7. CTL and LTL

Computer-Aided Verification

Dave Parker

University of Birmingham

2016/17

Lectures/tutorials

- Lectures:
  - Tue 2-3pm:
    - Haworth Building 203
  - Thur 2-3pm:
    - Wks 1-3: Poynting Building Small Lecture Theatre (S06)
    - Wk 4: BioSciences NG08
    - Wks 5-11: Gisbert Kapp Building Lecture Theatre 1 (E203)
- Tutorials – first session this week; attend one:
  - Thur 4-5pm (surnames A-K, by default):
    - BioSciences NG08
  - Fri 10-11am (surnames L-Z, by default):
    - Murray Learning Centre UG09
  - solutions/discussion/questions on Assm 1

Overview

- Computation tree logic (CTL)
  - syntax, semantics
  - examples, expressiveness
- CTL vs. LTL
  - expressiveness
  - key differences
  - CTL*
  - fairness
- See [BK08] sections 6-6.3

CTL syntax

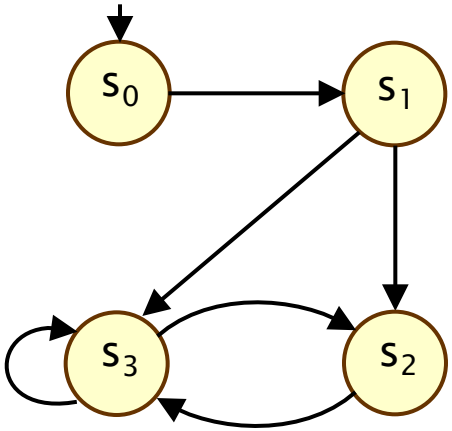
- Syntax split into state and path formulae
  - specify properties of states/paths, respectively
  - a CTL formula is a state formula ϕ
- State formulae:
  - ϕ ::= true | a | $\phi \wedge \phi$ | $\neg\phi$ | $\forall\ \psi$ | $\exists\ \psi$
  - where a $\in$ AP and ψ is a path formula
- Path formulae
  - ψ ::= $\bigcirc\ \phi$ | ϕ U ϕ | $\Diamond\ \phi$ | $\Box\ \phi$
  - where ϕ is a state formula
- Example:
  - $\forall\Box\ \exists\Diamond$ initial

CTL – Alternative styles

- Temporal operators:
  - ◯ a ≡ X a ("next")
  - ◇ a ≡ F a ("future", "finally")
  - □ a ≡ G a ("globally")
- Path quantifiers:
  - ∀ ψ ≡ A ψ
  - ∃ ψ ≡ E ψ
- Brackets: none/round/square
  - A□ ϕ
  - A (ϕ_1 U ϕ_2)
  - A [ϕ_1 U ϕ_2]

