LCD (16/04/2024)

Hennessy - Milner's Logic

$$\varphi, \psi ::= \quad T \mid F \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \langle a \rangle \varphi \mid [a] \varphi$$

closely related to <u>bisimilarity</u> (program equivalence)

Hennessy - Milner's Theorem

If P, Q are image-finite processes

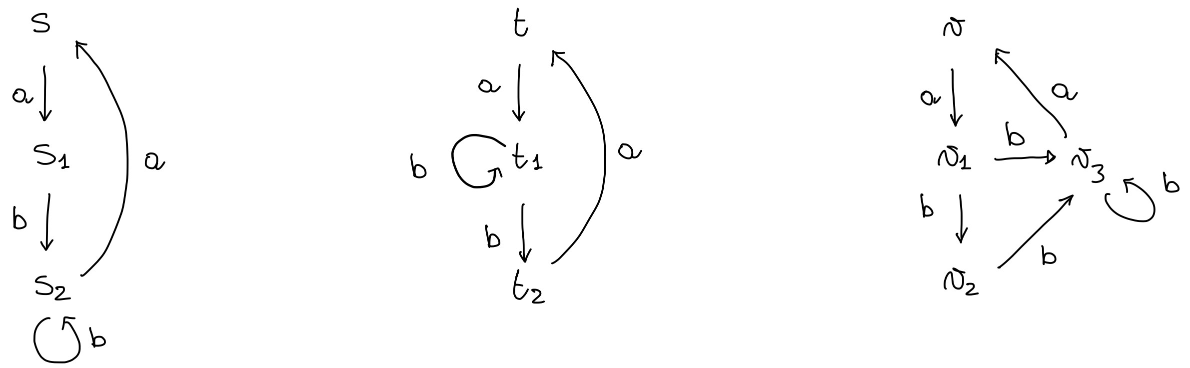
$$P \sim Q \qquad \text{iff} \qquad \forall \varphi \; (P \models \varphi \iff Q \models \varphi)$$

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 $\exists \varphi \quad P \models \varphi$ and $Q \not\models \varphi$

Example



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

$s \not\sim v \qquad s \models \quad \text{"} \quad \not\models v$

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

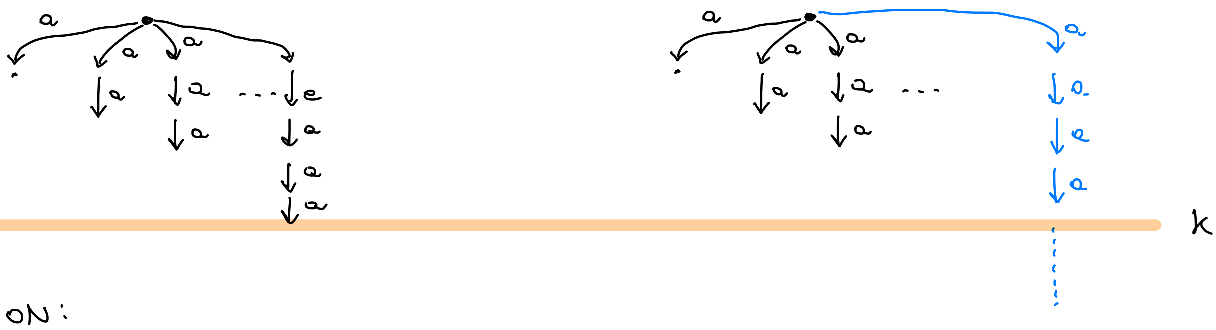
* Counterexample showing the need of image-finiteness

$P, Q \qquad P \not\sim Q \quad \text{and} \quad \forall \varphi \in HML \qquad P \models \varphi \iff Q \models \varphi$

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

$$A^{\omega} = a.\,A^{\omega}$$

$$a^m = \underbrace{a.a.\,\ldots\,a}_{m \text{ times}}.0$$

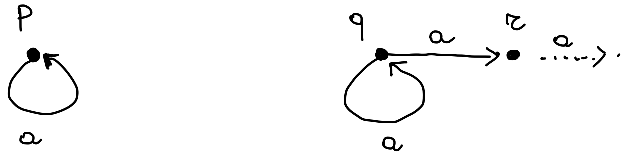


OBSERVATION:

(1) $A^{<\omega} \not\sim A^{\leq\omega}$ EXERCISE

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

* Hennessy Milner's logic with recursion



$r = 0 \qquad p \not\sim q \qquad \varphi \text{ s.t. } p \models \varphi,\ q \not\models \varphi \qquad \varphi = [a]\langle a\rangle T$

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

$r = a.a.0$

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

distinguishing property $\quad Inv(\langle a\rangle T) = \bigwedge_{m \in \mathbb{N}} \underbrace{[a]\ldots[a]}_{m}\langle a\rangle T$

$$Pos([a]F) = \bigvee_{m \in \mathbb{N}} \langle a\rangle\ldots\langle a\rangle[a]F$$

We use recursion for defining $Inv(\langle a\rangle T)$

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

→ is there a solution

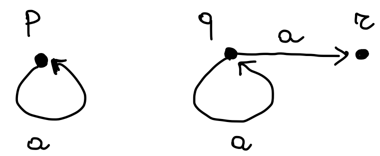
→ unique / canonical?

$$\underbrace{[\![X]\!]}_{S} = [\![\langle a\rangle T \wedge [a]\, X]\!]$$

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

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

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



which solution?

