5. Temporal Logic

Computer-Aided Verification

Dave Parker

University of Birmingham

2017/18

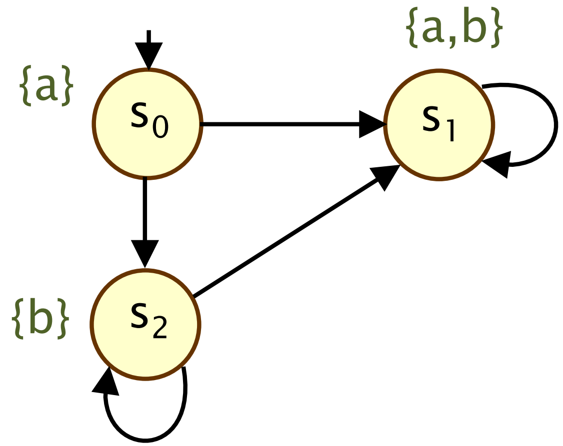
This week

- Next lecture
  - is moved to the tutorial slot:
  - Fri 10am (SportEx Lecture Theatre 1)

- Office hour
  - I am away on Thurs
  - extra office hour today 4.30-5.30

Recap: Traces & properties

- Paths
  - infinite state sequence $\pi = s_0 s_2 s_2 s_1 s_1 s_1 \ldots$
- Traces
  - infinite words over 2^{AP}
  - $\mathrm{trace}(\pi)$ = {a} {b} {b} {a,b} {a,b} {a,b}…



- Linear-time properties
  - set of allowable ("good") traces/words $P \subseteq (2^{AP})^\omega$
  - satisfaction: $M \models P$ if all traces of M are in P
  - e.g. "a is always eventually followed by b"
  - $P = \{ A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid \text{for all } i \geq 0\text{: } a \in A_i \Rightarrow b \in A_j \text{ for some } j \geq i \}$
  - or: linear temporal logic: $\Box(a \rightarrow \Diamond b)$ (see later)

Recap: Properties

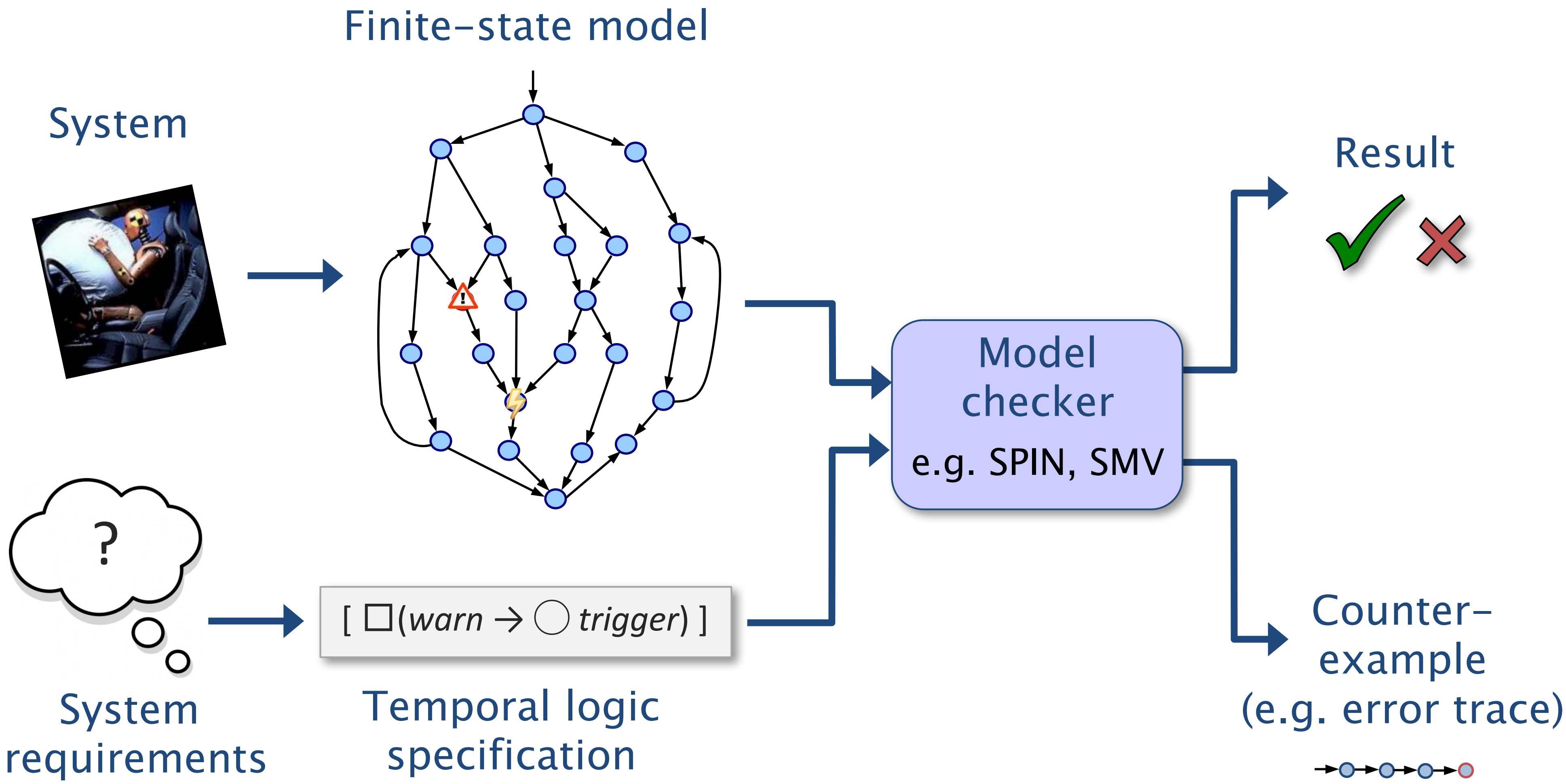
- Key classes of property:

