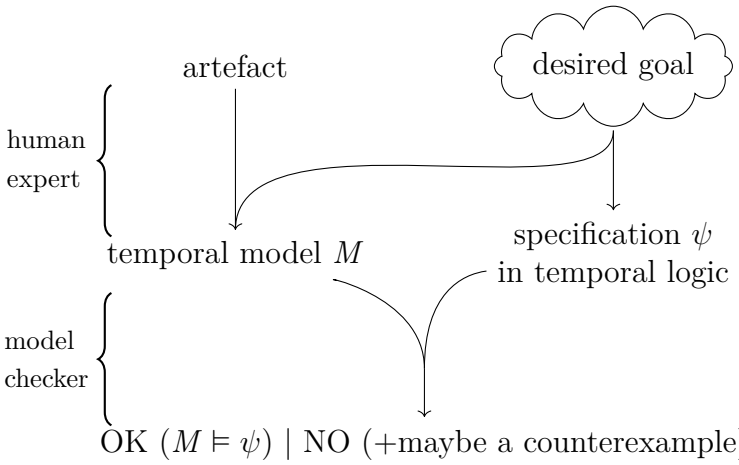
Supporting material for Model checking

Jean Pichon-Pharabod

2019/2020

This document contains the nitty-gritty details.

Motivation



This diagram gives a very static, top-down picture, but it is the feedback that provides the value.

7 Temporal models

7.1 Definition

$$
\begin{array}{l}
AP, \ldots \in \mathsf{Set} \\
\mathsf{TModel} \in \mathsf{Set} \to \mathsf{Set} \\
M, \ldots \in \mathsf{TModel}\ AP \stackrel{def}{=} \\
\quad (S \in \mathsf{Set}) \times \qquad\qquad\qquad\qquad\qquad\qquad \text{states} \\
\quad\quad (S_0 \in S \to \mathsf{Prop}) \times \qquad\qquad\qquad\qquad \text{initial states} \\
\quad\quad\quad (\text{-}\ T \in S \to S \to \mathsf{Prop}) \times \qquad\qquad \text{transition} \\
\quad\quad\quad\quad (\ell \in S \to AP \to \mathsf{Prop}) \times \qquad\quad \text{state labelling} \\
\quad\quad\quad\quad\quad (\forall s \in S.\, \exists s' \in S.\, s\ T\ s') \qquad \text{left-total}
\end{array}
$$

Elements of AP are denoted $p, \ldots$.
Elements of S are denoted $s, \ldots$.

7.1.1 Remarks

Some definitions require S to be finite.

Some definitions require ℓ to be boolean-valued, and interpret s not being labelled with p as s being labelled with $\neg p$. However, this is not compatible with abstraction.

7.2 Corner cases

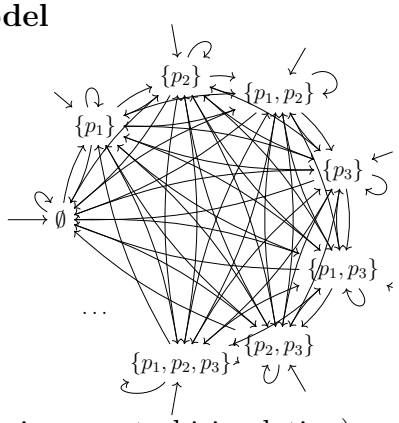
7.2.1 The initial temporal model

$$
\begin{array}{l}
\mathbb{0}\text{-} \in (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP \\
\mathbb{0}_{AP} \stackrel{def}{=} \left\langle \begin{array}{l} \mathbb{0}, \\ s \mapsto \dagger, \\ s_0 \mapsto s_1 \mapsto \dagger, \\ s \mapsto p \mapsto \dagger, \\ \ldots \end{array} \right\rangle \qquad\qquad \textit{(it is empty)}
\end{array}
$$

7.2.2 The terminal temporal model

$$\mathbb{1}\_ \in (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP$$

$$\mathbb{1}_{AP} \stackrel{def}{=} \left\langle \begin{array}{l} AP \to \mathbb{B}, \\ s \mapsto \top, \\ s_0 \mapsto s_1 \mapsto \top, \\ s \mapsto p \mapsto s\ p, \\ \ldots \end{array} \right\rangle$$



Exercise: It is not unique! (only unique up to bisimulation) — can you find another (interestingly different) one?

7.3 Useful notions

7.3.1 Paths

(Infinite) **paths**

$$\mathsf{IsPath} \in (AP \in \mathsf{Set}) \to (M \in \mathsf{TModel}\ AP) \to \mathsf{stream}\ (M.S) \to \mathsf{Prop}$$
$$\mathsf{IsPath}\ AP\ M\ \pi \stackrel{def}{=} (\forall n \in \mathbb{N}.\, (\pi\ n)\ M.T\ (\pi\ (n+1)))$$

$$\mathsf{Path} \in (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP \to \mathsf{Set}$$
$$\mathsf{Path}\ AP\ M \stackrel{def}{=} (\pi \in \mathsf{stream}\ M.S) \times \mathsf{IsPath}\ AP\ M\ \pi$$

7.3.2 Reachable states

Because the transition relation is left-total, these infinite paths are "complete", in that they coincide with reachability:

$$\mathsf{Reachable} \in (AP \in \mathsf{Set}) \to (M \in \mathsf{TModel}\ AP) \to M.S \to \mathsf{Prop}$$
$$\mathsf{Reachable}\ AP\ M\ s \stackrel{def}{=} \exists \pi \in \mathsf{stream}\ M.S, n \in \mathbb{N}.\ \mathsf{IsPath}\ AP\ M\ \pi \wedge M.S_0\ (\pi\ 0) \wedge s = \pi\ n$$

7.3.3 Stuttering

A temporal model is **stuttering** when all states loop back to themselves:

$$\begin{array}{l}\mathsf{stuttering} \ \in (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP \to \mathsf{Prop} \\ \mathsf{stuttering}\ AP\ M \stackrel{def}{=} \forall s \in M.S.\ s\ M.T\ s\end{array}$$

If the temporal model is not stuttering, then we can count transitions. This is only sound if they exactly match those of the system being analysed [**?**].

7.4 Temporal models from operational semantics

$$\begin{array}{lcl} C, \ldots \in \mathsf{Cmd} & ::= & \ldots \\ \sigma, \ldots \in \mathsf{Stack} & \stackrel{def}{=} & \mathsf{Var} \to \mathbb{Z} \\ \mathsf{Cfg} & \stackrel{def}{=} & \mathsf{Cmd} \times \mathsf{Stack} \\ \mathsf{step} & \in & \mathsf{Cfg} \to \mathsf{Cfg} \to \mathsf{Prop} \end{array}$$

Interesting atomic properties could be along the lines of

$$\begin{array}{lcl} X, Y, Z, \ldots & \in & \mathsf{Var} \\ v & \in & \mathbb{Z} \\ AP & ::= & X \dot{=} v \mid \\ & & X \dot{=} Y \mid \\ & & X \dot{<} Y \mid \\ & & X \dot{+} Y \dot{<} Z \mid \\ & & X \dot{\times} Y \dot{<} Z \mid \\ & & \ldots \end{array}$$

We write $s \vDash^{\mathsf{AP}} p$ when a stack s satisfies p.

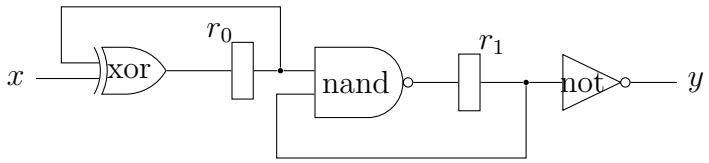
Given an initial stack σ_0, we can build

$$\left\langle \begin{array}{l} \mathsf{Cfg}, \\ s \mapsto s = \langle C_0, \sigma_0 \rangle, \\ s_0 \mapsto s_1 \mapsto \mathsf{step}\ s_0\ s_1, \\ s \mapsto p \mapsto s.\mathsf{stack} \vDash^{\mathsf{AP}} p, \\ \ldots \end{array} \right\rangle$$

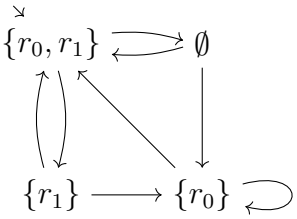
7.5 Temporal models from circuits

7.5.1 other example circuit 1

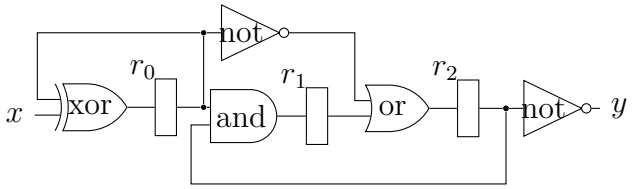
Another circuit, with input arity 1.



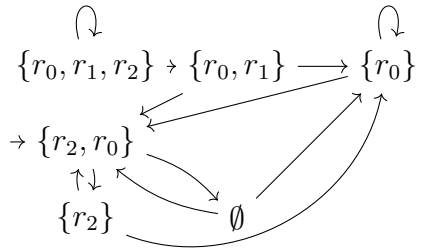
Assuming the registers are initially both set to 1:



7.5.2 Other example circuit



Assuming registers r_0 and r_2 are initially both set to 1, and r_1 is initially set to 0:



7.5.3 Statics

We consider a very simple circuit language:

$$
\begin{array}{l}
G, \ldots \in \mathsf{Gate} ::= \mathtt{id} \mid \mathtt{join} \mid \\
\qquad\qquad\qquad\quad \mathtt{not} \mid \mathtt{and} \mid \mathtt{or} \mid \mathtt{xor} \mid \mathtt{nor} \mid \mathtt{nand} \mid \\
\qquad\qquad\qquad\quad \mathtt{reg} \mid \mathtt{in} \mid \mathtt{out} \\
C, \ldots \in \mathsf{SCircuit}\ (i, o \in \mathbb{N}) \stackrel{def}{=} \\
\quad (N \in \mathsf{Set}) \times \\
\qquad (I \in \mathsf{NoDupList}\ N) \times \\
\qquad\quad (O \in \mathsf{NoDupList}\ N) \times \\
\qquad\qquad (\ell \in N \to \mathsf{Gate}) \times \\
\qquad\qquad\quad (w \in N \to N \to \mathsf{Prop}) \times \\
\qquad\qquad\qquad (\mathsf{wf}\ N\ I\ O\ \ell\ w)
\end{array}
$$

Well-formedness condition for simple circuits:

$$
\begin{array}{l}
\mathsf{wf}\ N\ I\ O\ \ell\ w \stackrel{def}{=} \\
\ \mathsf{disjoint}\ I\ O \wedge \\
\ (\forall n, n'.\, w\ n\ n' \to w\ n'\ n) \wedge \\
\ \left(\forall n, n', n'' \in N. \left(\begin{array}{l} w\ n\ n' \wedge \\ w\ n\ n'' \wedge \\ \ell\ n = \mathtt{not} \end{array} \right) \to n' = n'' \right) \wedge \\
\ \ldots
\end{array}
$$

7.5.4 Types of the dynamics

Defining the dynamic semantics is most straightforwardly done using a temporal model.

$$
\begin{array}{ll}
\mathsf{Reg} & \in \mathsf{SCircuit} \to \mathsf{Set} \\
\mathsf{Reg}\ C & \stackrel{def}{=} (n \in C.N) \times (C.\ell\ n = \mathtt{reg}) \\
\mathsf{IO} & \in \mathsf{SCircuit} \to \mathsf{Set} \\
\mathsf{IO}\ C & \stackrel{def}{=} (n \in C.N) \times (C.\ell\ n = \mathtt{in} \vee_{\mathbb{B}} C.\ell\ n = \mathtt{out}) \\
\mathsf{Regs} & \in \mathsf{SCircuit} \to \mathsf{Set} \\
\mathsf{Regs}\ C & \stackrel{def}{=} \mathbb{1} \to \mathsf{Reg}\ C \\
\mathsf{IOs} & \in \mathsf{SCircuit} \to \mathsf{Set} \\
\mathsf{IOs}\ C & \stackrel{def}{=} \mathbb{1} \to \mathsf{IO}\ C \\
\mathsf{RMap} & \in \mathsf{SCircuit} \to \mathsf{Set} \\
s, \ldots \in \mathsf{RMap}\ C & \stackrel{def}{=} \mathsf{Regs}\ C \to \mathbb{B} \\
I, \ldots \in \mathsf{InputValuation}\ C & \stackrel{def}{=} (n \in C.N) \to (C.\ell\ n = \mathtt{in}) \to \mathbb{B}
\end{array}
$$

7.5.5 Dynamics

$$\mathsf{T} \quad \in \quad (i, o \in \mathbb{N}) \to (C \in \mathsf{SCircuit}\ i\ o) \to \mathsf{InputValuation}\ C \to \mathsf{RMap}\ C \to \mathsf{RMap}\ C \to \mathsf{Prop}$$

