7. Computational Tree Logic

Computer-Aided Verification

Dave Parker

University of Birmingham

2017/18

Reminder

- Tutorials this week (Assignment 1 feedback)
  - Thur 4pm (surnames A–L, by default):
    - UG06, Murray Learning Centre
  - Fri 10am (surnames M–Z, by default):
    - Lecture Theatre 1, Sports and Exercise Sciences

Recap

- Linear Temporal Logic (LTL)
  - $\bigcirc$ (next), U (until), $\Diamond$ (eventually), $\Box$ (always)

- Examples, common patterns
  - $\Box a$, $\Box(a \rightarrow \Diamond b)$, $\Box(a \rightarrow \bigcirc b)$, $\Box\Diamond a$, $\Diamond\Box a$
  - invariants, safety properties, liveness properties

- Semantics
  - LTL evaluated over infinite paths/traces (and LTSs)
  - $M \vDash \psi \;\Leftrightarrow\;$ <u>all</u> paths of LTL M satisfy LTL formula ψ

- Equivalence of LTL formulas: $\psi_1 \equiv \psi_2$
  - proof using common simpler equivalences/dualities

Overview

- Linear temporal logic (LTL)
  - (non)equivalences, negation

- Computation tree logic (CTL)
  - syntax, semantics
  - examples
  - CTL vs. LTL

- See [BK08] sections 5.1.4 and 6–6.3

Example 1

- Prove (or disprove):

$$\Diamond\psi \ \equiv\ \psi \vee \bigcirc\Diamond\psi \quad ?$$

Yes

- Can prove directly, using the relevant semantics for LTL:
- For any trace $\sigma \in (2^{AP})^{\omega}$ …

$$
\begin{aligned}
\sigma \vDash \Diamond\psi \quad &\Leftrightarrow \quad \exists k \geq 0 \text{ s.t. } \sigma[k\ldots] \vDash \psi \\
&\Leftrightarrow \quad \sigma[0\ldots] \vDash \psi \text{ or } \exists k \geq 1 \text{ s.t. } \sigma[k\ldots] \vDash \psi \\
&\Leftrightarrow \quad \sigma \vDash \psi \text{ or } \exists k \geq 1 \text{ s.t. } \sigma[k\ldots] \vDash \psi \\
&\Leftrightarrow \quad \sigma \vDash \psi \text{ or } \exists j \geq 0\ \sigma[1\ldots][j\ldots] \vDash \psi \\
&\Leftrightarrow \quad \sigma \vDash \psi \text{ or } \sigma[1\ldots] \vDash \Diamond\psi \\
&\Leftrightarrow \quad \sigma \vDash \psi \text{ or } \sigma \vDash \bigcirc\Diamond\psi \\
&\Leftrightarrow \quad \sigma \vDash \psi \vee \bigcirc\Diamond\psi
\end{aligned}
$$

Example 2

- Prove (or disprove):

$$\neg(\Box a \rightarrow \Diamond b) \equiv \Box a \wedge \Box\neg b \quad ? \quad \text{Yes}$$

- Can prove by reusing simpler known equivalences

$$\begin{aligned}
\neg(\Box a \rightarrow \Diamond b) &\equiv \neg(\neg\Box a \vee \Diamond b) && \text{since } \psi_1 \rightarrow \psi_2 \equiv \neg\psi_1 \vee \psi_2 \\
&\equiv \neg\neg\Box a \wedge \neg\Diamond b && \text{since } \neg(\psi_1 \vee \psi_2) \equiv \neg\psi_1 \wedge \neg\psi_2 \\
&\equiv \Box a \wedge \neg\Diamond b && \text{since } \neg\neg\psi \equiv \psi \\
&\equiv \Box a \wedge \neg\Diamond\neg\neg b && \text{since } \psi \equiv \neg\neg\psi \\
&\equiv \Box a \wedge \Box\neg b && \text{since } \neg\Diamond\neg\psi \equiv \Box\psi
\end{aligned}$$

Example 3

- Prove (or disprove):

$$\Box\Diamond a \wedge \Box\Diamond b \equiv \Box\Diamond(a \wedge b) \quad ? \quad \text{No}$$

- Just need to provide a single trace as a counterexample
  - e.g. {a} {b} {a} {b} …
  - (which is satisfied by the left formula only)

