HML with Recursive Formulas

- Two basic temporal properties

  Inv (F) : always (in time) we have F

  Pos (F) : it is possible to get an state where F
- Capturing Inv and Pos by "infinite" HML formulas :

  Inv by $\bigwedge_{n \in \mathbb{N}} \underbrace{[Act] \dots [Act]}_{n} F$

  Pos by $\bigwedge_{n \in \mathbb{N}} \underbrace{\langle Act \rangle \dots \langle Act \rangle}_{n} F$
- Representing "regular" infinite formulas by recursive formulas

  Inv (F) by $X_{Inv} \overset{def}{=} F \wedge [Act] F$

  Pos (F) by $X_{Pos} \overset{def}{=} F \vee \langle Act \rangle X$

Interlude on Fixed Points Theory

- Partially Ordered Sets : $\sqsubseteq$

  Algebraicaly structured via sup ($\sqcup$) and inf ($\sqcap$)

  Complete lattices : $\sqcup X$, $\sqcap X$ always exist

  Monotonic functions $f : D \to D$ Fixed points $f(d) = d$

  Largest z_{max} , and Least z_{min}

  $f(z_{min}) = z_{min}$; $f(z) = z \Rightarrow z_{min} \sqsubseteq z$

  D finite : $z_{min} = \sqcup f^n(\bot)$ $\bot = \sqcap D$
- Bisimulation is a Fixed Point

  $\sim \; = \bigcup \{ R \mid R \text{ is a bisimulation} \}$

  Lattice of Relations : $\mathcal{P}(Proc \times Proc)$

  Bisimilarity operator : $(s,t) \in \mathcal{F}_\sim(R) \Leftrightarrow \left\{ \begin{array}{l} s \xrightarrow{a} s' \Rightarrow t \xrightarrow{a} t' \\ t \xrightarrow{a} t' \Rightarrow s \xrightarrow{a} s' \end{array} \right\} (s',t') \in R$

  $\mathcal{F}_\sim$ is monotonic ; R bisim $\Leftrightarrow R \subseteq \mathcal{F}_\sim(R)$

  $\sim \; = \bigsqcup \{ R \mid R \sqsubseteq \mathcal{F}_\sim(R) \} = gfp(\mathcal{F}_\sim)$

HML with a single Variable

- Syntax : Simply we add X as basic formula
- Semantics : In open formulas (where X is still "undefined") X is interpreted as a "mathematical" variable

  Interpetation of X by $Proc_X \subseteq Proc$

  Semantics of F by $\Theta_F : \underset{\text{meaning of } X}{\mathcal{P}(Proc)} \to \underset{\text{meaning of } F(X)}{\mathcal{P}(Proc)}$

  Extending $\Theta_X = Id_{Proc}$ by HML operators.

  Θ_X is monotonic Because we have not negation !

- Semantics of Recursive Formulas

  $X = F(X)$ Two ! natural candidates $\begin{cases} gfp(\Theta_F) \\ lfp(\Theta_F) \end{cases}$

  $\left.\begin{array}{l} \text{For } Inv(F) \text{ we expect } gfp \\ \text{For } Pos(F) \text{ we expect } lfp \end{array}\right\}$ $\begin{array}{l} X_{Inv} \overset{max}{=} F \wedge [Act] F \\ X_{Pos} \overset{min}{=} F \vee \langle Act \rangle F \end{array}$

  $\vee$ cannot be delayed forever

- Games for the satisfaction relation $s \overset{?}{\models} F$

  The attacker wins disproving it

  The defender wins proving it

  The attacker plays at $\wedge$ and $[a]$ formulas

  The defender plays at $\vee$ and $\langle a \rangle$ formulas

When $\begin{Bmatrix} X \overset{max}{=} F(X) \\ X \overset{min}{=} F(X) \end{Bmatrix}$ the $\begin{Bmatrix} \text{defender} \\ \text{attacker} \end{Bmatrix}$ wins any infinite play

Minimal formula = Inductive meaning = Finite proofs

Maximal formula = Coinductive meaning = Possible "infinite" (coinductive) proofs

- Some interesting examples

Safeness : there is some path along which F is preserved

$$\text{Safe}(F): \quad X_{safe} \overset{max}{=} F \wedge ([Act]ff \vee \langle Act \rangle F)$$

any finite path is broken

termination is good

must be preserved by some continuation

Liveness : F will be satisfied at some time along any path...

$$\text{Even}(F): \quad X_{even} \overset{min}{=} F \vee (\langle Act \rangle tt \wedge [Act] X_{even})$$

we need to get it finitely

we get it if we initially have it

otherwise we need to look for it in the (finite) future

we fail if we cannot continue

we must get it for any continuation

Getting G through F

We get G

We have F

We continue in the same way

$$F \, U^w G : \quad X_w \overset{max}{=} G \vee^{or} (F \wedge^{and} [Act] X_w)$$

$$F \, U^s G : \quad X_s \overset{min}{=} G \vee (F \wedge \langle Act \rangle tt \wedge [Act] X_s)$$

we need to get G finitely

we need to have F in the meantime

we fail if we cannot continue

We continue in the same way

- Meaning of Formulas with Several Variables

No problem if <u>all of them</u> are either min o max

No problem if there is not mutual recursion

The semantics of variables are obtained in the adequate order

No semantics ! if there is mutual recursion combining both min and max .