Modal and Temporal Logics

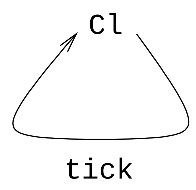
Colin Stirling
School of Informatics
University of Edinburgh

July 24, 2003

Summary

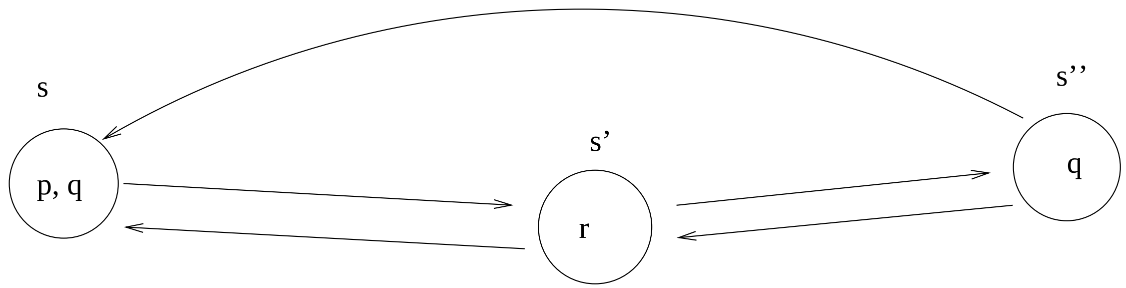
Introduced transition systems

- a set of states S
- a set of labels A
- a set of transitions: $s \xrightarrow{a} s'$ where $s, s' \in S$, $a \in A$



Summary

Or Kripke structure where labels appear at states ("colours")



Summary

Modal logic

$\Phi ::= \texttt{tt} \mid \texttt{ff} \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [K]\Phi \mid \langle K \rangle \Phi$

Semantics $E \models \Phi$

$$
\begin{array}{lll}
E \models \texttt{tt} & & \\
E \not\models \texttt{ff} & & \\
E \models \Phi \wedge \Psi & \text{iff} & E \models \Phi \text{ and } E \models \Psi \\
E \models \Phi \vee \Psi & \text{iff} & E \models \Phi \text{ or } E \models \Psi \\
E \models [K]\Phi & \text{iff} & \forall F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi \\
E \models \langle K \rangle \Phi & \text{iff} & \exists F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi
\end{array}
$$

Summary

Kripke model = Kripke structure + $L : \mathrm{Colours} \rightarrow S$.

Modal Logic: Syntax

$$\Phi ::= p \mid \neg p \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [-]\Phi \mid \langle - \rangle \Phi$$

Summary

Semantics

$$
\begin{array}{lll}
E \models p & \text{iff} & E \in L(p) \\
E \models \neg p & \text{iff} & E \notin L(p) \\
E \models \Phi \wedge \Psi & \text{iff} & E \models \Phi \text{ and } E \models \Psi \\
E \models \Phi \vee \Psi & \text{iff} & E \models \Phi \text{ or } E \models \Psi \\
E \models [-]\Phi & \text{iff} & \forall F \in \{E' : E \longrightarrow E'\}.\ F \models \Phi \\
E \models \langle - \rangle\Phi & \text{iff} & \exists F \in \{E' : E \xrightarrow{a} E'\}.\ F \models \Phi
\end{array}
$$

Summary: Bisimulation on Transition Systems

A binary relation B between states of a transition system is a bisimulation provided that, whenever $(E, F) \in B$ and $a \in A$,

- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $(E', F') \in B$, and
- if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $(E', F') \in B$

Two states E and F are bisimulation equivalent, $E \sim F$, if there is a bisimulation relation B such that $(E, F) \in B$.

Summary: Bisimulation on Kripke Models

A binary relation B between states of a Kripke model is a bisimulation provided that, whenever $(E, F) \in B$

- for all colours p, $E \in L(p)$ iff $F \in L(p)$
- if $E \longrightarrow E'$ then $F \longrightarrow F'$ for some F' such that $(E', F') \in B$, and
- if $F \longrightarrow F'$ then $E \longrightarrow E'$ for some E' such that $(E', F') \in B$

Two states E and F are bisimulation equivalent, $E \sim F$, if there is a bisimulation relation B such that $(E, F) \in B$.