$\mathsf{T}\ i\ o\ C\ I\ s\ s' \stackrel{def}{=}$

$\exists W \in (n \in C.N) \to (n' \in C.N) \to (C.w\ n\ n' = \top_{\mathbb{B}}) \to \mathbb{B}.$

the wire valuation agrees with the input wire valuation

$(\forall n, n'.\, C.\ell\ n = \texttt{in} \to W\ n\ n' = I\ n) \wedge$

the wire outgoing value agrees with the previous register state

$(\forall n, n'.\, C.\ell\ n = \texttt{reg} \wedge C.w\ i\ o\ n\ n' \to W\ n\ n' = s\ n) \wedge$

the wire outgoing value for an `and` is the conjunction of the wire ingoing values

$(\forall n_1 n_2, n_3, n_4.\, C.\ell\ n_3 = \texttt{and} \to W\ n_3\ n_4 = W\ n_1\ n_3 \wedge_{\mathbb{B}} W\ n_2\ n_3) \wedge$

and the corresponding conditions for other operators...

the wire ingoing value agrees with the new register state

$(\forall n, n'.\, C.\ell\ n' = \texttt{reg} \wedge C.w\ n\ n' \to W\ n\ n' = s'\ n')$

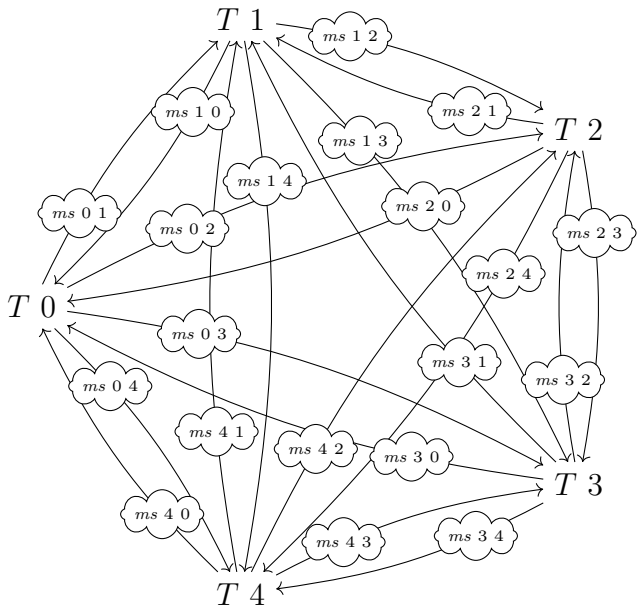
???internal model

$$\mathsf{model} \in (i, o \in \mathbb{N}) \to \mathsf{SCircuit}\ i\ o \to (AP \in \mathsf{Set}) \times \mathsf{TModel}\ AP$$

$$\mathsf{model}\ i\ o\ C \stackrel{def}{=} \left\langle \mathsf{Regs}\ C, \left\langle \begin{array}{l} \mathsf{RMap}\ C, \\ (n \mapsto \bot_{\mathbb{B}}), \\ s \mapsto s' \mapsto \exists I.\, \mathsf{T}\ i\ o\ C\ I\ s\ s', \\ (s \mapsto n \mapsto s\ n = \top_{\mathbb{B}}), \\ \ldots \end{array} \right\rangle \right\rangle$$

or we could have $\mathsf{IOs}\ C$ as the labels, giving a ??? view

7.6 Distributed algorithms operational semantics



If we have

$$
\begin{array}{l}
m, \ldots \in \mathsf{Msg} \in \mathsf{Set} \\
st, \ldots \in \mathsf{NodeState} \in \mathsf{Set} \\
i, \ldots \in \mathsf{NodeId} \in \mathsf{Set} \\
\rightarrow_{\mathsf{node}} \in \ (\mathsf{NodeState} \times \mathsf{Id} \times \mathsf{Msg}) \rightarrow \\
\qquad\qquad \mathsf{NodeState} \times (\mathsf{NodeId} \rightarrow \mathsf{Msg} \rightarrow \mathsf{Prop}) \rightarrow \mathsf{Prop}
\end{array}
$$

then we can define

$$
\begin{array}{rl}
\mathsf{NetworkState} \stackrel{def}{=} & (\mathsf{NodeId} \rightarrow \mathsf{NodeState}) \times \\
& (\mathsf{NodeId} \rightarrow \mathsf{NodeId} \rightarrow \mathsf{Msg} \rightarrow \mathsf{Prop})
\end{array}
$$

$$
\frac{\forall i, i', m.\, X'\ i\ i'\ m \rightarrow X\ i\ i'\ m}{\langle \mathcal{T}, X \rangle \rightarrow \langle \mathcal{T}, X' \rangle}
$$

$$
\frac{\begin{array}{c} \langle T\ i, i, m \rangle \rightarrow_{\mathsf{node}} \langle S', ms \rangle \\ \forall i', i'', m.\, X'\ i'\ i''\ m \leftrightarrow (X\ i'\ i''\ m \vee (i' = i \wedge \mathsf{ms}\ i''\ m)) \end{array}}{\langle \mathcal{T}, X \rangle \rightarrow \langle \mathcal{T}[i \mapsto S'], X' \rangle}
$$

7.7 Temporal models from cryptographic protocols

[?]

8 Temporal logics

8.1 Syntax of CTL$^{*\mathbf{IF}}$

$$
\begin{array}{l}
\mathsf{StateProp}^{\mathsf{IF}} \in \mathsf{Set} \to \mathsf{Set} \\
\psi^{\mathsf{IF}}, \ldots \in \mathsf{StateProp}^{\mathsf{IF}}\ AP ::= \\
\quad \begin{array}{lll}
\bot^{\mathsf{IF}} & | & \text{false} \\
\top^{\mathsf{IF}} & | & \text{true} \\
\psi_1^{\mathsf{IF}} \wedge^{\mathsf{IF\text{-}S}} \psi_2^{\mathsf{IF}} & | & \text{conjunction} \\
\psi_1^{\mathsf{IF}} \vee^{\mathsf{IF\text{-}S}} \psi_2^{\mathsf{IF}} & | & \text{disjunction} \\
\mathsf{injp}^{\mathsf{IF}}\ p & | & \text{atomic predicate} \\
\mathsf{A}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & | & \text{universal} \\
\mathsf{E}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & & \text{existential}
\end{array}
\end{array}
\qquad
\begin{array}{l}
\mathsf{PathProp}^{\mathsf{IF}} \in \mathsf{Set} \to \mathsf{Set} \\
\phi^{\mathsf{IF}}, \ldots \in \mathsf{PathProp}^{\mathsf{IF}}\ AP ::= \\
\quad \begin{array}{lll}
\phi_1^{\mathsf{IF}} \wedge^{\mathsf{IF\text{-}P}} \phi_2^{\mathsf{IF}} & | & \text{conjunction} \\
\phi_1^{\mathsf{IF}} \vee^{\mathsf{IF\text{-}P}} \phi_2^{\mathsf{IF}} & | & \text{disjunction} \\
\mathsf{injs}^{\mathsf{IF}}\ \psi^{\mathsf{IF}} & | & \text{state property} \\
\mathsf{X}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & | & \text{next} \\
\mathsf{F}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & | & \text{future} \\
\mathsf{G}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & | & \text{generally} \\
\phi_1^{\mathsf{IF}}\ \mathsf{U}^{\mathsf{IF}}\ \phi_2^{\mathsf{IF}} & & \text{until}
\end{array}
\end{array}
$$

8.2 Semantics of CTL$^{*\mathbf{IF}}$

We define whether M satisfies ψ,

$$
\begin{array}{c}
\_ \vDash^{\mathsf{IF}}_{\_} \_ \quad \in (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP \to \\
\qquad \mathsf{StateProp}^{\mathsf{IF}}\ AP \to \mathsf{Prop} \\
M \vDash^{\mathsf{IF}}_{AP} \psi^{\mathsf{IF}} \stackrel{def}{=} \forall s \in M.S.\ M.S_0\ s \to s \vDash^{\mathsf{IF\text{-}S}}_{AP,M} \psi^{\mathsf{IF}}
\end{array}
$$

using two auxiliary mutually inductive predicates

$$
\begin{array}{c}
\_ \vDash^{\mathsf{IF\text{-}S}}_{\_,\_} \_ \in (AP \in \mathsf{Set}) \to (M \in \mathsf{TModel}\ AP) \to \\
\qquad M.S \to \mathsf{StateProp}^{\mathsf{IF}}\ AP \to \mathsf{Prop} \\
\_ \vDash^{\mathsf{IF\text{-}P}}_{\_,\_} \_ \in (AP \in \mathsf{Set}) \to (M \in \mathsf{TModel}\ AP) \to \\
\qquad \mathsf{stream}\ M.S \to \mathsf{PathProp}^{\mathsf{IF}}\ AP \to \mathsf{Prop}
\end{array}
$$

8.2.1 Semantics of CTL$^{*\mathsf{IF}}$: state properties

$$
\begin{array}{ll}
s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \top^{\mathsf{IF}} & \stackrel{def}{=} \top \\
s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \bot^{\mathsf{IF}} & \stackrel{def}{=} \bot \\
s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \psi_1^{\mathsf{IF}} \wedge^{\mathsf{IF\text{-}s}} \psi_2^{\mathsf{IF}} & \stackrel{def}{=} \left(s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \psi_1^{\mathsf{IF}}\right) \wedge \left(s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \psi_2^{\mathsf{IF}}\right) \\
s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \psi_1^{\mathsf{IF}} \vee^{\mathsf{IF\text{-}s}} \psi_2^{\mathsf{IF}} & \stackrel{def}{=} \left(s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \psi_1^{\mathsf{IF}}\right) \vee \left(s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \psi_2^{\mathsf{IF}}\right) \\
s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \mathsf{injp}^{\mathsf{IF}}\ p & \stackrel{def}{=} M.\ell\ s\ p \\
s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \mathsf{A}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & \stackrel{def}{=} \left(\begin{array}{l} \forall \pi \in \mathsf{stream}\ M.S. \\ \quad \mathsf{IsPath}\ AP\ M\ \pi \rightarrow \pi\ 0 = s \rightarrow \pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi^{\mathsf{IF}} \end{array}\right) \\
s \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \mathsf{E}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & \stackrel{def}{=} \left(\begin{array}{l} \exists \pi \in \mathsf{stream}\ M.S. \\ \quad \mathsf{IsPath}\ AP\ M\ \pi \wedge \pi\ 0 = s \wedge \\ \quad \pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi^{\mathsf{IF}} \end{array}\right)
\end{array}
$$

8.2.2 Semantics of CTL$^{*\mathsf{IF}}$: path properties

$$
\begin{array}{ll}
\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \mathsf{injs}^{\mathsf{IF}}\ \psi & \stackrel{def}{=} (\pi\ 0) \vDash^{\mathsf{IF\text{-}s}}_{AP,M} \psi^{\mathsf{IF}} \\
\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_1^{\mathsf{IF}} \wedge^{\mathsf{IF\text{-}p}} \phi_2^{\mathsf{IF}} & \stackrel{def}{=} \left(\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_1^{\mathsf{IF}}\right) \wedge \left(\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_2^{\mathsf{IF}}\right) \\
\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_1^{\mathsf{IF}} \vee^{\mathsf{IF\text{-}p}} \phi_2^{\mathsf{IF}} & \stackrel{def}{=} \left(\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_1^{\mathsf{IF}}\right) \vee \left(\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_2^{\mathsf{IF}}\right) \\
\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \mathsf{X}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & \stackrel{def}{=} (\mathsf{tailn}\ M.S\ 1\ \pi) \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi^{\mathsf{IF}} \\
\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \mathsf{F}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & \stackrel{def}{=} \exists n \in \mathbb{N}.\, (\mathsf{tailn}\ M.S\ n\ \pi) \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi^{\mathsf{IF}} \\
\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \mathsf{G}^{\mathsf{IF}}\ \phi^{\mathsf{IF}} & \stackrel{def}{=} \forall n \in \mathbb{N}.\, (\mathsf{tailn}\ M.S\ n\ \pi) \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi^{\mathsf{IF}} \\
\pi \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_1^{\mathsf{IF}}\ \mathsf{U}^{\mathsf{IF}}\ \phi_2^{\mathsf{IF}} \stackrel{def}{=} & \\
\quad \exists n \in \mathbb{N}. \left(\begin{array}{l} \left(\forall k \in \mathbb{N}.\, 0 \leq k < n \rightarrow (\mathsf{tailn}\ M.S\ k\ \pi) \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_1^{\mathsf{IF}}\right) \wedge \\ (\mathsf{tailn}\ M.S\ n\ \pi) \vDash^{\mathsf{IF\text{-}p}}_{AP,M} \phi_2^{\mathsf{IF}} \end{array}\right) &
\end{array}
$$