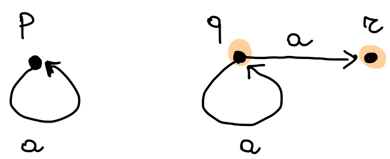
$S = \emptyset$ $\qquad \emptyset = \underbrace{\langle a\rangle\, Proc}_{\text{processes which can do "a"}} \cap \underbrace{[a]\, \emptyset}_{\text{processes not able to do "a"}}$

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

* $Pos([a]F)$

$$\underbrace{Y}_{S} = [a]F \vee \langle a\rangle Y$$

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



$S = Proc$ is a solution $\qquad Proc = \underbrace{[a]\emptyset}_{\text{not able to do "a"}} \cup \underbrace{\langle a\rangle\, Proc}_{\text{able to do "a"}}$

I want $S = \{q, r\}$ $\qquad$ smallest solution

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

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

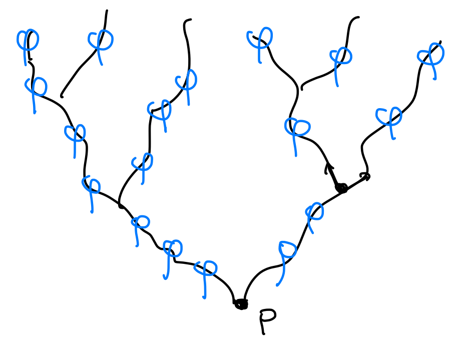
$\mathrm{Pos}([a]F)$ $\qquad$ $Y \overset{min}{=} [a]F \vee \langle a\rangle Y$

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

* Other properties

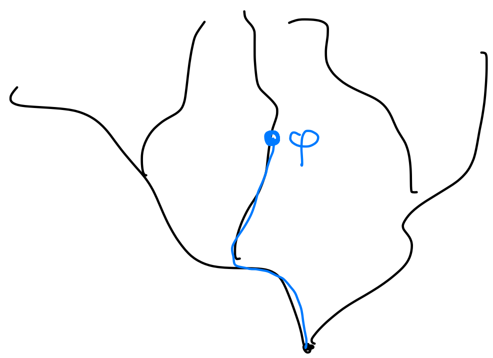
* given φ

$$\mathrm{Inv}(\varphi) = \nu X.(\varphi \wedge \overbrace{[Act]X}^{[a_1]X \wedge \ldots \wedge [a_n]X})$$



No deadlock: $\mathrm{Inv}(\underbrace{\langle Act\rangle}_{\langle a_1\rangle T \vee .. \vee \langle a_n\rangle T} T)$

$$\mathrm{Pos}(\varphi) = \mu Y(\varphi \vee \langle Act\rangle Y)$$



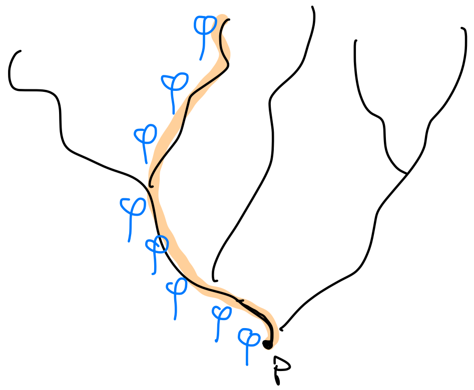
* $\mathrm{Safe}(\varphi)$ = there is a complete computation (trace)

$p = p_0 \to p_1 \to p_2 \to \ldots$ infinite

$\dashrightarrow p_n \not\to$ finite

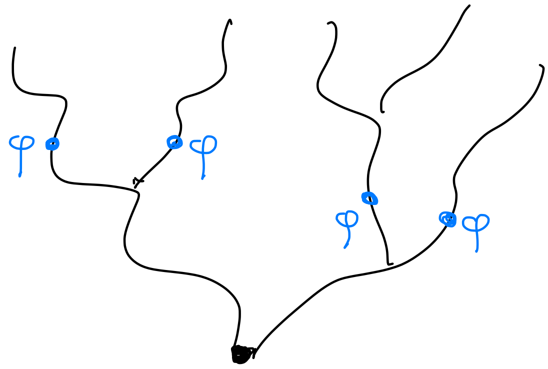
where φ always hold $\forall i$ $p_i \models \varphi$

$$\mathrm{Safe}(\varphi) = \nu X.(\underbrace{\varphi}_{\varphi \text{ holds now}} \wedge (\underbrace{\langle Act\rangle X}_{\text{I can make and then satisfy } X} \vee \underbrace{[Act]F}_{\text{no step possible}}))$$



