Temporal Logic

Ralf Huuck

Australian Government
Department of Communications, Information Technology and the Arts
Australian Research Council

Outline

- Why not "standard" logic?
- What is temporal logic?
- LTL
- CTL*
- CTL
- Fairness

Model Checking Problem

$$M \overset{?}{\models} \phi$$

model, program	satisfies, Implements, refines	property, specification

How to formalize the different components?

for this lecture we assume is M given as a Kripke structure, what about $\models$ and ϕ ?

Kripke Structure

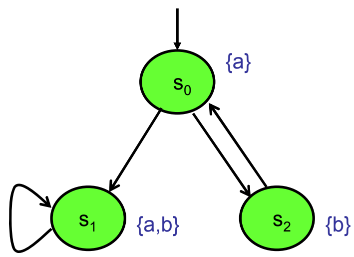
Given a set of atomic propositions AP.

Kripke structure $M=(S, s_0, \rightarrow, \mu)$ is defined by

- S set of states
- s_0 initial state
- $\rightarrow \subseteq S \times S$ transition relation (total)
- $\mu: S \rightarrow 2^{AP}$ labeling function

Any infinite run is accepting, i.e., like Buchi automaton where every state is a final state. Product as for NFA.

Example



How to define properties formally?

- Kripke structure
- automata
- ω regular expression
- logics

Logic can provide succinct notation, "close" to natural language.

Propositional Logic

Syntax

$$\phi ::= p \mid \neg\phi \mid \phi_1 \vee \phi_2$$

Other connectivities ($\wedge$, $\Leftrightarrow$, $\Rightarrow$, …) can be derived (see next slide)

Propositional Logic

Semantics

Given a state s in a Kipke structure M
we define $M,s \models \phi$ inductively by:

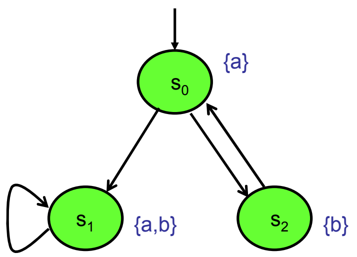
$M,s \models p$	$\Leftrightarrow$	$p \in \mu(s)$
$M,s \models \neg\phi$	$\Leftrightarrow$	not $M,s \models \phi$
$M,s \models \phi_1 \vee \phi_2$	$\Leftrightarrow$	$M,s \models \phi_1$ or $M,s \models \phi_2$

($\wedge$, $\Leftrightarrow$, $\Rightarrow$, true,false…)

Propositional logic is good at describing "static" situations.

Example

- M,s_0 ⊨ a∨b
- M,s_1 ⊨ a
- M,s_2 ⊨ ¬a

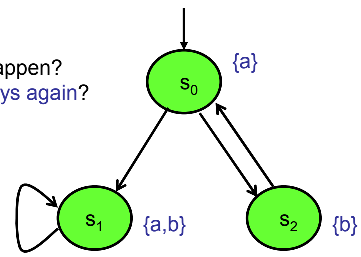


Propositional logic is good for describing “static” situations.

The imagination driving Australia's ICT future.

Example

How to describe:
- eventually b will happen?
- a will happen always again?



Propositional logic is unsuitable for describing “dynamic” behavior.

The imagination driving Australia's ICT future.

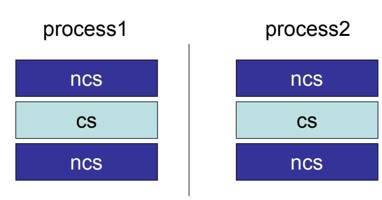
Dynamic Behavior

Important for reactive systems

- security protocols
- hardware
- operating systems
- embedded systems
- ...

The imagination driving Australia's ICT future.

Mutual Exclusion



always only one process in cs
eventually process1 in cs
always eventually process2 in cs

cs = critical section
ncs = none critical section

LTL

The imagination driving Australia's ICT future.
NATIONAL ICT AUSTRALIA

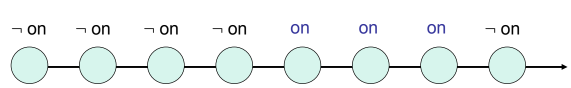
Temporal Logics

- originate from philosophy
- how to express statements including time?
- what is an appropriate model?
  - real-time vs discrete time
  - linear time vs branching time
    (deterministic vs non-deterministic)
  - ...