8.3 Semantics of CTL$^{*\mathbf{WI}}$

8.3.1 Definite temporal model

A **definite temporal model**

$$
\begin{array}{l}
\mathsf{DTModel} \in \mathsf{Set} \to \mathsf{Set} \\
DM, \ldots \in \mathsf{DTModel}\ AP \stackrel{def}{=} \\
\quad (S \in \mathsf{Set}) \times \\
\quad\quad (F \in \mathsf{finType}\ S) \times \\
\quad\quad\quad (S_0 \in S \to \mathbb{B}) \times \\
\quad\quad\quad\quad (① T ② \in S \to S \to \mathbb{B}) \times \\
\quad\quad\quad\quad\quad (\ell \in S \to AP \to \mathbb{B}) \times \\
\quad\quad\quad\quad\quad\quad (\forall s \in S.\, \exists s' \in S.\, s\ T\ s' = \top_{\mathbb{B}})
\end{array}
$$

Any definite temporal model on AP induces a (plain) temporal model on AP by "forgetting":

$$
\mathsf{injectDM} \in (AP \in \mathsf{Set}) \to \mathsf{DTModel}\ AP \to \mathsf{TModel}\ AP
$$

$$
\mathsf{injectDM}\ AP\ DM = \left\langle \begin{array}{l} DM.S, \\ s \mapsto (DM.S_0\ s) = \top_{\mathbb{B}}, \\ s_0 \mapsto s_1 \mapsto (s_0\ DM.T\ s_1) = \top_{\mathbb{B}}, \\ s \mapsto p \mapsto M.\ell\ s\ p = \top_{\mathbb{B}}, \\ \ldots \end{array} \right\rangle
$$

$$
\begin{array}{l}
\mathsf{reflect\text{-}model} \in (AP \in \mathsf{Set}) \to \mathsf{DTModel}\ AP \to \mathsf{TModel}\ AP \to \mathsf{Prop} \\
\mathsf{reflect\text{-}model}\ AP\ DM\ M \stackrel{def}{=} \\
\quad M.S = DM.S \wedge \\
\quad (\forall s.\, \mathsf{reflect}\ (DM.S_0\ s)\ (M.S_0\ s)) \wedge \\
\quad (\forall s_0, s_1.\, \mathsf{reflect}\ (DM.T\ s_0\ s_1)\ (M.T\ s_0\ s_1)) \wedge \\
\quad (\forall s, p.\, \mathsf{reflect}\ (DM.\ell\ s\ p)\ (M.\ell\ s\ p))
\end{array}
$$

$$
\begin{array}{l}
\forall AP \in \mathsf{Set}, DM \in \mathsf{DTModel}\ AP. \\
\quad \mathsf{reflect\text{-}model}\ AP\ DM\ (\mathsf{injectDM}\ AP\ DM)
\end{array}
$$

8.3.2 Syntax of CTL$^{*\mathbf{WI}}$

$$
\begin{array}{l}
\mathsf{StateProp}^{\mathsf{WI}} \in \mathsf{Set} \to \mathsf{Set} \\
\psi^{\mathsf{WI}}, \ldots \in \mathsf{StateProp}^{\mathsf{WI}}\ AP ::= \\
\quad \bot^{\mathsf{WI}} \mid \top^{\mathsf{WI}} \mid \neg^{\mathsf{WI\text{-}S}} \psi \mid \psi_1^{\mathsf{WI}} \wedge^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}} \mid \psi_1^{\mathsf{WI}} \vee^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}} \mid \psi_1^{\mathsf{WI}} \to^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}} \mid \\
\quad \mathsf{injp}^{\mathsf{WI}}\ p \mid \mathsf{A}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} \mid \mathsf{E}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}
\end{array}
$$

$$
\begin{array}{l}
\mathsf{PathProp}^{\mathsf{WI}} \in \mathsf{Set} \to \mathsf{Set} \\
\phi^{\mathsf{WI}}, \ldots \in \mathsf{PathProp}^{\mathsf{WI}}\ AP ::= \\
\quad \neg^{\mathsf{WI\text{-}P}} \phi \mid \phi_1^{\mathsf{WI}} \wedge^{\mathsf{WI\text{-}P}} \phi_2^{\mathsf{WI}} \mid \phi_1^{\mathsf{WI}} \vee^{\mathsf{WI\text{-}P}} \phi_2^{\mathsf{WI}} \mid \phi_1^{\mathsf{WI}} \to^{\mathsf{WI\text{-}P}} \phi_2^{\mathsf{WI}} \mid \\
\quad \mathsf{injs}^{\mathsf{WI}}\ \psi \mid \mathsf{X}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} \mid \mathsf{F}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} \mid \mathsf{G}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} \mid \phi_1^{\mathsf{WI}}\ \mathsf{U}^{\mathsf{WI}}\ \phi_2^{\mathsf{WI}}
\end{array}
$$

8.3.3 Injecting CTL$^{*\mathbf{IF}}$ into CTL$^{*\mathbf{WI}}$

$$
\begin{array}{l}
\mathsf{injectStateProp} \in (AP \in \mathsf{Set}) \to \mathsf{StateProp}^{\mathsf{IF}}\ AP \to \mathsf{StateProp}^{\mathsf{WI}}\ AP \\
\mathsf{injectStateProp}\ AP\ \top^{\mathsf{IF}} \stackrel{def}{=} \top^{\mathsf{WI}} \\
\mathsf{injectStateProp}\ AP\ \bot^{\mathsf{IF}} \stackrel{def}{=} \bot^{\mathsf{WI}} \\
\mathsf{injectStateProp}\ AP\ (\psi_1^{\mathsf{IF}} \wedge^{\mathsf{IF\text{-}S}} \psi_2^{\mathsf{IF}}) \stackrel{def}{=} \\
\quad (\mathsf{injectStateProp}\ AP\ \psi_1^{\mathsf{IF}}) \wedge^{\mathsf{WI\text{-}S}} (\mathsf{injectStateProp}\ AP\ \psi_2^{\mathsf{IF}}) \\
\mathsf{injectStateProp}\ AP\ (\psi_1^{\mathsf{IF}} \vee^{\mathsf{IF\text{-}S}} \psi_2^{\mathsf{IF}}) \stackrel{def}{=} \\
\quad (\mathsf{injectStateProp}\ AP\ \psi_1^{\mathsf{IF}}) \vee^{\mathsf{WI\text{-}S}} (\mathsf{injectStateProp}\ AP\ \psi_2^{\mathsf{IF}}) \\
\ldots \\
\mathsf{injectPathProp} \in (AP \in \mathsf{Set}) \to \mathsf{PathProp}^{\mathsf{IF}}\ AP \to \mathsf{PathProp}^{\mathsf{WI}}\ AP \\
\ldots
\end{array}
$$

8.3.4 Semantics of CTL$^{*\mathbf{WI}}$

We define whether DM satisfies ψ,

$$
\begin{array}{rl}
= \models^{\mathsf{WI}}_{\_} \equiv & \in (AP \in \mathsf{Set}) \to \mathsf{DTModel}\ AP \to \\
& \quad \mathsf{StateProp}^{\mathsf{WI}}\ AP \to \mathsf{Prop} \\
DM \models^{\mathsf{WI}}_{AP} \psi^{\mathsf{WI}} \stackrel{def}{=} \forall s \in S.\, DM.S_0\ s \to s \models^{\mathsf{WI\text{-}S}}_{AP,DM} \psi^{\mathsf{WI}} &
\end{array}
$$

using two auxiliary predicates

$$\begin{aligned}
\equiv \vDash^{\mathsf{WI\text{-}S}}_{_,=} \equiv \; &\in \; (AP \in \mathsf{Set}) \to (DM \in \mathsf{DTModel}\ AP) \to \\
&\quad DM.S \to \mathsf{StateProp}^{\mathsf{WI}}\ AP \to \mathsf{Prop} \\
\equiv \vDash^{\mathsf{WI\text{-}P}}_{_,=} \equiv \; &\in \; (AP \in \mathsf{Set}) \to (DM \in \mathsf{TModel}\ AP) \to \\
&\quad \mathsf{stream}\ DM.S \to \mathsf{PathProp}^{\mathsf{WI}}\ AP \to \mathsf{Prop}
\end{aligned}$$

which we define mutually inductively

$$\begin{aligned}
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \top^{\mathsf{WI}} &\stackrel{def}{=} \top \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \bot^{\mathsf{WI}} &\stackrel{def}{=} \bot \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \neg^{\mathsf{WI\text{-}S}} \psi^{\mathsf{WI}} &\stackrel{def}{=} \neg \left(s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi^{\mathsf{WI}}\right) \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_1^{\mathsf{WI}} \wedge^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}} &\stackrel{def}{=} \left(s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_1^{\mathsf{WI}}\right) \wedge \left(s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_2^{\mathsf{WI}}\right) \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_1^{\mathsf{WI}} \vee^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}} &\stackrel{def}{=} \left(s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_1^{\mathsf{WI}}\right) \vee \left(s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_2^{\mathsf{WI}}\right) \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_1^{\mathsf{WI}} \to^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}} &\stackrel{def}{=} \left(\neg \left(s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_1^{\mathsf{WI}}\right)\right) \vee \left(s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi_2^{\mathsf{WI}}\right) \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \mathsf{injp}^{\mathsf{WI}}\ p &\stackrel{def}{=} DM.\ell\ s\ p = \top_{\mathbb{B}} \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \mathsf{A}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} &\stackrel{def}{=} \forall \pi \in \mathsf{stream}\ DM.S. \\
&\qquad \mathsf{IsPath}\ AP\ DM\ \pi \to \pi\ 0 = s \to \pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi^{\mathsf{WI}} \\
s \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \mathsf{E}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} &\stackrel{def}{=} \exists \pi \in \mathsf{stream}\ DM.S. \\
&\qquad \mathsf{IsPath}\ AP\ DM\ \pi \wedge \pi\ 0 = s \wedge \\
&\qquad \pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi^{\mathsf{WI}}
\end{aligned}$$

$$
\begin{aligned}
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \mathsf{injs}^{\mathsf{WI}}\ \psi^{\mathsf{WI}} &\stackrel{def}{=} (\pi\ 0) \vDash^{\mathsf{WI\text{-}S}}_{AP,DM} \psi^{\mathsf{WI}} \\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \neg^{\mathsf{WI\text{-}P}} \phi^{\mathsf{WI}} &\stackrel{def}{=} \neg \left(\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi^{\mathsf{WI}}\right) \\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_1^{\mathsf{WI}} \wedge^{\mathsf{WI\text{-}P}} \phi_2^{\mathsf{WI}} &\stackrel{def}{=} \left(\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_1^{\mathsf{WI}}\right) \wedge \left(\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_2^{\mathsf{WI}}\right) \\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_1^{\mathsf{WI}} \vee^{\mathsf{WI\text{-}P}} \phi_2^{\mathsf{WI}} &\stackrel{def}{=} \left(\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_1^{\mathsf{WI}}\right) \vee \left(\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_2^{\mathsf{WI}}\right) \\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_1^{\mathsf{WI}} \rightarrow^{\mathsf{WI\text{-}P}} \phi_2^{\mathsf{WI}} &\stackrel{def}{=} \left(\neg \left(\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_1^{\mathsf{WI}}\right)\right) \vee \left(\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_2^{\mathsf{WI}}\right) \\
\\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \mathsf{X}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} &\stackrel{def}{=} (\mathsf{tailn}\ DM.S\ 1\ \pi) \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi^{\mathsf{WI}} \\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \mathsf{F}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} &\stackrel{def}{=} \exists n \in \mathbb{N}.\, (\mathsf{tailn}\ DM.S\ n\ \pi) \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi^{\mathsf{WI}} \\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \mathsf{G}^{\mathsf{WI}}\ \phi^{\mathsf{WI}} &\stackrel{def}{=} \forall n \in \mathbb{N}.\, (\mathsf{tailn}\ M.S\ n\ \pi) \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi^{\mathsf{WI}} \\
\pi \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_1^{\mathsf{WI}}\ \mathsf{U}^{\mathsf{WI}}\ \phi_2^{\mathsf{WI}} &\stackrel{def}{=} \\
\exists n \in \mathbb{N}.\, &\left(\begin{array}{l}\left(\forall k \in \mathbb{N}.\, 0 \leq k < n \rightarrow (\mathsf{tailn}\ DM.S\ k\ \pi) \vDash^{\mathsf{WI\text{-}P}}_{AP,M} \phi_1^{\mathsf{WI}}\right) \wedge \\ (\mathsf{tailn}\ DM.S\ n\ \pi) \vDash^{\mathsf{WI\text{-}P}}_{AP,DM} \phi_2^{\mathsf{WI}}\end{array}\right)
\end{aligned}
$$

Because we are working with definite temporal models, we have

$$
\begin{array}{l}\forall AP \in \mathsf{Set}, DM \in \mathsf{DTModel}\ AP, \psi^{\mathsf{WI}} \in \mathsf{StateProp}^{\mathsf{WI}}\ AP. \\ \quad (DM \vDash^{\mathsf{WI}}_{AP} \neg^{\mathsf{WI\text{-}S}}(\neg^{\mathsf{WI\text{-}S}}\psi^{\mathsf{WI}})) \rightarrow DM \vDash^{\mathsf{WI}}_{AP} \psi^{\mathsf{WI}}\end{array}
$$

(a proof of this is a model checker — see Lecture 11).