Summary: Invariance

- $E \equiv F$ if for all modal Φ, $E \models \Phi$ iff $F \models \Phi$.

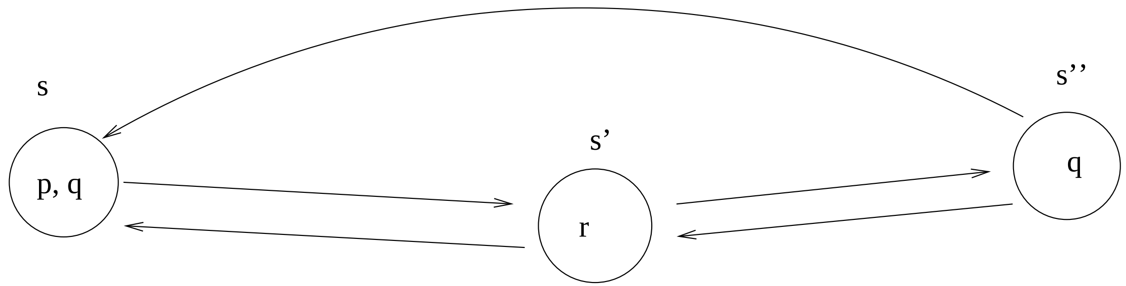
  (E and F have the same modal properties.)

- Proposition if $E \sim F$ then $E \equiv F$

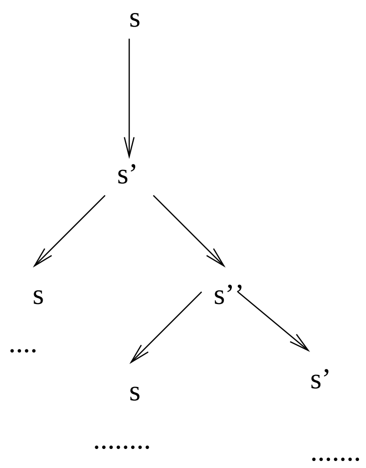
- Proposition if E and F belongs to a finitely branching transition system/Kripke model and $E \equiv F$ then $E \sim F$.

Summary: Unfolding

A transition system/Kripke model can be unfolded into a (bisimulation equivalent) possibly infinite tree



becomes


s
s'
s
s''
....
s
s'
....
....

Computational Properties

- Satisfiability Problem "Given a modal formula Φ, is Φ satisfiable (realisable)?" is **NP**-complete.

- Finite Tree Property If Φ is satisfiable then Φ is satisfiable in a transition system/Kripke model that is a finite tree. (Hence, also **Finite Model Property**: if Φ is satisfiable then Φ is satisfiable in a finite transition system/Kripke model.)

- Model Checking Problem "Given a finite transition system/Kripke model, a state E of it, and a modal formula, does $E \models \Phi$?" is **P**-complete.

Mutual Exclusion: Crucial Properties

- Mutual exclusion
- Absence of deadlock
- Absence of starvation

PROBLEM: None of these properties is expressible in modal logic!

Summary: Runs

- A run from state E_0 is a finite or infinite length sequence of transitions $E_0 \xrightarrow{a_1} E_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} E_n \xrightarrow{a_{n+1}} \ldots$ with "maximal" length.

- Similarly, for a Kripke model.

- A run is a branch in the unfolded transition system/Kripke model.

Beyond Modal Logic

Runs provide a means for expressing long term features .

- **Mutual exclusion**: no run has the property that two components are in their critical section at the same time.
- **Absence of deadlock**: every run has infinite length
- **Absence of starvation**: in every run if a component requests entry into critical section then eventually that component will be in its critical section

Summary: Temporal Operators on Runs

- **Next**: $(K)\Phi$

$$\begin{array}{cccccccc} E_0 & \xrightarrow{a_1} & & E_1 & \xrightarrow{a_2} & \dots & E_i & \xrightarrow{a_{i+1}} \dots \\ & a_1 \in K & & \models & & \dots & & \\ & & & \Phi & & & & \end{array}$$

- **Until**: $\Phi \, \mathrm{U} \, \Psi$. Note: the index i can be 0

$$\begin{array}{cccccccc} E_0 & \xrightarrow{a_1} & E_1 & \xrightarrow{a_2} & \dots & E_i & \xrightarrow{a_{i+1}} \dots \\ \models & & \models & & \dots & \models & \\ \Phi & & \Phi & & & \Psi & \end{array}$$