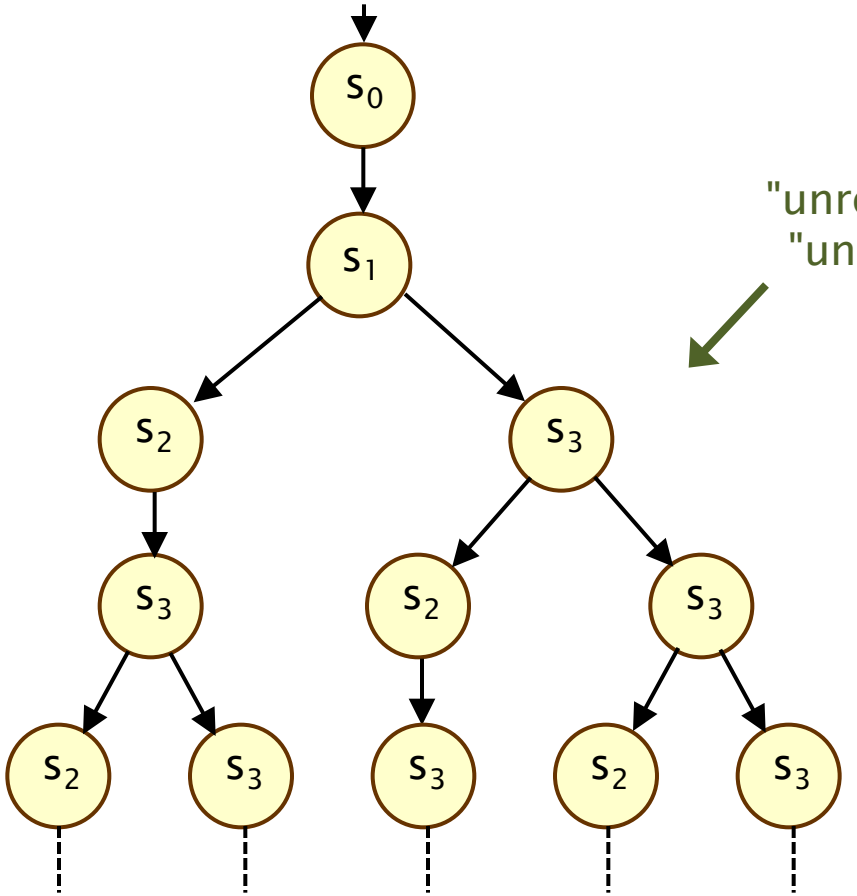
Computation trees

LTS



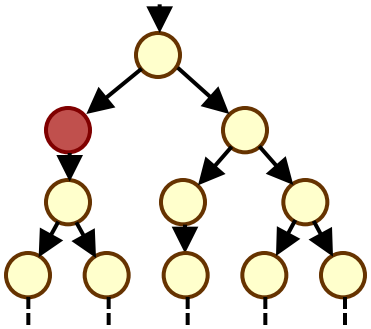
linear-time
vs.
branching-time

Infinite computation tree (a prefix of)

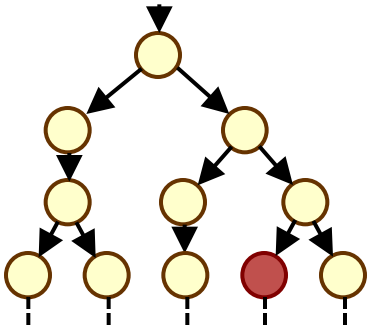


"unrolling" or
"unfolding"

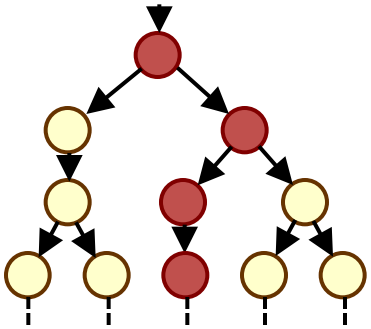
CTL – Intuitive semantics



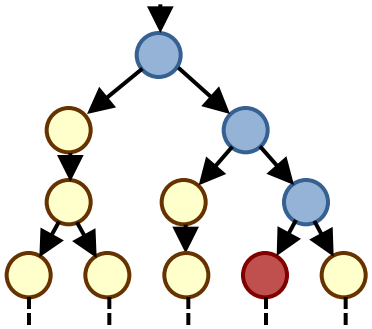
∃◯ red



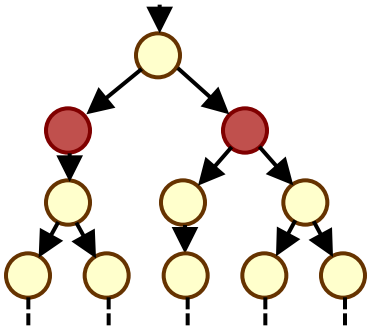
∃◇ red



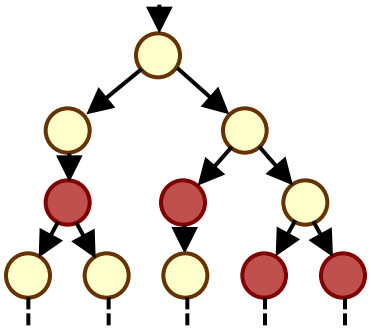
∃□ red



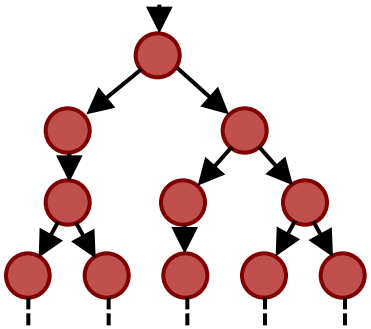
∃ [blue U red]