This means that implication collapses:

$$
\begin{array}{l}\forall AP \in \mathsf{Set}, DM \in \mathsf{DTModel}\ AP, \psi_1^{\mathsf{WI}}, \psi_2^{\mathsf{WI}} \in \mathsf{StateProp}^{\mathsf{WI}}\ AP. \\ \quad (DM \vDash^{\mathsf{WI}}_{AP} \psi_1^{\mathsf{WI}} \rightarrow^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}}) \rightarrow (DM \vDash^{\mathsf{WI}}_{AP} (\neg^{\mathsf{WI\text{-}S}}\psi_1^{\mathsf{WI}}) \vee^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}})\end{array}
$$

and all the De Morgan laws hold:

$$
\begin{array}{l}\forall AP \in \mathsf{Set}, DM \in \mathsf{DTModel}\ AP, \psi_1^{\mathsf{WI}}, \psi_2^{\mathsf{WI}} \in \mathsf{StateProp}^{\mathsf{WI}}\ AP. \\ \quad (DM \vDash^{\mathsf{WI}}_{AP} \neg^{\mathsf{WI\text{-}S}}(\psi_1^{\mathsf{WI}} \wedge^{\mathsf{WI\text{-}S}} \psi_2^{\mathsf{WI}})) \rightarrow (DM \vDash^{\mathsf{WI}}_{AP} (\neg^{\mathsf{WI\text{-}S}}\psi_1^{\mathsf{WI}}) \vee^{\mathsf{WI\text{-}S}} (\neg^{\mathsf{WI\text{-}S}}\psi_2^{\mathsf{WI}})) \\ \ldots\end{array}
$$

and so any formula is equivalent to its negation normal form — which is implication-free, and therefore on which we can use abstraction — see Lecture 10.

8.3.5 Embedding CTL$^{*\mathsf{WI}}$ in CTL$^{*\mathsf{IF}}$

Why not always use $\mathsf{CTL}^{*\mathsf{WI}}$? Because it conflates not being labelled with p with being labelled with $\neg p$!
And (1) this is not preserved by abstraction (2) provability is not preserved: there is no embedding of IF in WI that preserves provability
On the other hand, we can embed $\mathsf{CTL}^{*\mathsf{WI}}$ in $\mathsf{CTL}^{*\mathsf{IF}}$ in a way that preserves provability[1].

$$
\begin{array}{l}
\mathsf{Inductive\ split}\ (AP \in \mathsf{Set}) \in \mathsf{Set} := \\
\quad \oplus\text{-} \in AP \to \mathsf{split}\ AP \\
\mid \ominus\text{-} \in AP \to \mathsf{split}\ AP
\end{array}
$$

$$
\begin{array}{ll}
\mathsf{nf}^{\mathsf{s}} & \in (AP \in \mathsf{Set}) \to \\
& \quad \mathsf{StateProp}^{\mathsf{WI}}\ AP \to \\
& \quad\quad \mathsf{StateProp}^{\mathsf{IF}}\ (\mathsf{split}\ AP) \\
\mathsf{nf}^{\mathsf{s}}\ AP\ \bot^{\mathsf{WI}} & \stackrel{def}{=} \bot^{\mathsf{IF}} \\
\mathsf{nf}^{\mathsf{s}}\ AP\ \top^{\mathsf{WI}} & \stackrel{def}{=} \bot^{\mathsf{IF}} \\
\mathsf{nf}^{\mathsf{s}}\ AP\ (\neg^{\mathsf{WI-S}} \psi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi^{\mathsf{WI}} \\
\mathsf{nf}^{\mathsf{s}}\ AP\ (\psi_1^{\mathsf{WI}} \wedge^{\mathsf{WI-S}} \psi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi_1^{\mathsf{WI}}) \wedge^{\mathsf{IF-S}} (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi_2^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{s}}\ AP\ (\psi_1^{\mathsf{WI}} \vee^{\mathsf{WI-S}} \psi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi_1^{\mathsf{WI}}) \vee^{\mathsf{WI-S}} (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi_2^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{s}}\ AP\ (\psi_1^{\mathsf{WI}} \to^{\mathsf{WI-S}} \psi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi_1^{\mathsf{WI}}) \vee^{\mathsf{WI-S}} (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi_2^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{s}}\ AP\ (\mathsf{injp}^{\mathsf{WI}}\ p) & \stackrel{def}{=} \mathsf{injp}^{\mathsf{IF}}\ (\oplus p) \\
\mathsf{nf}^{\mathsf{s}}\ AP\ (\mathsf{A}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{A}^{\mathsf{IF}}\ (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{s}}\ AP\ (\mathsf{E}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{E}^{\mathsf{IF}}\ (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}})
\end{array}
$$

[1] This is reminiscent of how intuitionistic logic is more expressive that classical logic

$$
\begin{array}{ll}
\mathsf{nf}^{\mathsf{p}} & \in (AP \in \mathsf{Set}) \to \mathsf{PathProp}^{\mathsf{WI}}\ AP \to \\
 & \quad \mathsf{PathProp}^{\mathsf{IF}}\ (\mathsf{split}\ AP) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\mathsf{injs}^{\mathsf{WI}}\ \psi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{injs}^{\mathsf{IF}}\ (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\neg^{\mathsf{WI-p}} \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}} \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}} \wedge^{\mathsf{WI-p}} \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}}) \wedge^{\mathsf{WI-s}} (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}} \vee^{\mathsf{WI-p}} \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}}) \vee^{\mathsf{WI-p}} (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}} \to^{\mathsf{WI-p}} \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}}) \vee^{\mathsf{WI-p}} (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\mathsf{X}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{X}^{\mathsf{IF}}\ (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\mathsf{F}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{F}^{\mathsf{IF}}\ (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\mathsf{G}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{G}^{\mathsf{IF}}\ (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}}\ \mathsf{U}^{\mathsf{WI}}\ \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}})\ \mathsf{U}^{\mathsf{IF}}\ (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}})
\end{array}
$$

Negation can be defined as an operation:

$$
\begin{array}{ll}
\mathsf{nf\text{-}neg}^{\mathsf{s}} & \in (AP \in \mathsf{Set}) \to \mathsf{StateProp}^{\mathsf{WI}}\ AP \to \\
 & \quad \mathsf{StateProp}^{\mathsf{IF}}\ (\mathsf{split}\ AP) \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \bot^{\mathsf{WI}} & \stackrel{def}{=} \top^{\mathsf{IF}} \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \top^{\mathsf{WI}} & \stackrel{def}{=} \bot^{\mathsf{IF}} \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ (\neg^{\mathsf{WI-s}} \psi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{nf}^{\mathsf{p}}\ AP\ \psi^{\mathsf{WI}} \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ (\psi_1^{\mathsf{WI}} \wedge^{\mathsf{WI-s}} \psi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi_1^{\mathsf{WI}}) \vee^{\mathsf{IF-s}} (\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi_2^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ (\psi_1^{\mathsf{WI}} \vee^{\mathsf{WI-s}} \psi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi_1^{\mathsf{WI}}) \wedge^{\mathsf{IF-s}} (\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi_2^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ (\psi_1^{\mathsf{WI}} \to^{\mathsf{WI-s}} \psi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi_1^{\mathsf{WI}}) \wedge^{\mathsf{IF-s}} (\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi_2^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ (\mathsf{injp}^{\mathsf{WI}}\ p) & \stackrel{def}{=} \mathsf{injp}^{\mathsf{IF}}\ (\ominus p) \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ (\mathsf{A}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{E}^{\mathsf{IF}}\ (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ (\mathsf{E}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{A}^{\mathsf{IF}}\ (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}})
\end{array}
$$

$$
\begin{array}{ll}
\mathsf{nf\text{-}neg}^{\mathsf{p}} & \in (AP \in \mathsf{Set}) \to \mathsf{PathProp}^{\mathsf{IF}}\ AP \to \\
& \quad \mathsf{PathProp}^{\mathsf{WI}}\ (\mathsf{split}\ AP) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\mathsf{injs}^{\mathsf{WI}}\ \psi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{injs}^{\mathsf{IF}}\ (\mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\neg^{\mathsf{WI\text{-}p}} \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{nf}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}} \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}} \wedge^{\mathsf{WI\text{-}p}} \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}}) \vee^{\mathsf{IF\text{-}p}} (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}} \vee^{\mathsf{WI\text{-}p}} \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}}) \wedge^{\mathsf{IF\text{-}p}} (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}} \to^{\mathsf{WI\text{-}p}} \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}}) \wedge^{\mathsf{IF\text{-}p}} (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\mathsf{X}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{X}^{\mathsf{IF}}\ (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\mathsf{F}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{F}^{\mathsf{IF}}\ (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\mathsf{G}^{\mathsf{WI}}\ \phi^{\mathsf{WI}}) & \stackrel{def}{=} \mathsf{G}^{\mathsf{IF}}\ (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi^{\mathsf{WI}}) \\
\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ (\phi_1^{\mathsf{WI}}\ \mathsf{U}^{\mathsf{WI}}\ \phi_2^{\mathsf{WI}}) & \stackrel{def}{=} (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_1^{\mathsf{WI}})\ \mathsf{U}^{\mathsf{IF}}\ (\mathsf{nf\text{-}neg}^{\mathsf{p}}\ AP\ \phi_2^{\mathsf{WI}})
\end{array}
$$

$$
\mathsf{nf\text{-}model} \in (AP \in \mathsf{Set}) \to \mathsf{DTModel}\ AP \to \mathsf{DTModel}\ (\mathsf{split}\ AP)
$$

$$
\mathsf{nf\text{-}model}\ AP\ M = \left\langle \begin{array}{lll}
M.S, & & \\
M.S_0, & & \\
M.T, & & \\
s \mapsto p^{\#} \mapsto & \mathsf{match}\ p^{\#}\ \mathsf{with} & , \\
 & \oplus p \mapsto M.\ell\ s\ p & \\
 & \ominus p \mapsto \neg_{\mathbb{B}}(M.\ell\ s\ p) & \\
\ldots & &
\end{array} \right\rangle
$$

$$
\begin{array}{l}
\forall AP \in \mathsf{Set}, DM \in \mathsf{DTModel}\ AP, \psi \in \mathsf{StateProp}^{\mathsf{WI}}\ AP. \\
\left(\begin{array}{l} (DM \vDash^{\mathsf{WI}}_{AP} \psi) \leftrightarrow \\ \quad ((\mathsf{nf\text{-}model}\ AP\ M) \vDash^{\mathsf{WI}}_{AP} \mathsf{injectStateProp}\ (\mathsf{nf}^{\mathsf{s}}\ (\mathsf{split}\ AP)\ \psi)) \end{array} \right) \wedge \\
\left(\begin{array}{l} (DM \vDash^{\mathsf{WI}}_{AP} \psi) \leftrightarrow \\ \quad (\mathsf{injectDM}\ (\mathsf{nf\text{-}model}\ AP\ M) \vDash^{\mathsf{IF}}_{AP} (\mathsf{nf}^{\mathsf{s}}\ (\mathsf{split}\ AP)\ \psi)) \end{array} \right)
\end{array}
$$