Summary: Temporal Operators on Runs II

- **Eventually**: $\mathsf{F}\,\Psi = \mathtt{tt}\,\mathrm{U}\,\Psi$. The index i can be 0

$$
\begin{array}{ccccccccc}
E_0 & \xrightarrow{a_1} & E_1 & \xrightarrow{a_2} & \ldots & E_i & \xrightarrow{a_{i+1}} & \ldots \\
 & & & & \ldots & \models & & \\
 & & & & & \Psi & &
\end{array}
$$

- **Always**: $\mathsf{G}\,\Psi = \neg\mathrm{F}\neg\Psi$

$$
\begin{array}{ccccccccc}
E_0 & \xrightarrow{a_1} & E_1 & \xrightarrow{a_2} & \ldots & E_i & \xrightarrow{a_{i+1}} & \ldots \\
\models & & \models & & \ldots & \models & & \\
\Psi & & \Psi & & & \Psi & \ldots &
\end{array}
$$

Summary: Linear Time Temporal Logic (LTL)

Syntax

$$\Phi ::= p \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid (-)\Phi \mid \Phi \,\mathrm{U}\, \Psi$$

Semantics

A state E of a Kripke model satisfies an LTL formula Φ, written $E \models \Phi$,

if for any run π from E, the run $\pi \models \Phi$.

Summary: Invariance

$E \equiv F$ if for all LTL Φ, $E \models \Phi$ iff $F \models \Phi$.

Proposition If $E \sim F$ then $E \equiv F$.

Proof Sketch Bisimulation equivalence preserves runs.

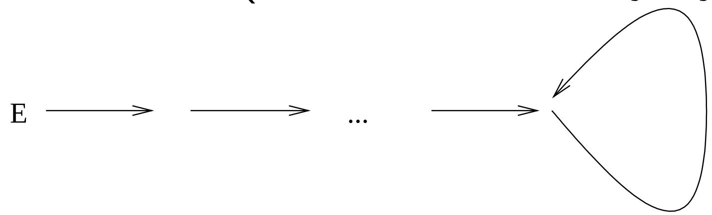
Summary: Computational Properties I

- Satisfiability Problem "Given an LTL formula Φ, is Φ satisfiable (realisable)?" is **PSPACE**-complete.

- "Tree" Model Property If Φ is satisfiable then Φ is satisfiable in a transition system/Kripke model that is a (regular) infinite tree whose branching degree is one.

Summary: Computational Properties II

- Finite Model Property If Φ is satisfiable then Φ is satisfiable in a finite transition system/Kripke model (that is eventually cyclic).



- Model Checking Problem "Given a finite transition system/Kripke model, a state E of it, and an LTL formula, does $E \models \Phi$?" is **PSPACE**-complete.

Branching Time Logics

For each temporal operator such as F, create two variants

- A F "for all runs eventually" (strong)
- E F "for some run eventually" (weak)

Modal operators are also branching time temporal operators

- $[K] = \mathrm{A}\neg(K)\neg$
- $\langle K\rangle = \mathrm{E}\,(K)$

Therefore, we can extend modal logic with branching time temporal operators.

Computation Tree Logic (CTL)

Syntax

$$\Phi ::= \texttt{tt} \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \langle K \rangle \Phi \mid \mathrm{A}(\Phi_1 \,\mathrm{U}\, \Phi_2) \mid \mathrm{E}(\Phi_1 \,\mathrm{U}\, \Phi_2)$$

Semantics

Define when a state of a transition system satisfies a CTL formula.