The imagination driving Australia's ICT future.
NATIONAL ICT AUSTRALIA

(P)LTL

- Propositional Linear time Temporal Logic
- discrete time
- linear (deterministic) progression



The imagination driving Australia's ICT future.
NATIONAL ICT AUSTRALIA

LTL Syntax

- PLTL formula are inductively defined by:

$$\phi ::= p \mid \neg \phi \mid \phi_1 \vee \phi_2 \mid X\phi \mid F\phi \mid G\phi \mid \phi_1 U \phi_2$$

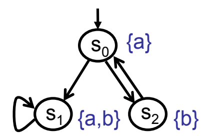
- p denotes atomic proposition
- X denotes next-state operator
- F denotes eventually/finally
- G denotes always/globally
- U denotes until

LTL semantics

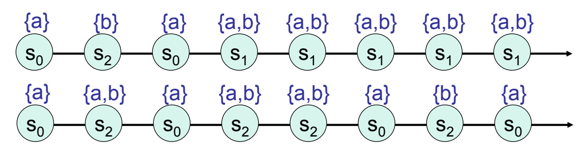
- LTL formula ϕ interpreted over infinite paths of states $\pi = s_0 s_1 s_2 \ldots$
- we define LTL wrt Kripke structure M
- $M,\pi \models \phi$ denotes ϕ holds in a path π of Kripke structure M.
- $M \models \phi$ iff all paths of M satisfy ϕ, i.e., for all π in M we have $M,\pi \models \phi$

The imagination driving Australia's ICT future.
NATIONAL ICT AUSTRALIA

Paths in Kripke Structures



remember transition relation is total



The imagination driving Australia's ICT future.
NATIONAL ICT AUSTRALIA

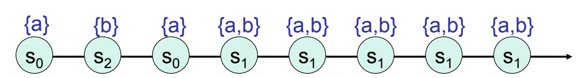
Semantics of LTL Operators

let π^k denote suffix $s_k s_{k+1} s_{k+2} \ldots$ where $k \geq 0$

$M,\pi \models p$ iff $s_0 \models p$, i.e., $p \in \mu(s_0)$
$M,\pi \models \neg\phi$ iff not $M,\pi \models \phi$
$M,\pi \models \phi_1 \vee \phi_2$ iff $M,\pi \models \phi_1$ or $M,\pi \models \phi_2$
$M,\pi \models X\phi$ iff $M,\pi^1 \models \phi$, i.e., $s_k s_{k+1} s_{k+2} \ldots$ satisfies ϕ
$M,\pi \models F\phi$ iff $\exists\, k \geq 0$ s.t. $M,\pi^k \models \phi$
$M,\pi \models G\phi$ iff $\forall k \geq 0$ $M,\pi^k \models \phi$
$M,\pi \models \phi_1 U \phi_2$ iff $\exists \geq 0$ s.t. $M,\pi^k \models \phi_2$ and $\forall 0 \leq j < k$ we have $M,\pi^j \models \phi_1$

The imagination driving Australia's ICT future.
NATIONAL ICT AUSTRALIA

Example



path π satisfies:

- $M,\pi \models a$
- $M,\pi \models \neg b$
- $M,\pi \models Xb$
- $M,\pi \models Fa$
- $M,\pi \models FG(a \wedge b)$
- $M,\pi \models F(bUa)$
- $M,\pi \models FG(aUb)$
- what else?

Exercise

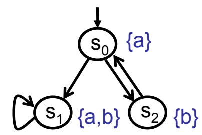
Which temporal operators can be expressed through one or more of the others?

Which cannot?

The imagination driving Australia's ICT future.

$M \models \phi$

$M \models \phi$ iff $M, \pi \models \phi$ for all paths π



Which properties satisfy this Kripke structure?

The imagination driving Australia's ICT future.

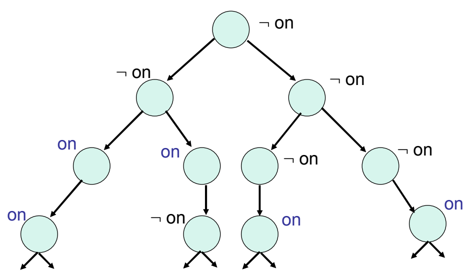
CTL*

The imagination driving Australia's ICT future.

CTL*

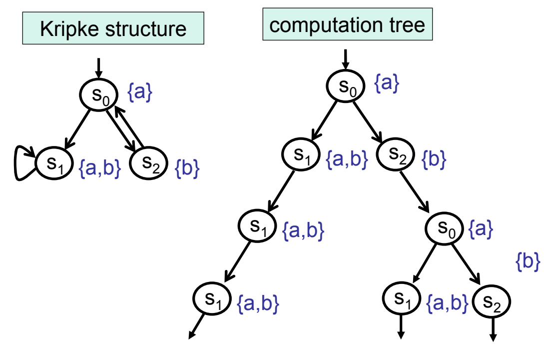
- Computational Tree Logic (star)
- discrete time
- branching (non-deterministic) progression
- describes properties of computation trees
- more powerful than LTL

Computation Trees



The imagination driving Australia's ICT future.