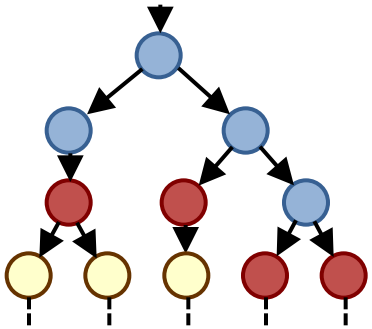
∀◯ red



∀◇ red



∀□ red



∀ [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

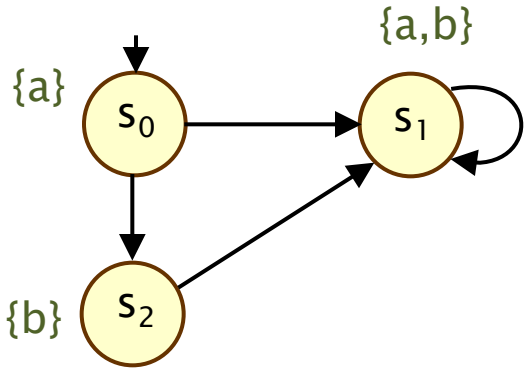
- Semantics of state formulae:
  - $s \models \phi$ denotes "s satisfies ϕ" or "ϕ is true in s"
- For a state s of an LTS (S,Act,→,I,AP,L):
  - $s \models \text{true}$ always
  - $s \models a \Leftrightarrow a \in L(s)$
  - $s \models \phi_1 \wedge \phi_2 \Leftrightarrow s \models \phi_1$ and $s \models \phi_2$
  - $s \models \neg\phi \Leftrightarrow s \not\models \phi$
  - $s \models \forall\, \psi \Leftrightarrow \pi \models \psi$ for all $\pi \in Path(s)$
  - $s \models \exists\, \psi \Leftrightarrow \pi \models \psi$ for some $\pi \in Path(s)$
- and for a path π:
  - $\pi \models \bigcirc \phi \Leftrightarrow \pi[1] \models \phi$
  - $\pi \models \phi_1 \,U\, \phi_2 \Leftrightarrow \exists k \geq 0$ s.t. $\pi[k] \models \phi_2$ and $\forall i < k\ \pi[i] \models \phi_1$

(i+1)th state of path π

Examples

- $s_0 \models \forall\bigcirc b$?
- $s_0 \models \exists\bigcirc \neg b$?
- $s_0 \models \exists(a \; U \; a \wedge b)$?
- $s_0 \models \exists\bigcirc \forall\Box (a \wedge b)$?

M:



CTL equivalences

- Again various operators can be derived
  - propositional logic: $\vee$, $\rightarrow$, $\leftrightarrow$, $\oplus$, …
- Path quantifier duality:
  - $\forall \psi \equiv \neg\exists \neg\psi$
  - $\exists \psi \equiv \neg\forall \neg\psi$
- Temporal operators:
  - $\Diamond \phi \equiv$ true U ϕ
  - $\Box \phi \equiv$?
- For example:
  - $\forall\Box \phi \equiv \neg\exists\Diamond(\neg\phi)$

Expressiveness of CTL and LTL

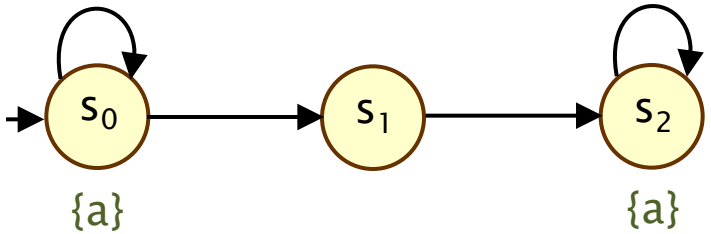
- How do we compare the expressiveness of CTL and LTL?
  - evaluated over states and paths, respectively
  - but they can both be evaluated over models…
- Satisfaction of a CTL formula ϕ by an LTS M:
  - $M \models \phi$ if $s \models \phi$ for all initial states s of M
- CTL formulae ϕ_1 and ϕ_2 are equivalent ($\phi_1 \equiv \phi_2$) if
  - $M \models \phi_1 \Leftrightarrow M \models \phi_2$ (for any LTS M)
- CTL formula ϕ and LTL formula ψ are equivalent ($\phi \equiv \psi$) if
  - $M \models \phi \Leftrightarrow M \models \psi$ (for any LTS M)

Expressiveness of CTL and LTL

- Is CTL more expressive than LTL?
- What can we express in LTL that we cannot in CTL?
  - what about $\Box\Diamond a$?
  - no: $\forall\Box\forall\Diamond a \equiv \Box\Diamond a$
  - similarly: $\forall\Box(a \rightarrow \forall\bigcirc b) \equiv \Box(a \rightarrow \bigcirc b)$
  - what about $\Diamond\Box a$?
  - $\forall\Diamond\forall\Box a \equiv \Diamond\Box a$?