Semantics of CTL

The new clauses

$$
\begin{array}{lll}
E \models \neg\Phi & \text{iff} & E \not\models \Phi \\[2ex]
E_0 \models \mathrm{A}(\Phi \,\mathrm{U}\, \Psi) & \text{iff} & \text{for all \ runs } E_0 \xrightarrow{a_1} E_1 \xrightarrow{a_2} \ldots \\
 & & \text{there is } i \geq 0 \text{ with } E_i \models \Psi \text{ and} \\
 & & \text{for all } j : 0 \leq j < i,\ E_j \models \Phi \\[2ex]
E_0 \models \mathrm{E}(\Phi \,\mathrm{U}\, \Psi) & \text{iff} & \text{for some \ run } E_0 \xrightarrow{a_1} E_1 \xrightarrow{a_2} \ldots \\
 & & \text{there is } i \geq 0 \text{ with } E_i \models \Psi \text{ and} \\
 & & \text{for all } j : 0 \leq j < i,\ E_j \models \Phi
\end{array}
$$

Derived Operators

$$\mathsf{A}\,\mathsf{F}\,\Phi = \mathsf{A}(\mathtt{tt}\,\mathsf{U}\,\Phi)$$

$$\mathsf{E}\,\mathsf{F}\,\Phi = \mathsf{E}(\mathtt{tt}\,\mathsf{U}\,\Phi)$$

$$\mathsf{A}\,\mathsf{G}\,\Phi = \neg\,\mathsf{E}\,\mathsf{F}\,\neg\,\Phi$$

$$\mathsf{E}\,\mathsf{G}\,\Phi = \neg\,\mathsf{A}\,\mathsf{F}\,\neg\,\Phi$$

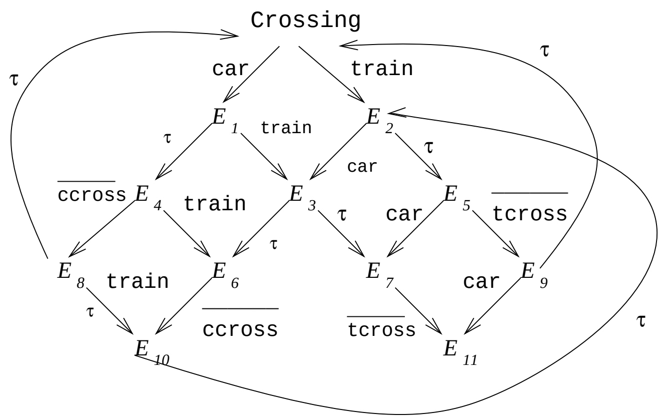
- Safety "nothing bad ever happens": in every run bad is never true. $\mathsf{A}\,\mathsf{G}\,\mathsf{good}$

- Liveness "something good eventually happens": in every run good is eventually true. $\mathsf{A}\,\mathsf{F}\,\mathsf{good}$

- Weak Safety in some run bad is never true. $\mathsf{E}\,\mathsf{G}\,\mathsf{good}$

- Weak Liveness in some run good is eventually true. $\mathsf{E}\,\mathsf{F}\,\mathsf{good}$

Example: Crossing



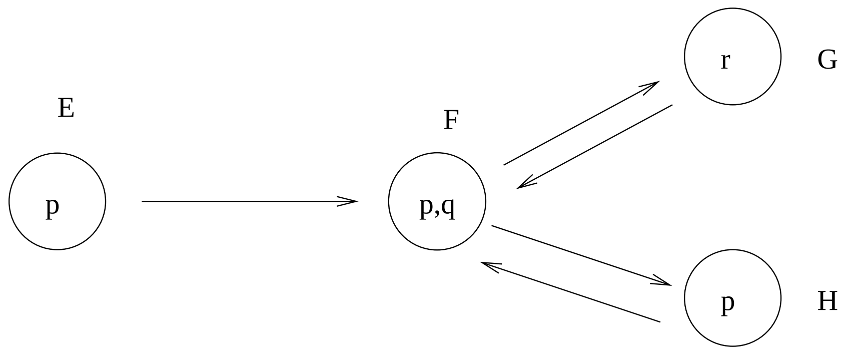
It is never the case that a train and a car can cross at the same time

$$\mathrm{A\,G}([\overline{\texttt{tcross}}]\texttt{ff} \vee [\overline{\texttt{ccross}}]\texttt{ff})$$

Example: Mutual Exclusion

- Mutual exclusion: AG ([exit1]ff $\vee$ [exit2] ff)
- Absence of deadlock: AG $\langle - \rangle$ tt
- Absence of starvation (for one component): AG [req1] AF $\langle$exit1$\rangle$ tt

Exercise



Which of the following are true?

$$E \models \mathrm{A}(p \, \mathrm{U} \, q) \qquad E \models \mathrm{A\,F}\, r$$

$$E \models \mathrm{E\,F}\, r \qquad E \models \mathrm{E\,G}\, p$$

$$E \models \mathrm{A\,F}(\mathrm{E\,G}\, r \vee \mathrm{E\,G}\, p)$$

Exercise: Which are Valid, Unsatisfiable, Neither?