Unwinding Kripke Structure



The imagination driving Australia's ICT future.

Operators in CTL*

- temporal operators
  - same as LTL, describe properties of paths
- path quantifiers
  - A: for all paths (∀) ...
  - E: there exists a path (∃) ...
- more formally ...

The imagination driving Australia's ICT future.

CTL* Formulae

- propositional logic as underlying "static" logic
- two different types of formulae
  - state formula: properties of a state
  - path formulas: property of a path
- all LTL formulae are path formulae
- state formulae ≠ "static" propositional formulae

The imagination driving Australia's ICT future.

State Formulae

$$\phi_s ::= p \mid \neg\phi_s \mid \phi_{s1} \vee \phi_{s2} \mid A\phi_\pi \mid E\phi_\pi$$

- ϕ_s denotes state formula
- ϕ_π path formula
- p atomic proposition
- $A\phi_\pi$ and $E\phi_\pi$ are state formulas
- set of all state formulae = set of all legal CTL* formulae

The imagination driving Australia's ICT future.

Path Quantifiers in State Formulae

- A and E are path quantifiers
- denote universal and existential quantification over paths starting in a certain state
- $A\phi_\pi$ holds in a state s
  iff for all paths starting in s, ϕ_π holds
- $E\phi_\pi$ holds in a state s
  iff there exisits a path starting in s,
  s.t. ϕ_π holds

The imagination driving Australia's ICT future.

Path Formulae

$$\phi_\pi ::= \phi_s \mid \neg\phi_\pi \mid \phi_{\pi1} \vee \phi_{\pi2} \mid X\phi_\pi \mid F\phi_\pi \mid G\phi_\pi \mid \phi_{\pi1} U \phi_{\pi2}$$

- every LTL formula is path formula
- all state formulae are also path formulae
- nesting: $A(GF(A\ a \vee b))$ (example tree?)

The imagination driving Australia's ICT future.

Semantics of CTL*

- define semantics w.r.t. Kripke structure M
- $M,s \models \phi$ denotes
  state formula ϕ holds in a state s of M
- $M,\pi \models \phi$ denotes
  path formula ϕ holds for path π in M
- $\models$ defined inductively, as before

Semantics of CTL* State Formulae

- $M,s \models p$ iff $p \in \mu(s)$
- $M,s \models \neg\phi$ iff not $M,s \models \phi$
- $M,s \models \phi_1 \wedge \phi_2$ iff $M,s \models \phi_1$ and $M,s \models \phi_2$
- $M,s \models A\phi$ iff for all paths π starting in s, $M,\pi \models \phi$
- $M,s \models E\phi$ iff there exists a paths π starting in s, such that $M,\pi \models \phi$

Semantics of CTL* Path Formulae

- $M,\pi \models \phi_s$ iff $s_0 \models \phi_s$
- $M,\pi \models \neg\phi$ iff not $M,\pi \models \phi$
- $M,\pi \models \phi_1 \vee \phi_2$ iff $M,\pi \models \phi_1$ or $M,\pi \models \phi_2$
- $M,\pi \models X\phi$ iff $M,\pi^1 \models \phi$, i.e., $s_k s_{k+1} s_{k+2} \ldots$ satisfies ϕ
- $M,\pi \models F\phi$ iff $\exists\, k \leq 0$ s.t. $M,\pi^k \models \phi$
- $M,\pi \models G\phi$ iff $\forall k \leq 0$ $M,\pi^k \models \phi$
- $M,\pi \models \phi_1 U \phi_2$ iff $\exists k \leq 0$ s.t. $M,\pi^k \models \phi_2$ and $\forall 0 \leq j < k$ we have $M,\pi^j \models \phi_1$

basically, same as before

LTL vs. CTL*

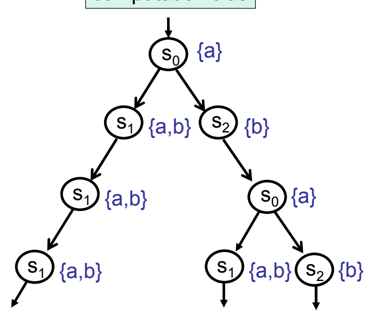
$\phi \in$ LTL implies $A\phi \in$ CTL*

$EF\phi \in$ CTL* but not expressible in LTL (other examples?)