8.4 Universal and existential state properties

$$
\begin{array}{ll}
\mathsf{us} & \in (AP \in \mathsf{Set}) \to \mathsf{StateProp}^{\mathsf{WI}}\ AP \to \mathbb{B} \\
\mathsf{us}\ AP\ \hat{\bot} & \stackrel{def}{=} \top_{\mathbb{B}} \\
\mathsf{us}\ AP\ \hat{\top} & \stackrel{def}{=} \top_{\mathbb{B}} \\
\mathsf{us}\ AP\ (\hat{\neg}\psi) & \stackrel{def}{=} \mathsf{es}\ AP\ \psi \\
\mathsf{us}\ AP\ (\psi_1\ \hat{\wedge}\ \psi_2) & \stackrel{def}{=} \mathsf{us}\ AP\ \psi_1 \wedge_{\mathbb{B}} \mathsf{us}\ AP\ \psi_2 \\
\mathsf{us}\ AP\ (\psi_1\ \hat{\vee}\ \psi_2) & \stackrel{def}{=} \mathsf{us}\ AP\ \psi_1 \wedge_{\mathbb{B}} \mathsf{us}\ AP\ \psi_2 \\
\mathsf{us}\ AP\ (\psi_1 \hat{\rightarrow} \psi_2) & \stackrel{def}{=} \mathsf{us}\ AP\ \psi_1 \wedge_{\mathbb{B}} \mathsf{us}\ AP\ \psi_2 \\
\mathsf{us}\ AP\ (\mathsf{injp}\ p) & \stackrel{def}{=} \top_{\mathbb{B}} \\
\mathsf{us}\ AP\ (\mathsf{A}\ \phi) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi \\
\mathsf{us}\ AP\ (\mathsf{E}\ \phi) & \stackrel{def}{=} \bot_{\mathbb{B}}
\end{array}
$$

$$
\begin{array}{ll}
\mathsf{es} & \in (AP \in \mathsf{Set}) \to \mathsf{StateProp}^{\mathsf{WI}}\ AP \to \mathbb{B} \\
\mathsf{es}\ AP\ \hat{\bot} & \stackrel{def}{=} \top_{\mathbb{B}} \\
\mathsf{es}\ AP\ \hat{\top} & \stackrel{def}{=} \top_{\mathbb{B}} \\
\mathsf{es}\ AP\ (\hat{\neg}\psi) & \stackrel{def}{=} \mathsf{us}\ AP\ \psi \\
\mathsf{es}\ AP\ (\psi_1\ \hat{\wedge}\ \psi_2) & \stackrel{def}{=} \mathsf{es}\ AP\ \psi_1 \wedge_{\mathbb{B}} \mathsf{es}\ AP\ \psi_2 \\
\mathsf{es}\ AP\ (\psi_1\ \hat{\vee}\ \psi_2) & \stackrel{def}{=} \mathsf{es}\ AP\ \psi_1 \wedge_{\mathbb{B}} \mathsf{es}\ AP\ \psi_2 \\
\mathsf{es}\ AP\ (\psi_1 \hat{\rightarrow} \psi_2) & \stackrel{def}{=} \mathsf{es}\ AP\ \psi_1 \wedge_{\mathbb{B}} \mathsf{es}\ AP\ \psi_2 \\
\mathsf{es}\ AP\ (\mathsf{injp}\ p) & \stackrel{def}{=} \top_{\mathbb{B}} \\
\mathsf{es}\ AP\ (\mathsf{A}\ \phi) & \stackrel{def}{=} \bot_{\mathbb{B}} \\
\mathsf{es}\ AP\ (\mathsf{E}\ \phi) & \stackrel{def}{=} \mathsf{ep}\ AP\ \phi
\end{array}
$$

$$
\begin{array}{ll}
\mathsf{up} & \in (AP \in \mathsf{Set}) \to \mathsf{PathProp}^{\mathsf{WI}}\ AP \to \mathbb{B} \\
\mathsf{up}\ AP\ (\tilde{\neg}\phi) & \stackrel{def}{=} \mathsf{ep}\ AP\ \phi \\
\mathsf{up}\ AP\ (\phi_1\ \tilde{\wedge}\ \phi_2) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{up}\ AP\ \phi_2 \\
\mathsf{up}\ AP\ (\phi_1\ \tilde{\vee}\ \phi_2) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{up}\ AP\ \phi_2 \\
\mathsf{up}\ AP\ (\phi_1 \tilde{\to} \phi_2) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{up}\ AP\ \phi_2 \\
\mathsf{up}\ AP\ (\mathsf{injs}\ \psi) & \stackrel{def}{=} \mathsf{us}\ AP\ \psi \\
\mathsf{up}\ AP\ (\mathsf{X}\ \phi) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi \\
\mathsf{up}\ AP\ (\mathsf{F}\ \phi) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi \\
\mathsf{up}\ AP\ (\mathsf{G}\ \phi) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi \\
\mathsf{up}\ AP\ (\phi_1\ \mathsf{U}\ \phi_2) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{up}\ AP\ \phi_2
\end{array}
$$

$$
\begin{array}{ll}
\mathsf{ep} & \in (AP \in \mathsf{Set}) \to \mathsf{PathProp}^{\mathsf{WI}}\ AP \to \mathbb{B} \\
\mathsf{ep}\ AP\ (\tilde{\neg}\phi) & \stackrel{def}{=} \mathsf{up}\ AP\ \phi \\
\mathsf{ep}\ AP\ (\phi_1\ \tilde{\wedge}\ \phi_2) & \stackrel{def}{=} \mathsf{ep}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{ep}\ AP\ \phi_2 \\
\mathsf{ep}\ AP\ (\phi_1\ \tilde{\vee}\ \phi_2) & \stackrel{def}{=} \mathsf{ep}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{ep}\ AP\ \phi_2 \\
\mathsf{ep}\ AP\ (\phi_1 \tilde{\to} \phi_2) & \stackrel{def}{=} \mathsf{ep}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{ep}\ AP\ \phi_2 \\
\mathsf{ep}\ AP\ (\mathsf{injs}\ \psi) & \stackrel{def}{=} \mathsf{es}\ AP\ \psi \\
\mathsf{ep}\ AP\ (\mathsf{X}\ \phi) & \stackrel{def}{=} \mathsf{ep}\ AP\ \phi \\
\mathsf{ep}\ AP\ (\mathsf{F}\ \phi) & \stackrel{def}{=} \mathsf{es}\ AP\ \phi \\
\mathsf{ep}\ AP\ (\mathsf{G}\ \phi) & \stackrel{def}{=} \mathsf{es}\ AP\ \phi \\
\mathsf{ep}\ AP\ (\phi_1\ \mathsf{U}\ \phi_2) & \stackrel{def}{=} \mathsf{ep}\ AP\ \phi_1 \wedge_{\mathbb{B}} \mathsf{ep}\ AP\ \phi_2
\end{array}
$$

8.5 Quantifiers

Unlike in Hoare logic, there are no quantifiers, as they would make it difficult to mechanically check properties.

To make up for this, we can use property schemas with big operators or bounded quantifiers, and indexed atomic propositions, which stand for the expanded property.

For example $\bigwedge_{i=0}^{n} p_i$, for $n = 3$, is expanded to $p_1 \wedge p_2 \wedge p_3$.

So is $\bigwedge_{i \in S} p_i$, for $S = \{1, 2, 3\}$.

This is is not as general as quantifiers, as the value of n or S has to be known. Because this is done as a preprocessing phase, it does not change the language of properties.

9 Using model checking

10 Relating models

10.1 Simulation

R is a **temporal model simulation** of M by M':

$$\begin{array}{l} \_ \preccurlyeq^{\equiv}_{\_} \_ \in \quad (AP \in \mathsf{Set}) \to (M \in \mathsf{TModel}\ AP) \to (M' \in \mathsf{TModel}\ AP) \to \\ \qquad\qquad (M.S \to M'.S \to \mathsf{Prop}) \to \mathsf{Prop} \\ M \preccurlyeq^{R}_{AP} M' \stackrel{def}{=} \end{array}$$

R is consistent with labels:

$$\left(\begin{array}{l} \forall s \in M.S, s' \in M'.S. \\ \quad s\ R\ s' \to \forall p \in AP.\ M'.\ell\ s'\ p \to M.\ell\ s\ p \end{array} \right) \wedge$$

R relates initial states of M to initial states in M':

$$(\forall s \in M.S.\ M.S_0\ s \to \exists s' \in M'.S.\ M.S_0\ s' \wedge s\ R\ s') \wedge$$

any step in M can be matched by a step in M' from any R-related start state to some R-related end state:

$$\left(\begin{array}{l} \forall s_0, s_1 \in M.S, s'_0 \in M'.S. \\ \quad s_0\ M.T\ s_1 \wedge s_0\ R\ s'_0 \to \\ \qquad \exists s'_1 \in M'.S. \\ \qquad\quad s'_0\ M'.T\ s'_1 \wedge s_1\ R\ s'_1 \end{array} \right)$$

$$\begin{array}{ccc} s_0 & \overset{R}{\dashrightarrow} & s'_0 \\ {\scriptstyle M.T}\downarrow & & \\ s_1 & & \end{array} \quad \to \quad \exists s'_1.\ \begin{array}{ccc} s_0 & \overset{R}{\dashrightarrow} & s'_0 \\ {\scriptstyle M.T}\downarrow & & \downarrow{\scriptstyle M'.T} \\ s_1 & \underset{R}{\dashrightarrow} & s'_1 \end{array}$$

(A **simulation** just requires condition 3, and is generally defined for labelled transition systems.)

The identity relation is a simulation:

$$\begin{array}{l}\forall AP \in \mathsf{Set}, M \in \mathsf{TModel}\ AP.\\ \quad \mathsf{let}\ R = (s \mapsto s)\ \mathsf{in}\\ \quad M \preccurlyeq^{R}_{AP} M\end{array}$$

The terrible punter can simulate any punter that respects physics (does not teleport goats, etc.).

10.2 Temporal model simulation

The details of the simulation are not so important, what matters is the existence of a simulation:

$$\begin{array}{l}② \preccurlyeq_{①} ③ \in (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP \to \mathsf{TModel}\ AP \to \mathsf{Prop}\\ (M \preccurlyeq_{AP} M') \stackrel{def}{=} \exists R.\, M \preccurlyeq^{R}_{AP} M'\end{array}$$

it means that M' is "more abstract" than M: it may have more behaviour, making it less precise, but that allows it to have possibly fewer states and transitions.

10.3 Temporal model simulation preorder

$$\begin{array}{l}\mathsf{reflexive} \in (A \in \mathsf{Set}) \to (A \to A \to \mathsf{Prop}) \to \mathsf{Prop}\\ \mathsf{reflexive}\ A\ P \stackrel{def}{=} \forall a \in A.\, P\ a\ a\end{array}$$

$$\begin{array}{l}\mathsf{transitive} \in (A \in \mathsf{Set}) \to (A \to A \to \mathsf{Prop}) \to \mathsf{Prop}\\ \mathsf{transitive}\ A\ P \stackrel{def}{=} \forall a_1, a_2, a_3 \in A.\, P\ a_1\ a_2 \to P\ a_2\ a_3 \to P\ a_1\ a_3\end{array}$$

$$\begin{array}{l}\mathsf{Preorder} \stackrel{def}{=}\\ \quad (S \in \mathsf{Set}) \times\\ \quad\quad (① \sqsubseteq ② \in S \to S \to \mathsf{Prop}) \times\\ \quad\quad\quad \mathsf{reflexive}\ S\ (\sqsubseteq) \times\\ \quad\quad\quad\quad \mathsf{transitive}\ S\ (\sqsubseteq)\end{array}$$

$$\mathsf{TModelPreorder} \in \mathsf{Set} \to \mathsf{Preorder}$$

$$\mathsf{TModelPreorder}\ AP \stackrel{def}{=} \left\langle \begin{array}{l} \mathsf{TModel}\ AP, \\ \preccurlyeq_{AP}, \\ \ldots, \\ \ldots \end{array} \right\rangle$$

10.4 Model simulation preorder category

Given atomic propositions AP, the simulation preorder - $\preccurlyeq_{AP}$ - induces a preorder category. The initial and terminal objects are the initial and terminal temporal models.

$$\forall M.\, \mathbb{0}_{AP} \preccurlyeq_{AP} M$$
$$\forall M.\, M \preccurlyeq_{AP} \mathbb{1}_{AP}$$
$$\forall M, M', M''.\, M \preccurlyeq_{AP} M \wedge M \preccurlyeq_{AP} M'' \to M \preccurlyeq_{AP} M' \times M''$$

10.5 Simulation preserves universal, implication-free propositions

$\mathsf{ACTL}^{*\mathsf{IF}}$ is compatible with the simulation preorder:

$$\forall AP \in \mathsf{Set}, M \in \mathsf{TModel}\ AP, M' \in \mathsf{TModel}\ AP, \psi \in \mathsf{StateProp}^{\mathsf{ACTL}^{*}\mathsf{IF}}\ AP.$$
$$(M \preccurlyeq_{AP} M' \wedge \mathsf{us}\ AP\ \psi \wedge M' \vDash_{AP} \psi) \to M \vDash_{AP} \psi$$

It suffices to show the property holds of the more abstract model to know it holds of the more concrete model.

However, not all interesting properties are "nice" in this sense, and care will have to be taken to make M' precise enough for the other properties we care about.

This property can seem strange, because $\mathsf{F}\ \phi$ has an existential feel to it. It is very fragile, and really depends on left-totality.

It is also possible to define a more "precise" notion of temporal model simulation that requires the abstract model to agree exactly on labels, and that preserved all of ACTL.