	V	U	N
$\mathrm{A\,G}\,\Phi \rightarrow \mathrm{A\,F}\,\Phi$			
$\mathrm{E\,G}\,\Phi \rightarrow \mathrm{A\,F}\,\Phi$			
$\mathrm{A\,F}(\Phi \vee \Psi) \rightarrow \mathrm{A\,F}\,\Phi \vee \mathrm{A\,F}\,\Psi$			
$\mathrm{A\,G}\,\Phi \rightarrow (\Phi \wedge [-]\mathrm{A\,G}\Phi$			
$\mathrm{A\,G\,A\,F}\Phi \rightarrow \mathrm{E\,F\,E\,G}\Phi$			
$\mathrm{A\,F\,A\,G}\,\Phi \rightarrow \mathrm{A\,G\,A\,F}\,\Phi$			
$\mathrm{A\,G}(\neg\Phi \vee \langle-\rangle\Phi) \wedge \Phi \wedge \mathrm{A\,F}\,\neg\Phi$			
$\mathrm{A\,G}(\Phi \rightarrow \Psi) \rightarrow (\mathrm{A\,G}\,\Phi \rightarrow \mathrm{A\,G}\,\Psi)$			

Exercise

Let $E \equiv F$ if for all CTL Φ, $E \models \Phi$ iff $F \models \Phi$.

Prove the following:

Proposition If $E \sim F$ then $E \equiv F$.

Computational Properties

- Satisfiability Problem "Given an CTL formula Φ, is Φ satisfiable ?" is **EXPTIME**-complete.

- Tree Model Property If Φ is satisfiable then Φ is satisfiable in a transition system/Kripke model that is a (regular) infinite tree.

- Finite Model Property If Φ is satisfiable then Φ is satisfiable in a finite transition system/Kripke model.

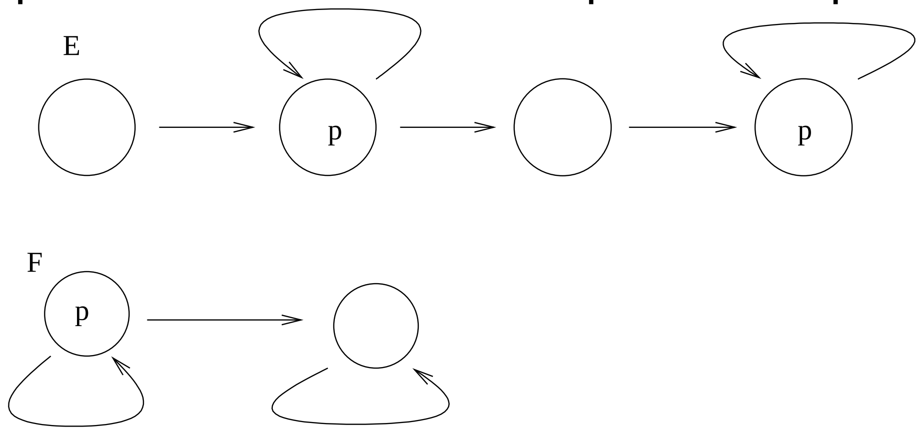
- Model Checking Problem "Given a finite transition system/Kripke model, a state E of it, and a CTL formula, does $E \models \Phi$?" is **P**-complete.

Incomparability of LTL and CTL

A formula of Φ of LTL is equivalent to a formula Ψ of CTL if for every model and state E, $E \models \Phi$ iff $E \models \Psi$.

- CTL and LTL are expressively incomparable .

- E F A G p is not expressible in LTL and F G p is not expressible in CTL



- Open Question: which formulas of LTL are in CTL? (Known: which formulas of CTL are in LTL.)