- Invariant: formula Φ is true in all (reachable) states
  - can be checked on each state individually

- Safety property: "nothing bad happens"
  - violating paths have a finite bad prefix

- Liveness: "something good happens in the long run"
  - any finite path can be extended to a satisfying one

Model checking



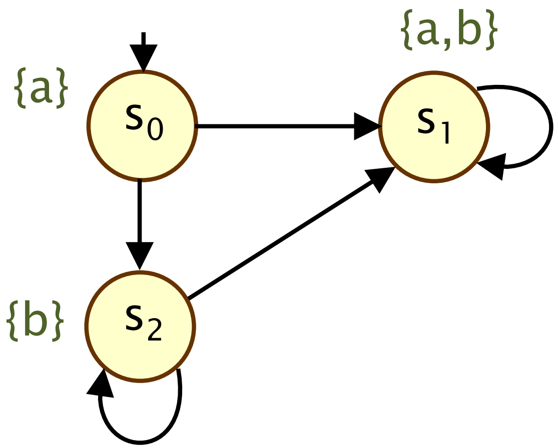
Next (today and next time)

- Propositional logic
- Temporal logic
- Linear temporal logic (LTL)
  - syntax, semantics, examples
- See [BK08] sections 5.1–5.1.4

Propositional logic

- Propositional logic formulas
  - for example: true, a, $\neg a$, $\neg(a \wedge b)$, $a \wedge (b \vee \neg c)$, $a \rightarrow c$
  - where a, b, c are atomic propositions
- Here: use for system observations (state properties)
  - $green_1 \vee green_2$, $fail \wedge \neg alarm$, $\neg(critical_1 \wedge critical_2)$

	a	b	$\neg(a \wedge b)$
$\emptyset$	F	F	T
$\{b\}$	F	T	T
$\{a\}$	T	F	T
$\{a,b\}$	T	T	F



$s_0 \models a$
$s_1 \models a$
$s_0 \models \neg(a \wedge b)$
$s_1 \not\models \neg(a \wedge b)$

Propositional logic: Syntax/semantics

- Syntax (which formulas are allowed)
- Formulas Φ in propositional logic are defined by the grammar:
  - Φ ::= true | false | a | $\Phi \wedge \Phi$ | $\Phi \vee \Phi$ | $\neg\Phi$
  - where $a \in AP$ is an atomic proposition