10.6 Temporal model bisimulation

R is a **temporal model bisimulation** of M by M':

$$\_ \approx^{\equiv}_{\_} \_ \in \ (AP \in \mathsf{Set}) \to (M \in \mathsf{TModel}\ AP) \to (M' \in \mathsf{TModel}\ AP) \to (M.S \to M'.S \to \mathsf{Prop}) \to \mathsf{Prop}$$

$$M \approx^R_{AP} M' \stackrel{def}{=} M \preccurlyeq^R_{AP} M' \wedge M' \preccurlyeq^R_{AP} M$$

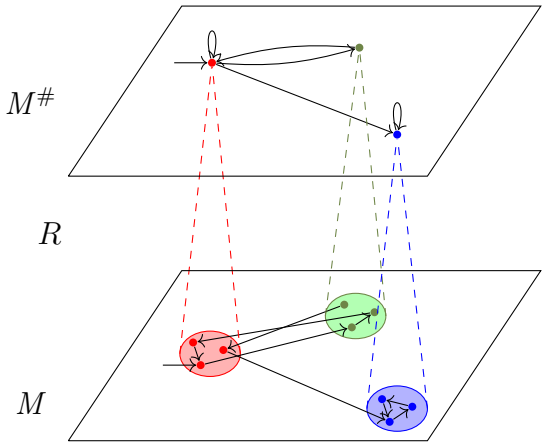
As for simulations, the details of the bisimulation are not so important, what matters is the existence of a bisimulation:

$$\_ \approx_{\_} \_ \in (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP \to \mathsf{TModel}\ AP \to \mathsf{Prop}$$

$$(M \approx_{AP} M') \stackrel{def}{=} \exists R.\ M \approx^R_{AP} M'$$

All of CTL^* is compatible with the bisimulation equivalence:

$$\forall AP \in \mathsf{Set}, M \in \mathsf{TModel}\ AP, M' \in \mathsf{TModel}\ AP, \psi \in \mathsf{StateProp}^{\mathsf{IF}}\ AP.$$
$$M \approx_{AP} M' \to (M \vDash_{AP} \psi \leftrightarrow M' \vDash_{AP} \psi)$$



11 Implementing model checking

We will see how to implement the world's worst CTL model checker. For the model checker to be effective, the input temporal model needs to be effective.

11.1 Specifying a CTL model checker

$$\mathsf{mc} \in (AP \in \mathsf{Set}) \to \mathsf{DTModel}\ AP \to \mathsf{StateProp}^{\mathsf{CTL}}\ AP \to \mathbb{B}$$

$$\forall AP \in \mathsf{Set}, DM \in \mathsf{DTModel}\ AP, \psi^{\mathsf{CTL}} \in \mathsf{StateProp}^{\mathsf{CTL}}\ AP.$$
$$\mathsf{reflect}\ (\mathsf{mc}\ AP\ DM\ \psi^{\mathsf{CTL}})\ (DM \models^{\mathsf{WI}}_{AP} \psi^{\mathsf{CTL}})$$

11.2 Implementing model checking

$$\mathsf{mc}\ AP\ DM\ \psi^{\mathsf{CTL}} \stackrel{def}{=}$$
$$\mathsf{forall\text{-}fin}\ DM.S\ (s \mapsto DM.S_0\ s \to_{\mathbb{B}} \mathsf{mca}\ DM\ \psi^{\mathsf{CTL}}\ s)$$

$$\mathsf{mca} \in (AP \in \mathsf{Set}) \to (DM \in \mathsf{DTModel}\ AP) \to$$
$$\mathsf{StateProp}^{\mathsf{CTL}}\ AP \to (DM.S \to \mathbb{B})$$

$$\forall AP \in \mathsf{Set}, DM \in \mathsf{DTModel}\ AP, \psi^{\mathsf{CTL}} \in \mathsf{StateProp}^{\mathsf{CTL}}\ AP, s \in M.S.$$
$$\mathsf{reflect}\ (\mathsf{mca}\ AP\ DM\ \psi^{\mathsf{CTL}}\ s)\ (s \models^{\mathsf{WI\text{-}S}}_{AP,DM} \psi^{\mathsf{CTL}})$$

11.3 CTL model checker: propositional fragment

$$
\begin{array}{lcl}
\mathsf{mca}\ AP\ DM\ p & \stackrel{def}{=} & s \mapsto DM.\ell\ s\ p \\
\mathsf{mca}\ AP\ DM\ (\hat{\neg}\phi^{\mathsf{CTL}}) & \stackrel{def}{=} & \mathsf{let}\ V = \mathsf{mca}\ AP\ DM\ \phi^{\mathsf{CTL}}\ \mathsf{in} \\
 & & s \mapsto \neg_{\mathbb{B}}(V\ s) \\
\mathsf{mca}\ AP\ DM\ (\phi_1^{\mathsf{CTL}}\ \hat{\wedge}\ \phi_2^{\mathsf{CTL}}) & \stackrel{def}{=} & \mathsf{let}\ V_1 = \mathsf{mca}\ AP\ DM\ \phi_1^{\mathsf{CTL}}\ \mathsf{in} \\
 & & \mathsf{let}\ V_2 = \mathsf{mca}\ AP\ DM\ \phi_2^{\mathsf{CTL}}\ \mathsf{in} \\
 & & s \mapsto V_1\ s \wedge_{\mathbb{B}} V_2\ s \\
\mathsf{mca}\ AP\ DM\ (\phi_1^{\mathsf{CTL}}\ \hat{\vee}\ \phi_2^{\mathsf{CTL}}) & \stackrel{def}{=} & \mathsf{let}\ V_1 = \mathsf{mca}\ AP\ DM\ \phi_1^{\mathsf{CTL}}\ \mathsf{in} \\
 & & \mathsf{let}\ V_2 = \mathsf{mca}\ AP\ DM\ \phi_2^{\mathsf{CTL}}\ \mathsf{in} \\
 & & s \mapsto V_1\ s \vee_{\mathbb{B}} V_2\ s \\
\mathsf{mca}\ AP\ DM\ (\phi_1^{\mathsf{CTL}}\ \hat{\rightarrow}\ \phi_2^{\mathsf{CTL}}) & \stackrel{def}{=} & \mathsf{let}\ V_1 = \mathsf{mca}\ AP\ DM\ \phi_1^{\mathsf{CTL}}\ \mathsf{in} \\
 & & \mathsf{let}\ V_2 = \mathsf{mca}\ AP\ DM\ \phi_2^{\mathsf{CTL}}\ \mathsf{in} \\
 & & s \mapsto V_1\ s \rightarrow_{\mathbb{B}} V_2\ s
\end{array}
$$

11.4 CTL model checker: next

If we know in which states ϕ^{CTL} holds, then we know in which states $\mathsf{X}\ \phi^{\mathsf{CTL}}$ holds: their predecessors:

$$
\begin{array}{l}
\mathsf{mca}\ AP\ DM\ (\mathsf{A}\ \mathsf{X}\ \phi^{\mathsf{CTL}}) \stackrel{def}{=} \\
\quad \mathsf{let}\ V = \mathsf{mca}\ AP\ DM\ \phi^{\mathsf{CTL}}\ \mathsf{in} \\
\quad s \mapsto \mathsf{forall\text{-}fin}\ DM.S\ (s' \mapsto s\ DM.T\ s' \rightarrow_{\mathbb{B}} V\ s') \\
\\
\mathsf{mca}\ AP\ M\ (\mathsf{E}\ \mathsf{X}\ \phi^{\mathsf{CTL}}) \stackrel{def}{=} \\
\quad \mathsf{let}\ V = \mathsf{mca}\ AP\ DM\ \phi^{\mathsf{CTL}}\ \mathsf{in} \\
\quad s \mapsto \mathsf{exists\text{-}fin}\ DM.S\ (s' \mapsto s\ DM.T\ s' \wedge_{\mathbb{B}} V\ s')
\end{array}
$$

11.5 CTL model checker: small paths

The remaining temporal operators talk about infinite paths.
But it is sufficient to consider paths smaller than the diameter of the model[2].

$$\begin{aligned}
&\mathsf{IsSmallPathFrom} \in \ (AP \in \mathsf{Set}) \to (DM \in \mathsf{DTModel}\ AP) \to DM.S \to \\
&\qquad\qquad \mathsf{list}\ DM.S \to \mathsf{Prop} \\
&\mathsf{IsSmallPathFrom}\ AP\ DM\ s\ \Pi \stackrel{def}{=} \\
&\quad (\mathsf{length}\ \Pi \leq \mathsf{size}\ DM.F) \wedge (\mathsf{nth}\ \Pi\ 0 = \mathsf{some}\ s) \wedge \\
&\quad (\mathsf{nth}\ \Pi\ (\mathsf{length}\ \Pi - 1) = \mathsf{some}\ s') \wedge (s'\ DM.T\ s) \wedge \\
&\quad \left(\forall n \in \mathbb{N}, s', s''. \left(\begin{array}{l} \mathsf{nth}\ \Pi\ n = \mathsf{some}\ s' \wedge \\ \mathsf{nth}\ \Pi\ (n+1) = \mathsf{some}\ s'' \end{array}\right) \to s'\ DM.T\ s'' = \top_{\mathbb{B}}\right)
\end{aligned}$$

And we can obtain all these paths:

$$\begin{aligned}
&\mathsf{small\text{-}paths\text{-}from} \in \ (AP \in \mathsf{Set}) \to (DM \in \mathsf{DTModel}\ AP) \to \\
&\qquad\qquad (s \in DM.S) \to \\
&\qquad\qquad\quad \mathsf{finType}\ (\mathsf{SmallPathFrom}\ AP\ DM\ s) \\
&\mathsf{small\text{-}paths\text{-}from} \stackrel{def}{=} \ldots
\end{aligned}$$

11.6 CTL model checker: generally

$$\begin{aligned}
&\mathsf{mca}\ AP\ DM\ (\mathsf{A\ G}\ \phi^{\mathsf{CTL}}) \stackrel{def}{=} \\
&\quad \mathsf{let}\ V = \mathsf{mca}\ AP\ DM\ \phi^{\mathsf{CTL}}\ \mathsf{in} \\
&\quad s \mapsto\ \mathsf{forall\text{-}fin} \\
&\qquad\qquad (\mathsf{small\text{-}paths\text{-}from}\ AP\ DM\ s) \\
&\qquad\qquad (\Pi \mapsto \mathsf{forall\text{-}list}\ \Pi\ (s' \mapsto V\ s'))
\end{aligned}$$

$$\begin{aligned}
&\mathsf{mca}\ AP\ DM\ (\mathsf{E\ G}\ \phi^{\mathsf{CTL}}) \stackrel{def}{=} \\
&\quad \mathsf{let}\ V = \mathsf{mca}\ AP\ DM\ \phi^{\mathsf{CTL}}\ \mathsf{in} \\
&\quad s \mapsto\ \mathsf{exists\text{-}fin} \\
&\qquad\qquad (\mathsf{small\text{-}paths\text{-}from}\ AP\ DM\ s) \\
&\qquad\qquad (\Pi \mapsto \mathsf{forall\text{-}list}\ \Pi\ (s' \mapsto V\ s'))
\end{aligned}$$

[2] reminiscent of the pumping lemma for automata.

11.7 CTL model checker: future

$$\mathsf{mca}\ AP\ DM\ (\mathsf{A}\ \mathsf{F}\ \phi^{\mathsf{CTL}}) \stackrel{def}{=} \ldots$$

$$\mathsf{mca}\ AP\ DM\ (\mathsf{E}\ \mathsf{F}\ \phi^{\mathsf{CTL}}) \stackrel{def}{=} \ldots$$

Left as an exercise.

11.8 CTL model checker: until

$$\begin{array}{l}\mathsf{mca}\ AP\ DM\ (\mathsf{A}\ (\phi_1^{\mathsf{CTL}}\ \mathsf{U}\ \phi_2^{\mathsf{CTL}})) \stackrel{def}{=} \\ \quad \mathsf{let}\ V_1 = \mathsf{mca}\ AP\ DM\ \phi_1^{\mathsf{CTL}}\ \mathsf{in} \\ \quad \mathsf{let}\ V_2 = \mathsf{mca}\ AP\ DM\ \phi_2^{\mathsf{CTL}}\ \mathsf{in} \\ \quad s \mapsto \left(\begin{array}{l}\mathsf{forall\text{-}fin}\ (\mathsf{small\text{-}paths\text{-}from}\ AP\ DM\ s) \\ \quad \left(\Pi \mapsto \left(\begin{array}{l}\mathsf{existi}\ \Pi \\ \quad \left(j\ s'' \mapsto \left(\begin{array}{l}(\mathsf{foralli}\ \Pi\ (i\ s' \mapsto j <_{\mathbb{B}} i \rightarrow_{\mathbb{B}} V_1\ s') \wedge_{\mathbb{B}} \\ V_2\ s''\end{array}\right)\right)\right)\end{array}\right)\right)\end{array}\right)\end{array}$$

$$\mathsf{mca}\ AP\ DM\ (\mathsf{E}\ (\phi_1^{\mathsf{CTL}}\ \mathsf{U}\ \phi_2^{\mathsf{CTL}})) \stackrel{def}{=} \ldots$$

Left as an exercise.

11.9 Counterexamples

Adapted from "Tree-Like Counterexamples in Model Checking" [**?**].

If the specification is not satisfied, and is in ACTL, then we can do better than just say "no": we can produce a counterexample.

The idea is that $M \not\models_{AP} \psi^{\mathsf{ACTL}}$ is equivalent to $M \models_{AP} \neg\psi^{\mathsf{ACTL}}$, which is itself equivalent to $\mathsf{nf\text{-}model}\ M \models_{AP} \mathsf{nf\text{-}neg}^{\mathsf{s}}\ AP\ \psi^{\mathsf{ACTL}}$, where the latter formula is (the embedding of a proposition) in ECTL???: it suffices to find a witness of that ECTL proposition.