Expressiveness of CTL and LTL

- Counterexample showing: $\forall\Diamond\forall\Box a \not\equiv \Diamond\Box a$:



- In fact, $\Diamond\Box a$ has no equivalent formula in CTL
- Similarly, $\forall\Box\exists\Diamond a$ has no equivalent formula in LTL
- The expressiveness of CTL and LTL are incomparable

CTL vs. LTL

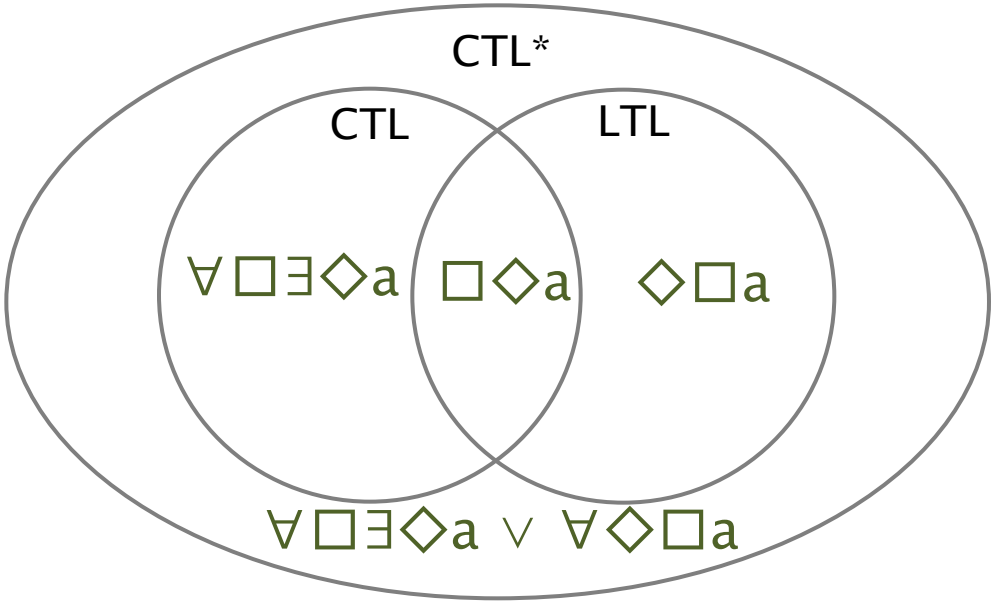
- Key differences between CTL and LTL:
  - branching-time vs. linear-time
  - state-based vs. path-based
  - expressiveness: incomparable
  - model checking algorithms differ
    - CTL simpler and lower complexity than LTL
    - (linear in size of ϕ vs. exponential in size of ψ)
  - fairness dealt with more easily in LTL

- Both CTL and LTL are a subset of the logic CTL*
  - path quantifiers ($\forall$,$\exists$) arbitrarily nested with temporal operators