* Even (φ) = in every complete computation there is a state where φ holds

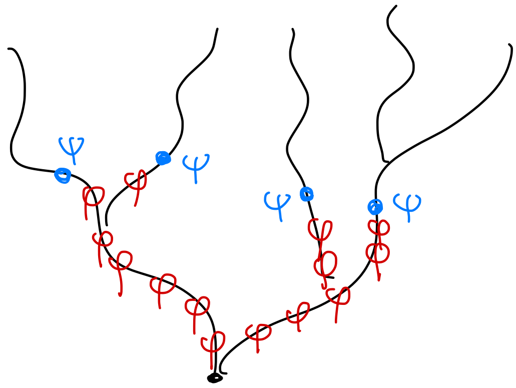
$$\text{Even}(\varphi) = \mu X. \left(\varphi \vee \left([Act]\, X \wedge \langle Act \rangle T \right) \right)$$



- Until $\qquad \varphi \,\mathcal{U}\, \psi$

$\quad //$

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



More precisely ....

$\boxed{\mu\text{- calculus}}$

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

$[\![\varphi]\!]_\eta \qquad \eta : Var \to 2^{Proc}$

$\eta(X) \subseteq Proc \qquad$ processes for which X is true

$[\![T]\!]_\eta = Proc$

$[\![F]\!]_\eta = \emptyset$

$\vdots$

$[\![\langle a \rangle \varphi]\!]_\eta = \{ P \mid \exists P \xrightarrow{a} P' \wedge P' \in [\![\varphi]\!]_\eta \}$

$\vdots$

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

$[\![\nu x.\ \varphi]\!]_\eta$

$\varphi = \ldots\ldots x \ldots x \ldots$

meaning for x

$S \longrightarrow [\![\varphi]\!]_{\eta[x \to S]}$

$$\eta[x \to S](y) = \begin{cases} \eta(y) & y \neq S \\ S & y = x \end{cases}$$

$$f_\varphi : 2^{Proc} \longrightarrow 2^{Proc}$$
$$S \longmapsto [\![\varphi]\!]_{\eta[x \mapsto S]}$$

then $[\![\nu x. \varphi]\!]_\eta = Fix(f_\varphi)$ largest fixpoint of f_φ

$(2^{Proc}, \subseteq)$ complete lattice

f_φ monotone

(depends on absence of negation)

$\boxed{\text{EXERCISE}}$ (exam)

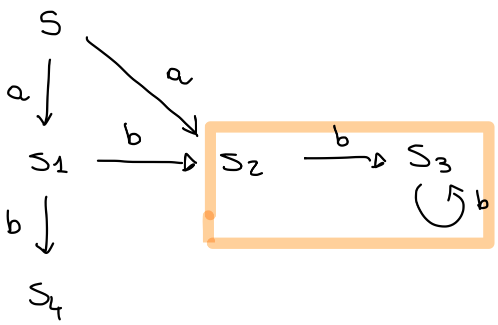
$[\![\mu x.\ \varphi]\!]_\eta = fix(f_\varphi)$

* with finite state processes (Proc is finite)

$\nu x. \varphi \qquad Proc \supseteq f_\varphi(Proc) \supseteq f_\varphi f_\varphi(Proc) \ldots \supseteq Fix(f_\varphi) = [\![\nu x. \varphi]\!]$

$\mu x. \varphi \qquad \emptyset \subseteq f_\varphi(\emptyset) \subseteq \ldots\ldots \subseteq f_\varphi^m(\varphi) = fix(\varphi)$

Exercise: Explicitly compute the semantics

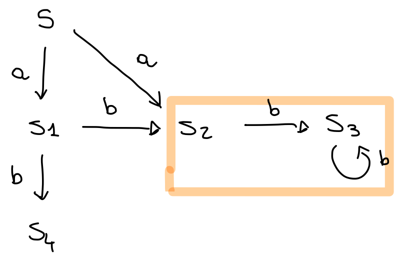


$$\varphi = \mathrm{Inv}(\langle b\rangle T)$$

$$= \nu X.\ \underbrace{\langle b\rangle T \wedge [Act]X}_{\psi}$$

$$f_\psi(S) = [\![\psi]\!]_{\eta[X \to S]} = \langle b\rangle [\![T]\!] \cap [Act]S$$

$$= \underbrace{\langle b\rangle Proc}_{\{s_1, s_2, s_3\}} \cap [Act]S$$



$$f^0_\psi(Proc) = Proc$$

$$f^1_\psi(Proc) = \underbrace{\langle b\rangle Proc} \cap \underbrace{[Act]Proc} = \{s_1, s_2, s_3\}$$

$$f^2_\psi(Proc) = \underbrace{\langle b\rangle Proc}_{\{s_1, s_2, s_3\}} \cap \underbrace{[Act]\{s_1, s_2, s_3\}}_{\{s, s_2, s_3, s_4\}} = \{s_2, s_3\}$$

$$f^3_\psi(Proc) = \{s_2, s_3\}$$

EXERCISE : We defined

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

EXAM

I could define the set of processes where φ invariantly holds directly

$$S = \{ P \mid \forall P \xrightarrow{*} P' \quad P' \models \varphi \}$$

Then show that

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

shows that the formula captures the intended behaviour.

The same can be done for Pos , Even , Safe , Until ....