11.10 Shape of ECTL witnesses

The shape of an ECTL witness:

$$
\begin{array}{l}
W, \ldots \in \mathsf{data}\ \mathsf{Witness}\ (AP \in \mathsf{Set})\ (M \in \mathsf{TModel}\ AP) \in \mathsf{Set} := \\
\quad \mathsf{wap} \in M.S \to \mathsf{Witness}\ AP\ M \\
\mid \mathsf{wand} \in \mathsf{Witness}\ AP\ M \to \mathsf{Witness}\ AP\ M \to \mathsf{Witness}\ AP\ M \\
\mid \mathsf{winjl} \in \mathsf{Witness}\ AP\ M \to \mathsf{Witness}\ AP\ M \\
\mid \mathsf{winjr} \in \mathsf{Witness}\ AP\ M \to \mathsf{Witness}\ AP\ M \\
\mid \mathsf{wX} \in M.S \to M.S \to \mathsf{Witness}\ AP\ M \to \mathsf{Witness}\ AP\ M \\
\mid \mathsf{wF} \in \mathsf{list}\ M.S \to \mathsf{Witness}\ AP\ M \to \mathsf{Witness}\ AP\ M \\
\mid \mathsf{wG} \in \mathsf{list}\ (M.S \times \mathsf{Witness}\ AP\ M) \to \mathsf{Witness}\ AP\ M \\
\mid \mathsf{wU} \in \quad \mathsf{list}\ (M.S \times \mathsf{Witness}\ AP\ M) \to M.S \to \mathsf{Witness}\ AP\ M \to \\
\qquad\qquad \mathsf{Witness}\ AP\ M
\end{array}
$$

11.10.1 Being an ECTL witness: atomic propositions

$$
\begin{array}{rl}
\_ \vDash \_ \equiv \_ \ \mathsf{wit\text{-}by}\ \equiv : & (AP \in \mathsf{Set}) \to (M \in \mathsf{TModel}\ AP) \to M.S \to \\
& (\psi \in \mathsf{StateProp}^{\mathsf{CTL}}\ AP) \to \mathsf{Witness}\ AP\ M\ s \to \\
& \mathsf{Prop}
\end{array}
$$

There are (on purpose) no cases for A ....

A witness for an atomic proposition is just that the atomic proposition holds immediately:

$s \vDash_{AP,M} p\ \mathsf{wit\text{-}by}\ W \stackrel{def}{=} M.\ell\ s\ p \wedge W = \mathsf{wap}\ AP\ M\ s$

11.10.2 Being an ECTL witness: next

A witness for next is a transition from the current state, and a witness that the sub-property holds from the end state:

$$s \vDash_{AP,M} \mathsf{E\ X}\ \psi\ \mathsf{wit\text{-}by}\ W \stackrel{def}{=}$$
$$\exists s' \in M.S, W' \in \mathsf{Witness}\ AP\ M.\ \left(\begin{array}{l} s\ M.T\ s' \wedge \\ s' \vDash_{AP,M} \psi\ \mathsf{wit\text{-}by}\ W' \wedge \\ W = \mathsf{wX}\ AP\ M\ s\ s'\ W' \end{array}\right)$$

11.10.3 Being an ECTL witness: future

$$\begin{array}{l} s \vDash_{AP,M} \mathsf{E\ F}\ \psi\ \mathsf{wit\text{-}by}\ W \stackrel{def}{=} \\ \quad \exists s' \in M.S, \Pi \in \mathsf{list}\ M.S, W' \in \mathsf{Witness}\ AP\ M. \\ \qquad \left(\begin{array}{l} \mathsf{IsSmallPathFrom}\ AP\ M\ s\ \Pi \wedge \\ \mathsf{last}\ \Pi = \mathsf{some}\ s' \wedge \\ s' \vDash_{AP,M} \psi\ \mathsf{wit\text{-}by}\ W' \wedge \\ W = \mathsf{wF}\ AP\ M\ s\ \Pi\ W' \end{array}\right) \end{array}$$

11.10.4 Being an ECTL witness: generally

$$\begin{array}{l} s \vDash_{AP,M} \mathsf{E\ G}\ \psi\ \mathsf{wit\text{-}by}\ W \stackrel{def}{=} \\ \quad \mathsf{let}\ T = (M.S \times \mathsf{Witness}\ AP\ M)\ \mathsf{in} \\ \quad \exists X \in \mathsf{list}\ T. \\ \qquad \left(\begin{array}{l} \mathsf{IsSmallPathFrom}\ AP\ M\ s\ X \wedge \\ (\exists i.\ (\mathsf{last}\ T\ X)\ M.T\ (\mathsf{nth}\ T\ X\ i)) \wedge \\ \left(\forall i \in \mathbb{N}, s' \in M.S, W' \in \mathsf{Witness}\ AP\ M\ s'.\ \left(\begin{array}{c} \left(\begin{array}{c} \mathsf{nth}\ T\ X\ i = \mathsf{some}\ \langle s', W' \rangle \rightarrow \\ s' \vDash_{AP,M} \psi\ \mathsf{wit\text{-}by}\ W' \end{array}\right) \end{array}\right)\right) \wedge \\ W = \mathsf{wG}\ AP\ M\ X) \end{array}\right) \end{array}$$

11.10.5 Being an ECTL witness: until

$$
\begin{array}{l}
s \vDash_{AP,M} \mathsf{E}\ \psi_1\ \mathsf{U}\ \psi_2\ \text{wit-by}\ W \stackrel{def}{=} \\
\quad \mathsf{let}\ T = (M.S \times \mathsf{Witness}\ AP\ M)\ \mathsf{in} \\
\quad \exists X \in \mathsf{list}\ T, s' \in M.S, W' \in \mathsf{Witness}\ AP\ M. \\
\quad\quad \left(\begin{array}{l}
\mathsf{IsSmallPathFrom}\ AP\ M\ s\ (X +\!\!+ [\langle s', W' \rangle]) \wedge \\
\left(\begin{array}{c} \forall i \in \mathbb{N}, s'' \in M.S, W'' \in \mathsf{Witness}\ AP\ M\ s'. \\ \left(\begin{array}{c} \mathsf{nth}\ T\ X\ i = \mathsf{some}\ \langle s'', W''' \rangle \rightarrow \\ s'' \vDash_{AP,M} \psi_1\ \text{wit-by}\ W'' \end{array} \right) \end{array} \right) \wedge \\
(s' \vDash_{AP,M} \psi_2\ \text{wit-by}\ W') \wedge \\
W = \mathsf{wU}\ AP\ M\ X\ s'\ W')
\end{array} \right)
\end{array}
$$

11.10.6 Being an ECTL witness: conjunction

$$
\begin{array}{l}
s \vDash_{AP,M} \psi_1 \hat{\wedge} \psi_2\ \text{wit-by}\ W \stackrel{def}{=} \\
\quad \exists W_1 \in \mathsf{Witness}\ AP\ M, W_2 \in \mathsf{Witness}\ AP\ M. \\
\quad \left(\begin{array}{l}
s \vDash_{AP,M} \psi_1\ \text{wit-by}\ W_1 \wedge s \vDash_{AP,M} \psi_2\ \text{wit-by}\ W_2 \wedge \\
W = \mathsf{wand}\ AP\ M\ W_1\ W_2
\end{array} \right)
\end{array}
$$

11.10.7 Being an ECTL witness: disjunction

$$
\begin{array}{l}
s \vDash_{AP,M} \psi_1 \hat{\vee} \psi_2\ \text{wit-by}\ W \stackrel{def}{=} \\
\quad \left(\begin{array}{l}
\exists W_1 \in \mathsf{Witness}\ AP\ M. \\
\quad \left(\begin{array}{l}
s \vDash_{AP,M} \psi_1\ \text{wit-by}\ W_1 \wedge \\
W = \mathsf{winjl}\ AP\ M\ W_1
\end{array} \right)
\end{array} \right) \vee \\
\quad \left(\begin{array}{l}
\exists W_2 \in \mathsf{Witness}\ AP\ M. \\
\quad \left(\begin{array}{l}
s \vDash_{AP,M} \psi_2\ \text{wit-by}\ W_2 \wedge \\
W = \mathsf{winjr}\ AP\ M\ W_2
\end{array} \right)
\end{array} \right)
\end{array}
$$

11.10.8 Satisfiability and existence of witnesses

The requirement for a DTModel is just a brutal way to require M to be finite (otherwise, the witness could be infinite, and we would need a coinductive

definition of a witness — but we would not be able to build them in general).

$$\begin{array}{l}\forall AP \in \mathsf{Set}, M \in \mathsf{TModel}\ AP, DM \in \mathsf{DTModel}\ AP, \\ \quad s \in M.S, \psi \in \mathsf{StateProp}^{\mathsf{CTL}}\ AP. \\ \quad\quad \mathsf{es}\ \psi \rightarrow \mathsf{reflect\text{-}model}\ AP\ M\ DM \rightarrow \\ \quad\quad\quad \left(\begin{array}{l} (s \models^{\mathsf{WI\text{-}S}}_{AP,M} \psi) \leftrightarrow \\ \left(\begin{array}{l} \exists W \in \mathsf{Witness}\ (\mathsf{split}\ AP)\ (\mathsf{nf\text{-}model}\ AP\ M). \\ \quad s \models_{(\mathsf{split}\ AP),(\mathsf{nf\text{-}model}\ AP\ M)} (\mathsf{nf}^{\mathsf{s}}\ AP\ \psi)\ \mathsf{wit\text{-}by}\ W \end{array} \right) \end{array} \right) \end{array}$$

Now, if we have $M \not\models_{AP} \psi^{\mathsf{ACTL}}$, there exists a corresponding W — and we can effectively find it by tweaking our model checking algorithm above (details elided).

11.10.9 Counterexamples beyond ACTL

Can we have counterexamples for more than just ACTL? Yes, for example, in LTL, counterexamples are just paths! But for fragments of CTL^* that are too expressive, they're often not very enlightening. Instead, focus has been mostly on making better counterexamples for common fragments.

11.10.10 Model checking LTL and CTL*

Requires a bit of machinery to check whether a state is visited infinitely often: Büchi automata.

We will not consider this further.

11.10.11 CEGAR (not examinable)

Assume that we have a way to automatically generate abstract models. Then we can take the following approach: recursively: pick an abstraction of the model
check the property in the abstract model
if it is true, happy
if it is false, is it a genuine counterexample?
try it on the base model: if it works, we have found a genuine counterexample
if it does not work, build an abstraction

11.11 Composing temporal models

11.11.1 For reference: synchronous product of two temporal models

$$
\begin{array}{rcl}
\equiv \times^{\mathsf{sync}}_{-,=} \equiv & \in & (AP \in \mathsf{Set}) \to \big(AP' \in \mathsf{Set}\big) \to \\
& & \quad \mathsf{TModel}\ AP \to \mathsf{TModel}\ AP' \to \\
& & \qquad \mathsf{TModel}\ (AP \times AP')
\end{array}
$$

$$
M \times^{\mathsf{sync}}_{AP,AP'} M' \stackrel{def}{=} \left\langle \begin{array}{l}
M.S \times M'.S, \\
\langle s, s' \rangle \mapsto M.S_0\ s \wedge M'.S_0\ s', \\
\langle s_0, s'_0 \rangle \mapsto \langle s_1, s'_1 \rangle \mapsto s_0\ M.T\ s_1 \wedge s'_0\ M'.T\ s'_1, \\
\langle s, s' \rangle \mapsto \langle M.\ell\ s, M'.\ell\ s' \rangle, \\
\ldots
\end{array} \right\rangle
$$

11.11.2 For reference: asynchronous product of two temporal models

$$
\begin{array}{rcl}
\equiv \times^{\mathsf{async}}_{-,=} \equiv \in & & (AP \in \mathsf{Set}) \to \big(AP' \in \mathsf{Set}\big) \to \\
& & \quad \mathsf{TModel}\ AP \to \mathsf{TModel}\ AP' \to \\
& & \qquad \mathsf{TModel}\ (AP \times AP')
\end{array}
$$

$$
M \times^{\mathsf{async}}_{AP,AP'} M' \stackrel{def}{=}
\left\langle \begin{array}{l}
M.S \times M'.S, \\
\langle s, s' \rangle \mapsto M.S_0\ s \wedge M'.S_0\ s', \\
\langle s_0, s'_0 \rangle \mapsto \langle s_1, s'_1 \rangle \mapsto \begin{array}{l} (s_0\ M.T\ s_1 \wedge s'_1 = s'_0) \vee \\ (s_1 = s_0 \wedge s'_0\ M'.T\ s'_1) \end{array}, \\
\langle s, s' \rangle \mapsto \langle M.\ell\ s, M'.\ell\ s' \rangle, \\
\ldots
\end{array} \right\rangle
$$

11.11.3 For reference: squashed (synchronous) product of two temporal models

$$
\begin{array}{rcl}
= \times_{-} \equiv & \in & (AP \in \mathsf{Set}) \to \\
& & \quad \mathsf{TModel}\ AP \to \mathsf{TModel}\ AP \to \mathsf{TModel}\ AP \\
M \times_{AP} M' & \stackrel{def}{=} & \left\langle \begin{array}{l}
M.S \times M'.S, \\
\langle s, s' \rangle \mapsto M.S_0\ s \wedge M'.S_0\ s', \\
\langle s_0, s_0' \rangle \mapsto \langle s_1, s_1' \rangle \mapsto s_0\ M.T\ s_1 \wedge s_0'\ M'.T\ s_1', \\
\langle s, s' \rangle \mapsto M.\ell\ s \wedge M'.\ell\ s', \\
\ldots
\end{array} \right\rangle
\end{array}
$$

11.11.4 Trimming

$$
\begin{array}{l}
\mathsf{trim}\ \ (AP \in \mathsf{Set}) \to \mathsf{TModel}\ AP \to \mathsf{TModel}\ AP \\
\mathsf{trim}\ AP\ M \stackrel{def}{=} \\
\quad \left\langle \begin{array}{l}
\mathbb{1} \to ((s \in S) \times \|\mathsf{Reachable}\ AP\ M\ s\|), \\
s \mapsto M.S_0\ (s\ \star), \\
s_0 \mapsto s_1 \mapsto (s_0\ \star)\ M.T\ (s_1\ \star), \\
s \mapsto p \mapsto M.\ell\ (s\ \star)\ p, \\
\ldots
\end{array} \right\rangle
\end{array}
$$

$$
\begin{array}{l}
\forall AP \in \mathsf{Set}, M \in \mathsf{TModel}\ AP, \psi \in \mathsf{StateProp}^{\mathsf{WI}}. \\
\quad M \vDash_{AP} \psi \leftrightarrow ((\mathsf{trim}\ AP\ M) \vDash_{AP} \psi)
\end{array}
$$

11.11.5 Model checking hybrid systems

Modelling physical systems is often best done with continuous variables. It is possible to extend model checking to capture this. This has been done for example for ACAS X, the Next-Generation Airborne Collision Avoidance System [**?**].