CTL*

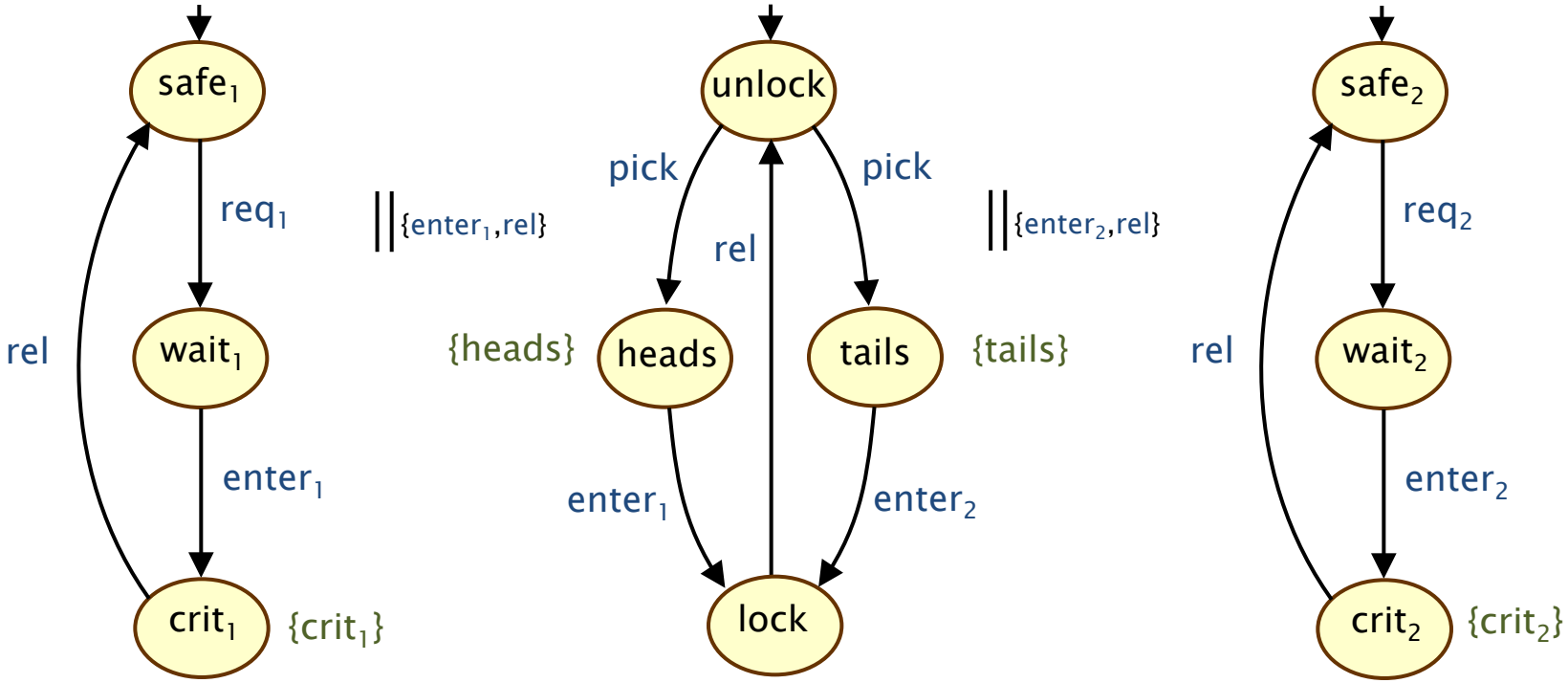
- CTL* syntax
  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid \forall\, \psi \mid \exists\, \psi$
  - $\psi ::= \phi \mid \psi \wedge \psi \mid \neg\psi \mid \bigcirc\, \psi \mid \psi\ U\ \psi \mid \Diamond\, \psi \mid \Box\, \psi$

- Example
  - $\forall\bigcirc\Box a \wedge \exists\Diamond\Box b$



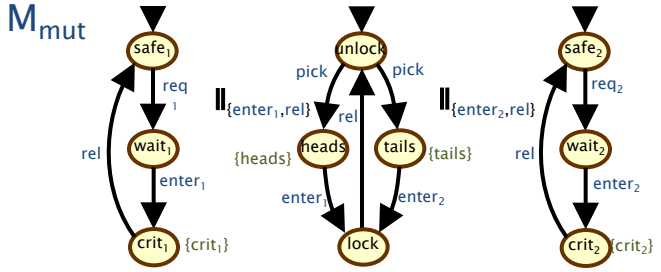
Fairness – motivation

- Rules out (infinite) behaviour considered to be unrealistic
  - often needed in order to verify liveness properties
- Example: two-process mutual exclusion + randomised arbiter
  - properties: $\Box\neg(crit_1 \wedge crit_2)$ and $\Box\Diamond crit_1 \wedge \Box\Diamond crit_2$



Verification under fairness

- For the example M_{mut}:
  - $M_{mut} \not\models \Box\Diamond crit_1 \wedge \Box\Diamond crit_2$



- LTL semantics
  - $M \models \psi \Leftrightarrow$ trace(π) $\models \psi$ for every path π of M
- LTL under fairness
  - $M \models_{fair} \psi \Leftrightarrow$ trace(π) $\models \psi$ for every fair path π of M
- Many fairness conditions can be expressed in LTL
  - e.g. π is fair $\Leftrightarrow \pi \models$ fair where fair $= \Box\Diamond heads \wedge \Box\Diamond tails$
  - for the example: $M_{mut} \models_{fair} \Box\Diamond crit_1 \wedge \Box\Diamond crit_2$
- LTL verification under fairness
  - $M \models_{fair} \psi \Leftrightarrow M \models (fair \rightarrow \psi)$ (assuming M has no terminal states)

Summary

- Temporal logic
  - extends propositional logic with modal/temporal operators
- Linear temporal logic (LTL)
  - logic for linear time properties (over traces, LTSs)
  - syntax (◯, U, ◇, □), semantics, equivalences
- Computation tree logic (CTL)
  - branching-time logic (over states, LTSs)
  - syntax (∀ψ,∃ψ), semantics (computation trees)
- Equivalences, expressiveness, negation, duality
- CTL vs LTL, CTL*, fairness