CTL*

Syntax

Allow the facility to have branching time formulas with arbitrary embeddings of linear time and boolean operators.

$$\Phi ::= p \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid (-)\Phi \mid \Phi \,\mathrm{U}\, \Psi \mid \mathrm{A}\,\Phi$$

Example formula: $\mathsf{A}(\mathsf{F}\,\mathsf{G}\,\Phi \wedge \mathsf{G}\,\mathsf{F}\,\Psi)$

Semantics of CTL*

A state E of a Kripke model satisfies an CTL* formula Φ, written $E \models \Phi$, if for any run π from E, the run $\pi \models \Phi$.

$\pi(0)$ is initial state of π

$$\begin{array}{lll} \pi \models \mathrm{A}\Phi & \text{iff} & \text{for any run } \pi' \text{ if } \pi'(0) = \pi(0) \\ & & \text{then } \pi' \models \Phi \end{array}$$

CTL* contains both LTL and CTL.

Let $E \equiv F$ if for all $\Phi \in \mathrm{CTL}^*$, $E \models \Phi$ iff $F \models \Phi$.

Proposition If $E \sim F$ then $E \equiv F$.

Computational Properties

- Satisfiability Problem "Given a CTL* formula Φ, is Φ satisfiable ?" is **2EXPTIME**-complete.

- Tree Model Property If Φ is satisfiable then Φ is satisfiable in a transition system/Kripke model that is a (regular) infinite tree.

- Finite Model Property If Φ is satisfiable then Φ is satisfiable in a finite transition system/Kripke model.

- Model Checking Problem "Given a finite transition system/Kripke model, a state E of it, and a CTL* formula, does $E \models \Phi$?" is **PSPACE**-complete.

Model Checking CTL Formulas

$\| \Phi \| \;=\; \{E : E \models \Phi\}$ in a fixed model.

Model checking is "bottom up" by computing $\| \Psi \|$ for any subformula of Φ and then computing $\| \Phi \|$.

- $\| \neg\Phi_1 \| = -\| \Phi_1 \|$,
- $\| \Phi_1 \wedge \Phi_2 \| = \| \Phi_1 \| \cap \| \Phi_2 \|$
- $\| \langle K \rangle \Phi \| = \{F \;:\; \exists F' \in \| \Phi \|, a \in K.\, F \xrightarrow{a} F'\}$

Model Checking CTL

- $\|\, \mathrm{E}(\Phi\, \mathrm{U}\, \Psi) \,\| = \bigcup S_i$ where $S_1 = \|\, \Psi \,\|$ and

  $S_{i+1} = S_i \;\cup\; \{F \in \|\, \Phi \,\| \;:\; \exists a, F' \in S_i.\, F \xrightarrow{a} F'\}$

- $\|\, \mathrm{A}(\Phi\, \mathrm{U}\, \Psi) \,\| = \bigcup S_i$ where $S_1 = \|\, \Psi \,\|$ and

  $S_{i+1} = S_i \cup \{F \in \|\, \Phi \,\| : \exists a, F'.\, F \xrightarrow{a} F' \text{ and } \forall a, F'.\ \text{if } F \xrightarrow{a} F' \text{ then } F' \in S_i\}$

Direct backwards reachability computation for E U in transition graph. For A U it is forward reachability which can be implemented efficiently using depth first search.

Both are fixed point computations: essence of model checking

Temporal Operators as Fixed points

1. $\mathsf{E}(\Phi\,\mathsf{U}\,\Psi) \equiv \Psi \lor (\Phi \land \langle - \rangle \mathrm{E}(\Phi\,\mathrm{U}\,\Psi))$

2. $\mathsf{A}(\Phi\,\mathsf{U}\,\Psi) \equiv \Psi \lor (\Phi \land \langle - \rangle \mathtt{tt} \land [-]\mathrm{A}(\Phi\,\mathrm{U}\,\Psi))$

Syntactically: property X such that

1. $X \equiv \Psi \lor (\Phi \land \langle - \rangle X)$

2. $X \equiv \Psi \lor (\Phi \land \langle - \rangle \mathtt{tt} \land [-]X)$

Temporal Operators as Fixed points

Semantically: set of states $S = f(S)$ where f is

1. $\lambda x. \| \Psi \vee (\Phi \wedge \langle - \rangle x) \|$

2. $\lambda x. \| \Psi \vee (\Phi \wedge \langle - \rangle \mathtt{tt} \wedge [-] x) \|$

If $S = f(S)$ then S is a fixed point of f.