11.11.6 Model checking in unexpected places

Smith Institute: model-checking for radio spectrum auctions!

Course summary

- How temporal models can be used to describe systems that evolve in time.
- How temporal logics can be used to specify those systems, and in particular CTL^*, CTL, LTL.
- How to write temporal models.
- How to relate temporal models with simulation.
- How to implement model-checking for CTL, and counterexample generation for ACTL.

A The meta-language

The meta-level is a univalent Martin-Löf-style type theory with a hierarchy of universes, Set_n for all $n \in \mathbb{N}$. We write Set to mean Set_n for the smallest n that works in that context. Prop stands either for an impredicative universe, or just for the right Set_n in the context.

$(x \in T_1) \to T_2$ is the dependent function type, where x, of type T_1, is bound in T_2; its constructor is $x \mapsto e$, the function mapping x to e (in which x is bound).

$(x \in T_1) \times T_2$ is the dependent pair type, where x, of type T_1, is bound in T_2; its constructor is $\langle e_1, e_2 \rangle$, the pair with first component e_1, and second component e_2. We write the projection of component c of t as $t.c$.

$\forall x \in A.\, B$ is syntactic sugar for $\|(x \in A) \to B\|$.
$\exists x \in A.\, B$ is syntactic sugar for $\|(x \in A) \times B\|$.

Proof terms and annotations for dependent pattern matching are omitted.

We conflate pairs where the second component is a squash type with the first component when not ambiguous.

We write † for impossible cases, when matching on an empty type.

A.1 Basic types

$$\mathsf{data}\ \mathbb{0} \in \mathsf{Set}\ \mathsf{where}$$

(there are no constructors)

$$\mathsf{data}\ \bot \in \mathsf{Prop}\ \mathsf{where}$$

(there are no constructors)

$$\begin{aligned}&\mathsf{data}\ \mathbb{1} \in \mathsf{Set}\ \mathsf{where}\\ &\quad * \in \mathbb{1}\end{aligned}$$

$$\begin{aligned}&\mathsf{data}\ \top \in \mathsf{Prop}\ \mathsf{where}\\ &\quad I \in \top\end{aligned}$$

$$\begin{array}{l}\mathsf{data}\ \mathbb{B} \in \mathsf{Set}\ \mathsf{where}\\ \quad \top_{\mathbb{B}} \in \mathbb{B}\\ \quad \bot_{\mathbb{B}} \in \mathbb{B}\end{array}$$

$$\begin{array}{l}\mathsf{data}\ \mathsf{option}\ (A \in \mathsf{Set}) \in \mathsf{Set}\ \mathsf{where}\\ \quad \mathsf{none} \in \mathsf{option}\ A\\ \quad \mathsf{some} \in A \to \mathsf{option}\ A\end{array}$$

$$\begin{array}{l}\mathsf{data}\ ②^{①*}((A \in \mathsf{Set})) \in (A \to A \to \mathsf{Prop}) \to (A \to A \to \mathsf{Prop})\ \mathsf{where}\\ \quad \mathsf{refl} \in (R \in A \to A \to \mathsf{Prop}) \to x \in A \to R^{A*}\ x\ x\\ \quad \mathsf{trans} \in (R \in A \to A \to \mathsf{Prop}) \to x, y, z \in A \to R^{A*}\ x\ y \to R^{A*}\ y\ z \to R^{A*}\ x\ z\\ \quad \mathsf{inj} \in (R \in A \to A \to \mathsf{Prop}) \to x, y \in A \to R\ x\ y \to R^{A*}\ x\ y\end{array}$$

We elide the A argument.

A.2 Equality

$$\begin{array}{l}\mathsf{data}\ ②\ =_{①}\ ③\ (A \in \mathsf{Set}) \in A \to A \to \mathsf{Prop}\ \mathsf{where}\\ \quad \mathsf{refl} \in (a \in A) \to a =_A a\end{array}$$

We write $=$ instead of $=_A$ when not ambiguous.

A.3 Reflection

$$\begin{array}{l}\mathsf{reflect} \in \mathbb{B} \to \mathsf{Prop} \to \mathsf{Prop}\\ \mathsf{reflect}\ b\ P \stackrel{def}{=} (b = \top_{\mathbb{B}}) \leftrightarrow P\end{array}$$

$$\begin{array}{l}\mathsf{eqType} \in \mathsf{Set} \to \mathsf{Set}\\ \mathsf{eqType}\ A \stackrel{def}{=}\ (eqb \in A \to A \to \mathbb{B}) \times\\ \qquad\qquad \mathsf{reflect}\ eqb\ (=_A)\end{array}$$

A.4 Sub

$$\begin{aligned}&\mathsf{sub} \in \mathsf{Set} \to \mathsf{Set}\\ &\mathsf{sub}\ A \stackrel{def}{=} A \to \mathsf{Prop}\end{aligned}$$

$\emptyset$ is syntactic sugar for $a \mapsto \bot$, or $a \mapsto b \mapsto \bot$, for sub and relation, respectively

$\{t\}$ is syntactic sugar for $x \mapsto x = t$

$\{x \mid P\}$ is syntactic sugar for $x \mapsto P$

$\{x, y \mid P\}$ is syntactic sugar for $x \mapsto y \mapsto P$

A.5 Squash type

$$\begin{aligned}&\mathsf{data}\ \|①\| \in \mathsf{Set} \to \mathsf{Set}\ \mathsf{where}\\ &\quad \mathsf{inj} \in (A \in \mathsf{Set}) \to \|A\|\\ &\quad \mathsf{confl} \in (A \in \mathsf{Set}) \to (x \in \|A\|) \to (y \in \|A\|) \to x = y\end{aligned}$$

We are a bit generous with using elements of squash types.

A.6 Lists

$$\begin{aligned}&\mathsf{data}\ \mathsf{list}\ (A \in \mathsf{Set}) \in \mathsf{Set}\ \mathsf{where}\\ &\quad [] \in \mathsf{list}\ A\\ &\quad ① :: ② \in A \to \mathsf{list}\ A \to \mathsf{list}\ A\end{aligned}$$

$$\begin{aligned}&\mathsf{nth} \in (A \in \mathsf{Set}) \to \mathsf{list}\ A \to \mathbb{N} \to \mathsf{option}\ A\\ &\mathsf{nth}\ A\ []\ n \stackrel{def}{=} \mathsf{none}\\ &\mathsf{nth}\ A\ (x :: l)\ 0 \stackrel{def}{=} \mathsf{some}\ x\\ &\mathsf{nth}\ A\ (x :: l)\ (n+1) \stackrel{def}{=} \mathsf{nth}\ A\ l\ n\end{aligned}$$

$$
\begin{array}{ll}
\mathsf{forall\text{-}list} & \in (A \in \mathsf{Set}) \to \mathsf{list}\ A \to (A \to \mathbb{B}) \to \mathbb{B} \\
\mathsf{forall\text{-}list}\ A\ []\ f & \stackrel{def}{=} \top_{\mathbb{B}} \\
\mathsf{forall\text{-}list}\ A\ (x :: xs)\ f & \stackrel{def}{=} (f\ x) \wedge_{\mathbb{B}} \mathsf{forall\text{-}list}\ A\ xs\ f
\end{array}
$$

$$
\begin{array}{ll}
\mathsf{existsi} & \in (A \in \mathsf{Set}) \to \mathsf{list}\ A \to (\mathbb{N} \to A \to \mathbb{B}) \to \mathbb{B} \\
\mathsf{existsi}\ A\ xs\ f & \stackrel{def}{=} \mathsf{aux}\ A\ xs\ 0\ f \\
\mathsf{aux} & \in (A \in \mathsf{Set}) \to \mathsf{list}\ A \to (\mathbb{N} \to A \to \mathbb{B}) \to \mathbb{N} \to \mathbb{B} \\
\mathsf{aux}\ A\ []\ i\ f & \stackrel{def}{=} \bot_{\mathbb{B}} \\
\mathsf{aux}\ A\ (x :: xs)\ i\ f & \stackrel{def}{=} (f\ i\ x) \vee_{\mathbb{B}} \mathsf{aux}\ A\ xs\ (i+1)\ f
\end{array}
$$

...

A.7 Streams

$$
\begin{array}{l}
\mathsf{stream} \in \mathsf{Set} \to \mathsf{Set} \\
\pi, \ldots \in \mathsf{stream}\ A \stackrel{def}{=} \mathbb{N} \to A
\end{array}
$$

$$
\begin{array}{l}
\mathsf{tailn} \in (A \in \mathsf{Set}) \to \mathbb{N} \to \mathsf{stream}\ A \to \mathsf{stream}\ A \\
\mathsf{tailn}\ A\ n\ \pi \stackrel{def}{=} i \mapsto \pi\ (i+n)
\end{array}
$$

A.8 Finite types

$$
\begin{array}{l}
\mathsf{finType} \in (A \in \mathsf{Set}) \to \mathsf{Set} \\
\mathsf{finType}\ A \stackrel{def}{=}\ (eqt \in \mathsf{eqType}\ A) \times \\
\qquad\qquad (l \in \mathsf{list}\ A) \times \\
\qquad\qquad\quad (f \in ((a \in A) \to ((i \in \mathbb{N}) \times (\mathsf{nth}\ A\ l\ i = \mathsf{some}\ a))))
\end{array}
$$

$$
\begin{array}{l}
\mathsf{data\ finfunon}\ ((A \in \mathsf{Set}))\ ((F \in \mathsf{finType}\ A))\ ((R \in A \to \mathsf{Set})) \in \mathsf{list}\ A \to \mathsf{Set}\ \mathsf{where} \\
\quad \mathsf{ffonil} \in \mathsf{finfunon}\ A\ F\ R\ [] \\
\quad \mathsf{ffoncons} \in (x \in A) \to (l \in R\ x) \to (l \in \mathsf{list}\ A) \to \mathsf{finfunon}\ A\ F\ R\ l \to \mathsf{finfunon}\ A\ F\ R\ (x :: l)
\end{array}
$$

$$
\begin{array}{l}
\mathsf{data\ ordinal} \in \mathbb{N} \to \mathsf{Set}\ \mathsf{where} \\
\quad \mathsf{I}_{①} \in (n \in \mathbb{N}) \to (m \in \mathbb{N}) \to m < n \to \mathsf{ordinal}\ n
\end{array}
$$

B Terminology and notation

these slides	alternatives
model checking	property checking
temporal model	Kripke structure, etc.
G	$\Box$
F	$\Diamond$

References

[1] David A. Basin, Cas Cremers, and Catherine A. Meadows. Model checking security protocols. In *Handbook of Model Checking.*, pages 727–762. 2018.

[2] Edmund M. Clarke, Somesh Jha, Yuan Lu, and Helmut Veith. Tree-like counterexamples in model checking. In *LICS*, pages 19–29, 2002.

[3] Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Aurora Schmidt, Ryan Gardner, Stefan Mitsch, and André Platzer. A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system. *Int. J. Softw. Tools Technol. Transf.*, 19(6):717–741, 2017.

[4] Leslie Lamport. What good is temporal logic? In R. E. A. Mason, editor, *IFIP*, pages 657–668, 1983.