LTL & Negation

- Are these statements equivalent? (for trace σ and LTL formula ψ)
  - $\sigma \models \neg\psi$
  - $\sigma \not\models \psi$
- Yes
  - in fact, this is just the semantics of LTL

- Are these statements equivalent? (for LTS M and LTL formula ψ)
  - $M \models \neg\psi$
  - $M \not\models \psi$
- No:
  - $M \models \neg\psi$ means no trace satisfies ψ
  - $M \not\models \psi$ means it is not true that all traces satisfy ψ
    - i.e. there exists some trace that does not satisfy ψ

Existential properties

- Can we verify this, using LTL?
  - "there exists an execution that reaches program location l_2"
- Yes: $M \nvDash \Box\neg l_2$

- Can we verify this, using LTL?
  - "there exists an execution that visits l_2 infinitely often, and never passes through program location l_4"
- Yes: $M \nvDash \neg((\Box\Diamond l_2) \wedge (\Box\neg l_4))$

- Can we verify this, using LTL?
  - "for every execution, it is always possible to return to the initial state of the program"
- No…

CTL

- CTL – Computation Tree Logic
  - branching notion of time (compared to linear time for LTL)
  - infinite trees of states, not infinite sequences of states
- Two path quantifiers: ∀ (for all paths), ∃ (there exists a path)
  - LTL implicitly uses ∀
- Example
  - $\exists\Diamond l_2$ – "does there exist an execution that reaches l_2?"
- CTL model checking
  - quite different to (and simpler than) LTL model checking

CTL syntax

- Syntax split into state and path formulas
  - specify properties of states/paths, respectively
  - a CTL formula is a state formula ϕ
- State formulae:
  - $\phi \ ::= \ \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid \forall\, \psi \mid \exists\, \psi$
  - where $a \in AP$ and ψ is a path formula
- Path formulae
  - $\psi \ ::= \ \bigcirc\, \phi \mid \phi \,\mathrm{U}\, \phi \mid \Diamond\, \phi \mid \Box\, \phi$
  - where ϕ is a state formula
- Examples (note the pairing of quantifiers/temporal operators)
  - $\exists\Diamond l_2$, $\forall\bigcirc b$, $\forall\Box\, \exists\Diamond$ initial

CTL – Alternative styles

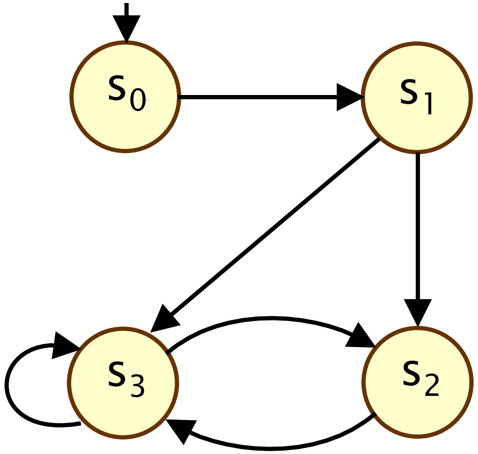
- Temporal operators:
  - $\bigcirc$ a $\equiv$ X a ("next")
  - $\Diamond$ a $\equiv$ F a ("future", "finally")
  - $\Box$ a $\equiv$ G a ("globally")

- Path quantifiers:
  - $\forall\ \psi \equiv A\ \psi$
  - $\exists\ \psi \equiv E\ \psi$