LTL strictly less expressive than CTL*

Examples

computation tree

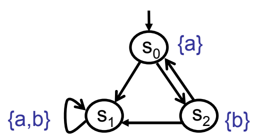


root state s_o satisfies:

- $M,s_0 \models Aa$
- $M,s_0 \models A(\neg b)$
- $M,s_0 \models AXb$
- $M,s_0 \models EFa$
- $M,s_0 \models EFAa$
- what else?

$M \models \phi$

$M \models \phi$ ($\phi \in$ CTL*) iff $M,s_0 \models \phi$ for **initial** state s_0



Which properties satisfies this Kripke structure?

The imagination driving Australia's ICT future.

CTL

a fragment of CTL*

The imagination driving Australia's ICT future.

CTL

- CTL is restricted version of CTL*
- temporal operators as in CTL*
- path quantifier as in CTL*
- However, no unrestricted nesting and Boolean combinations of path formulae
  (next slide)

The imagination driving Australia's ICT future.

CTL Syntax

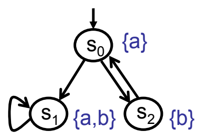
$$\phi ::= p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid AX\phi \mid EX\phi \mid AF\phi \mid EF\phi \mid AG\phi \mid EG\phi \mid A(\phi_1 U \phi_2) \mid E(\phi_1\ U\ \phi 2)$$

- no arbitrary nesting
- path qualifiers and temporal operators alternate
- no Boolean combinations of path formulae

Not allowed: $a \wedge Fq$, $E(A(F(a \vee b))$, ..., what else?

Example

Which CTL properties hold for this Kripke structure?



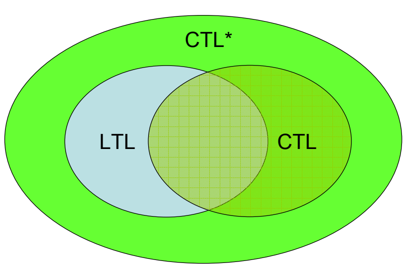
The imagination driving Australia's ICT future.

LTL vs. CTL vs. CTL*

- LTL sublogic of CTL*
  - (EF $\phi \in$ CTL*, not expressible in LTL)
- CTL sublogic of CTL*
  - (FG$\phi \in$ CTL*, not expressible in CTL)
- LTL and CTL not comparable
  - FG$\phi \in$ LTL, not expressible in CTL
  - EF$\phi \in$ CTL, not expressible in LTL

The imagination driving Australia's ICT future.

LTL vs. CTL vs. CTL*



The imagination driving Australia's ICT future.

Safety and Liveness

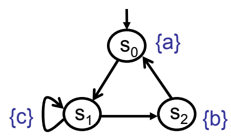
safety: "never something bad will happen"
AG $\neg$(in1 $\wedge$ in2)

liveness: "eventually something good will happen"
EF safe

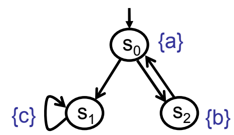
rule of thumb: liveness properties iff counter example requires an infinite trace/infinite deep tree

Fairness

Often liveness properties cannot be proven without certain assumptions, i.e., **fairness**.



M⊨Fb ?



M⊨Fc ?

Weak/Strong Fairness

weak fairness

if an event is continuously enabled, it will occur infinitely often

- in LTL: GF (¬enabled ∨ occurs)

strong fairness

if a event is infinitely often enabled it will occur infinitely often

- in LTL: GF enabled ⇒ GF occurs

Weak/Strong Fairness

weak fairness

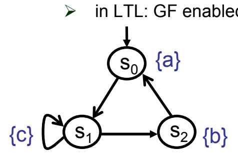
if an event is continuously enabled, it will occur infinitely often

- in LTL: GF (¬enabled ∨ occurs)

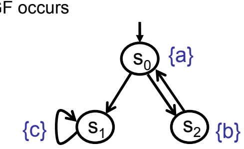
strong fairness

if a event is infinitely often enabled it will occur infinitely often

- in LTL: GF enabled ⇒ GF occurs



under weak fairness: M⊨Fb



under strong fairness: M⊨Fc

Fairness and Model Checking

Weak/strong fairness can be expressed in LTL, however, not in CTL

in **LTL model checking** fairness can be added directly as an assumption

in **CTL model checking** fairness has to be build into the model checking algorithm

Summary

This Lecture

- temporal logic to specify behavior over time
- LTL: linear structure (for all paths)
- CTL(*): branching structure (selective paths)
- LTL, CTL sublogics of CTL*
- CTL, LTL not comparable
- different classes of properties (safety/liveness, fairness)

Next Lecture

- CTL model checking
- how does it work