- Semantics (what formulas mean)
- For a valuation $A \in 2^{AP}$ (a set of “true” propositions for a state), $A \models \Phi$ indicates A "satisfies" a propositional formula Φ:
  - $A \models$ true   always
  - $A \models$ false   never
  - $A \models a \quad \Leftrightarrow \quad a \in A$
  - $A \models \Phi_1 \wedge \Phi_2 \quad \Leftrightarrow \quad A \models \Phi_1$ and $A \models \Phi_2$
  - $A \models \Phi_1 \vee \Phi_2 \quad \Leftrightarrow \quad A \models \Phi_1$ or $A \models \Phi_2$
  - $A \models \neg\Phi \quad \Leftrightarrow \quad A \not\models \Phi$

Example:

$\{a\} \models a \vee b$

$\{a,b\} \not\models \neg(a \wedge b)$

Logical equivalences

- We usually give more minimal grammars, e.g.:
  - Φ ::= true | a | $\Phi \wedge \Phi$ | $\neg\Phi$
  - where $a \in AP$ is an atomic proposition
- Standard logical equivalences
  - false $\equiv$ $\neg$true (false)
  - $\phi_1 \vee \phi_2 \equiv \neg(\neg\phi_1 \wedge \neg\phi_2)$ (disjunction)
  - $\phi_1 \rightarrow \phi_2 \equiv \neg\phi_1 \vee \phi_2$ (implication)
  - $\phi_1 \leftrightarrow \phi_2 \equiv (\phi_1 \rightarrow \phi_2) \wedge (\phi_2 \rightarrow \phi_1)$ (equivalence)
  - $\phi_1 \oplus \phi_2 \equiv (\phi_1 \wedge \neg\phi_2) \vee (\neg\phi_1 \wedge \phi_2)$ (exclusive or)

Temporal logic

- Temporal logic
  - extends propositional logic with modal/temporal operators
  - which can refer to the (infinite) behaviour of a system
  - "temporal" – refers to relative ordering of events, not the precise times at which they happen (LTSs are time-abstract)

- Various applications
  - used, e.g. in philosophy, for many years
  - introduced to formal verification by Pnueli in the 70s
  - increased prominence thanks to model checking (early 80s)

Temporal logic

- Temporal logic for property specification in model checking
  - mathematically precise
  - intuitive (mostly!)
  - concise (usually!)

- LTL: Linear Temporal Logic
  - temporal logic for linear-time properties
  - there are alternatives: branching time (see CTL, later)

- Some key temporal operators
  - $\Diamond$ a – "a is eventually true"
  - $\Box$ a – "a is always true"

LTL – Syntax

- LTL formulas ψ are defined by the grammar:
  - $\psi ::= \text{true} \mid a \mid \psi \wedge \psi \mid \neg\psi \mid \bigcirc\, \psi \mid \psi \;\mathrm{U}\; \psi$
  - where $a \in AP$ is an atomic proposition
- Temporal operators: "next" ($\bigcirc$) and "until" (U)
  - $\bigcirc\, \psi$ means "ψ is true in the next state"
  - $\psi_1 \;\mathrm{U}\; \psi_2$ means "ψ_2 is true eventually and ψ_1 is true until then"
- Equivalences (in addition to false, $\vee$, $\rightarrow$, $\leftrightarrow$, $\oplus$)
  - "eventually ψ": $\Diamond\, \psi \;\equiv\; \text{true} \;\mathrm{U}\; \psi$
  - "always ψ": $\Box\, \psi \;\equiv\; \neg\Diamond(\neg\psi)$

LTL

- Some simple examples:
- $\Box \neg(\text{critical}_1 \wedge \text{critical}_2)$
  - "the processes never enter the critical section simultaneously"
- $\Diamond$ end
  - "the program eventually terminates"
- $\neg$error U end
  - "the program terminates without any errors occurring"

- Alternative styles of syntax
  - $\bigcirc$ a $\equiv$ X a ("next")
  - $\Diamond$ a $\equiv$ F a ("future", "finally")
  - $\Box$ a $\equiv$ G a ("globally")

LTL – Intuitive semantics