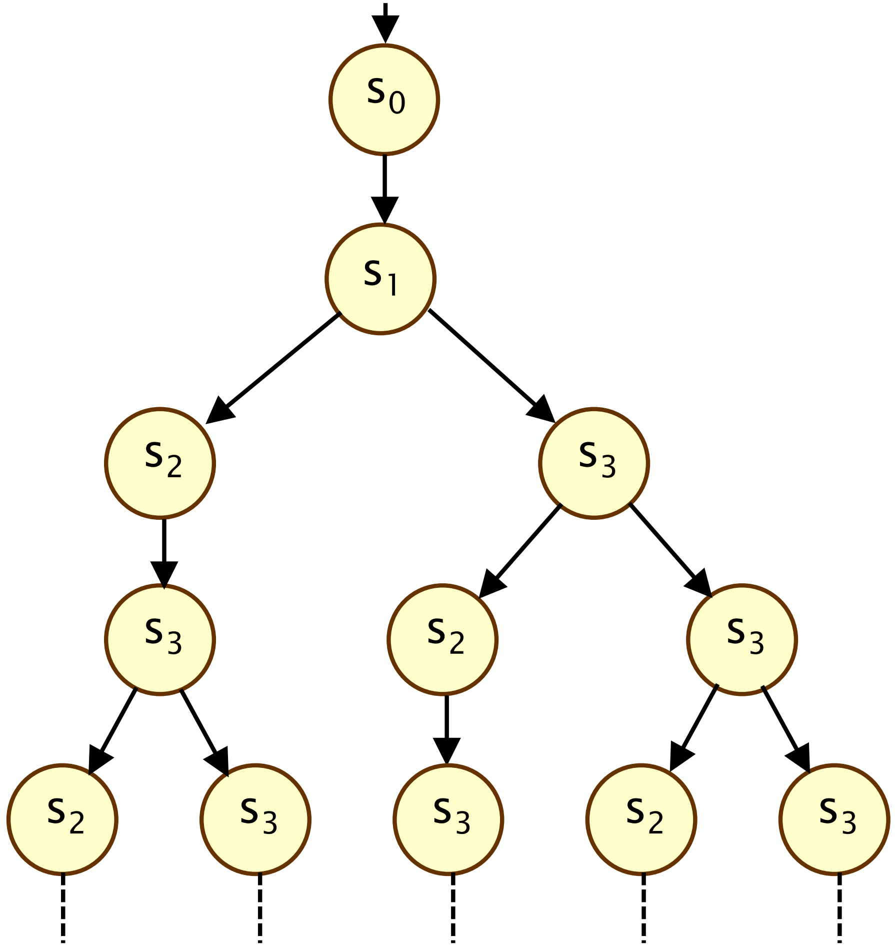
- Brackets for quantifier scope: none/round/square
  - $\forall \Diamond\ \psi$
  - $\forall\ (\ \psi_1\ U\ \psi_2\ )$
  - $\forall\ [\ \psi_1\ U\ \psi_2\ ]$

Computation trees

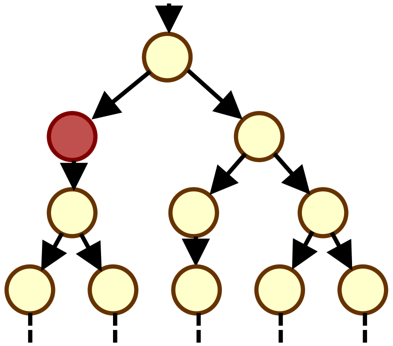
- LTS:



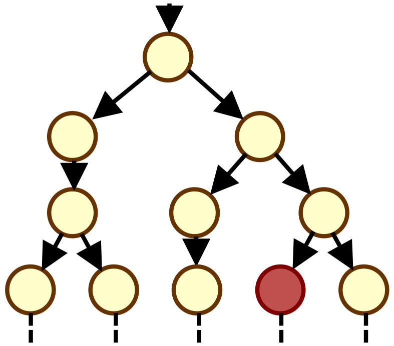
- (Prefix of) infinite computation tree
  - i.e. “unrolling of the LTS”



