) = \{s_1, s_2, s_3\} \cap \overbrace{[Act] Proc}^{Proc} = \{s_1, s_2, s_3\}$



$f_\varphi^2(Proc) = \{s_1, s_2, s_3\} \cap \underbrace{[Act] \{s_1, s_2, s_3\}}_{\{s, s_2, s_3, s_4\}} = \{s_2, s_3\}$

$f_\varphi^3(Proc) = \{s_1, s_2, s_3\} \cap \underbrace{[Act] \{s_2, s_3\}}_{\{s_2, s_3, s_4\}} = \{s_2, s_3\} = gfp(f_\varphi)$

EXERCISE : $Inv(\varphi) = \nu X.\ (\varphi \wedge [Act] X)$

(EXAM) $S = \{ P \mid \forall P \to^* P' \quad P' \models \varphi \}$

$[\![Inv(\varphi)]\!] = S$?