In both cases f is monotonic: $S \subseteq S' \rightarrow f(S) \subseteq f(S')$

f is essentially modal (using $\langle - \rangle$ and $[-]$).

Fixed points

S is a prefixed point of f, if $f(S) \subseteq S$

S is a postfixed point of f, if $S \subseteq f(S)$

Proposition If f is monotonic (w.r.t $\subseteq$) then f

1. has a least fixed point, $\bigcap\{S : f(S) \subseteq S\}$
2. has a greatest fixed point, $\bigcup\{S : S \subseteq f(S)\}$

Exercise: Prove this.

Example

- $S = f(S)$ when $f = \lambda x. \| \Psi \vee (\Phi \wedge \langle - \rangle x) \|$
- There can be many different fixed points of f.
- The one wanted that expresses $\mathsf{E}(\Phi\,\mathsf{U}\,\Psi)$ is the **least** fixed point.
- **Exercise:** what does the greatest fixed point of f express?

Example

- $\mathrm{A\,G}\,\Phi \;\equiv\; \Phi \wedge [-]\mathrm{A\,G}\,\Phi$. Computing $\|\,\mathrm{A\,G}\,\Phi\,\|$: Simple depth first search, that stops when reach a state in $\|\,\neg\Phi\,\|$.

- $\|\,\mathrm{A\,G}\,\Phi\,\| = \bigcap S_i$ where $S_1 = \|\,\Phi\,\|$ and

  $S_{i+1} = S_i \cap \{F \in S_i \,:\, \forall a, F'.F \xrightarrow{a} F' \text{ implies } F' \in S_i\}$

- Syntactically: Property X such that $X \;\equiv\; \Phi \wedge [-]X$

  Semantically: fixed point of $f \;=\; \lambda x.\|\,\Phi \wedge [-]x\,\|$

- Required property, $\mathrm{A\,G}\,\Phi$, is **greatest** fixed point of f

- **Exercise:** What does the least fixed point of f express?

Exercise

What property is defined by the following fixed points?

1. $X \equiv \Phi \vee \langle - \rangle X$ least

2. $X \equiv \Phi \wedge \langle - \rangle X$ greatest

3. $X \equiv \Phi \wedge [-][-]X$ greatest

A Scheduler

Problem: assume n tasks when $n > 1$.

a_i initiates the ith task and b_i signals its completion

The scheduler plans the order of task initiation, ensuring

1. actions $a_1 \ldots a_n$ carried out cyclically and tasks may terminate in any order

2. but a task can not be restarted until its previous operation has finished. (a_i and b_i happen alternately for each i.)

More complex temporal properties. Not expressible in CTL* ("not first order" but are "regular").

Expressible using fixed points

Modal Logic+

Z ranges over propositional variables

$\Phi ::= Z \mid \mathtt{tt} \mid \mathtt{ff} \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [K]\Phi \mid \langle K\rangle\Phi$

- $\models$ refined to $\models_{\mathsf{V}}$ where V is a **valuation** that assigns a set of states $\mathsf{V}(X)$ to each variable X

$$E \models_{\mathsf{V}} X \text{ iff } E \in \mathsf{V}(X)$$

- $\parallel \Phi \parallel$ refined too: $\parallel \Phi \parallel_{\mathsf{V}} \; = \{E \,:\, E \models_{\mathsf{V}} \Phi\}$

- $\mathsf{V}[S/X]$ is valuation V' like V except $\mathsf{V}'(X) = S$.

Modal Logic$+$ II

Proposition The function $\lambda x.\| \Phi \|_{\mathsf{V}[x/X]}$ is monotonic for any modal Φ.

- If $\neg$ explicitly in logic then above not true: $\neg X$: $\lambda x. -x$ not monotonic.

  However, define when Φ is **positive** in X: if X occurs within an even number of negations in Φ

  Proposition If Φ is positive in X then $\lambda x.\| \Phi \|_{\mathsf{V}[x/X]}$ is monotonic

1. Property given by **least** fixed point of $\lambda x.\| \Phi \|_{\mathsf{V}[x/X]}$ is written $\mu X.\Phi$.

2. Property given by **greatest** fixed point of $\lambda x.\| \Phi \|_{\mathsf{V}[x/X]}$ is written $\nu X.\Phi$.

Alternative basis for temporal logic: **modal logic + fixed points**