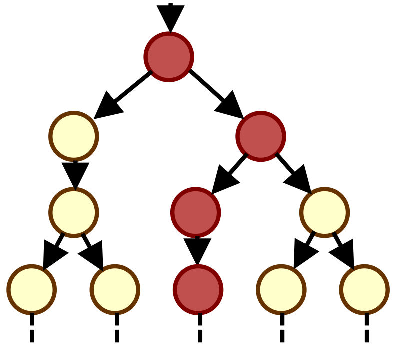
CTL – Intuitive semantics



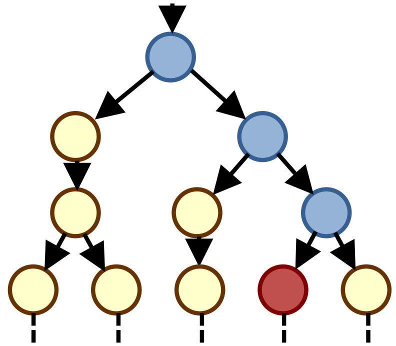
$\exists \bigcirc$ red



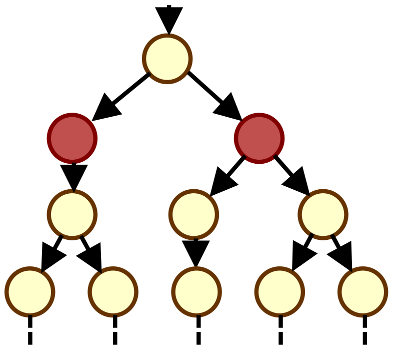
$\exists \Diamond$ red



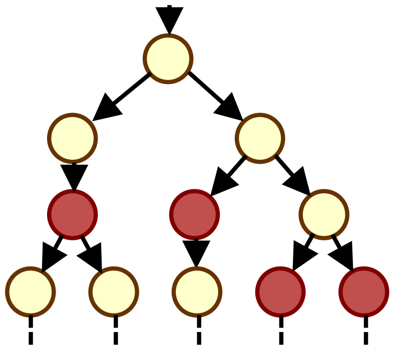
$\exists \Box$ red



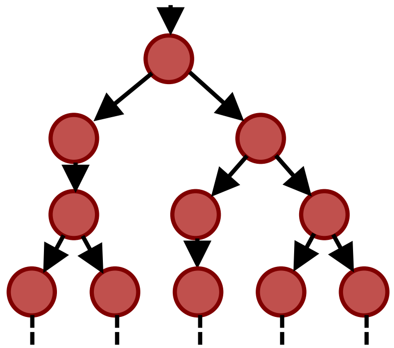
$\exists$ [blue U red]



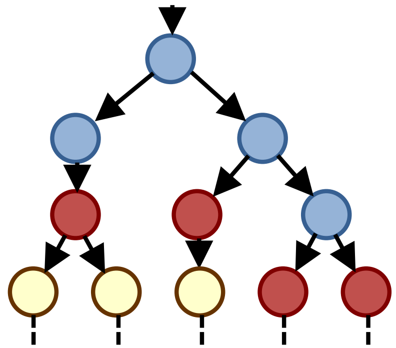
$\forall \bigcirc$ red



$\forall \Diamond$ red



$\forall \Box$ red



$\forall$ [blue U red]

CTL examples

- $\forall\Box\,(\neg(crit_1 \wedge crit_2))$
  - mutual exclusion
- $\forall\Box\,\exists\Diamond$ initial
  - for every computation, it is always possible to return to the initial state
- $\forall\Box\,(request \rightarrow \forall\Diamond\, response)$
  - every request will eventually be granted
- $\forall\Box\,\forall\Diamond\, crit_1 \wedge \forall\Box\,\forall\Diamond\, crit_2$
  - each process has access to the critical section infinitely often

CTL semantics

- Semantics of state formulae:
  - $s \vDash \phi$ denotes "s satisfies ϕ" or "ϕ is true in s"
- For a state s of an LTS $(S, Act, \rightarrow, I, AP, L)$:
  - $s \vDash \text{true}$ always
  - $s \vDash a \;\Leftrightarrow\; a \in L(s)$
  - $s \vDash \phi_1 \wedge \phi_2 \;\Leftrightarrow\; s \vDash \phi_1$ and $s \vDash \phi_2$
  - $s \vDash \neg\phi \;\Leftrightarrow\; s \nvDash \phi$
  - $s \vDash \forall\, \psi \;\Leftrightarrow\; \pi \vDash \psi$ for all $\pi \in Path(s)$
  - $s \vDash \exists\, \psi \;\Leftrightarrow\; \pi \vDash \psi$ for some $\pi \in Path(s)$

(i+1)th state of path π

- and for a path π:
  - $\pi \vDash \bigcirc\, \phi \;\Leftrightarrow\; \pi[1] \vDash \phi$
  - $\pi \vDash \phi_1 \,U\, \phi_2 \;\Leftrightarrow\; \exists k \geq 0$ s.t. $\pi[k] \vDash \phi_2$ and $\forall i < k\; \pi[i] \vDash \phi_1$

Examples

- $s_0 \models \forall\bigcirc\, b$?
- $s_0 \models \exists\bigcirc\, \neg b$?
- $s_0 \models \exists(a \mathbin{U} a \wedge b)$?
- $s_0 \models \exists\bigcirc\, \forall\Box\, (a \wedge b)$?

M: