# 13. LTL Model Checking

Computer-Aided Verification

Dave Parker

University of Birmingham

2017/18

# Recap: Regular safety properties

- Model checking regular safety property $P_{safe}$ on LTS $M$
  - 1. find NFA $\mathcal{A}$ representing the bad prefixes of $P_{safe}$
  - 2. build LTS-NFA product $M \otimes \mathcal{A}$
  - 3. is an "accept" state reachable in $M \otimes \mathcal{A}$?
    - "no": $P_{safe}$ <u>is</u> satisfied in $M$
    - "yes": $P_{safe}$ is <u>not</u> satisfied in $M$

"invariant"

negation

$$\boxed{M \models P_{safe} \quad \Leftrightarrow \quad M \otimes \mathcal{A} \models \Box \neg accept}$$

$$\boxed{M \not\models P_{safe} \Leftrightarrow \text{some path satisfies } \Diamond accept \text{ in } M \otimes \mathcal{A}}$$

- In this lecture, we generalise this approach
  - to model checking for LTL over LTSs

# Overview

- Nondeterministic Büchi automata (NBAs)
  - to represent ω-regular languages, LTL formulae
  - non-blocking NBAs

- LTS-NBA product

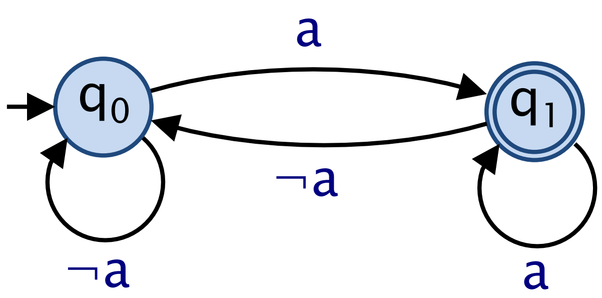
- LTL model checking

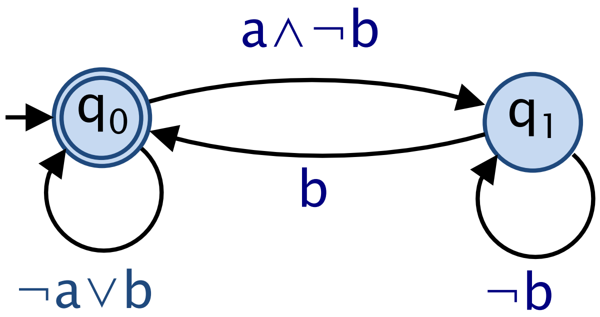
- See [BK08] Sections 4.3-4.4, 5.2

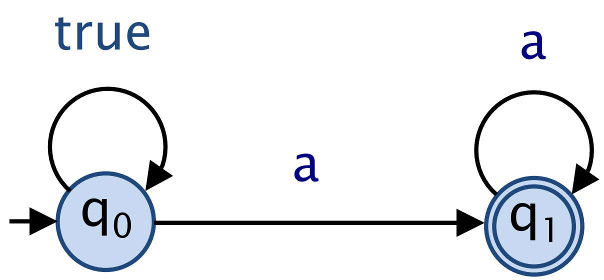
# Nondeterministic Büchi automata

- Nondeterministic Büchi automata (NBAs)
  - represent ω-regular languages (e.g. LTL formulae)

- Examples:

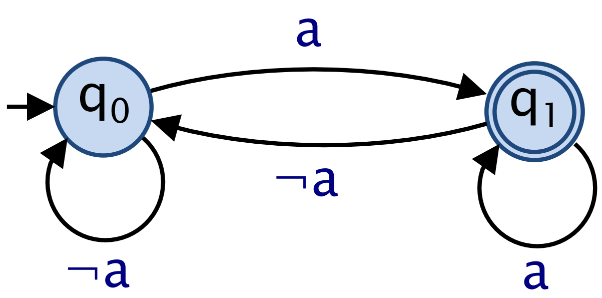


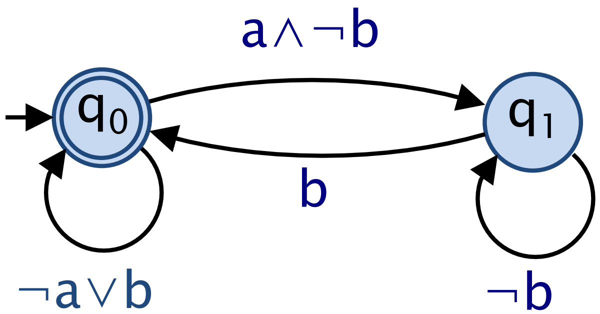



# Nondeterministic Büchi automata

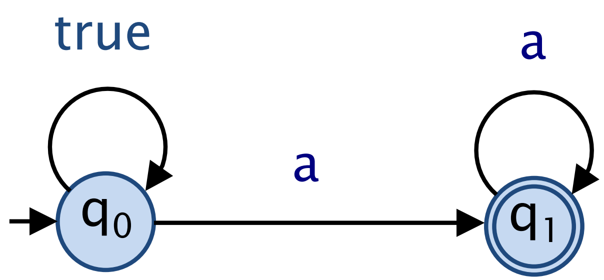
- Nondeterministic Büchi automata (NBAs)
  - represent ω-regular languages (e.g. LTL formulae)
- Examples:




"infinitely often a" – $\Box\Diamond a$



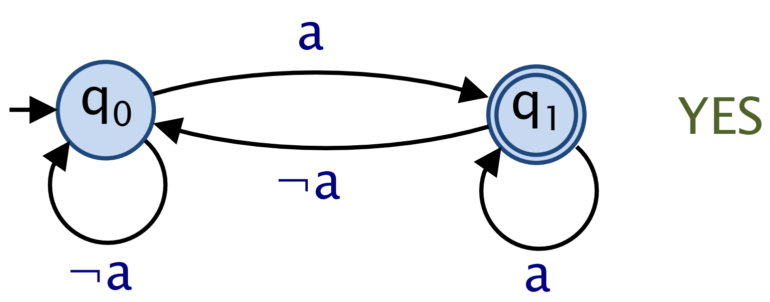

"b always follows a" – $\Box(a \rightarrow \Diamond b)$

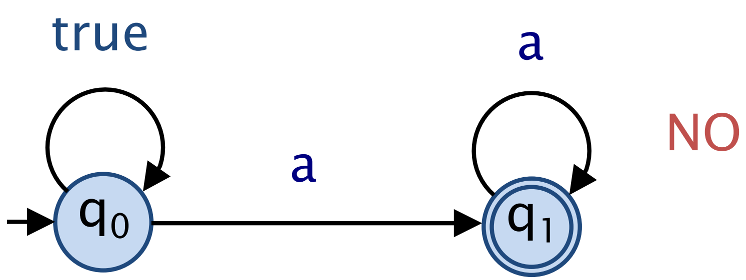


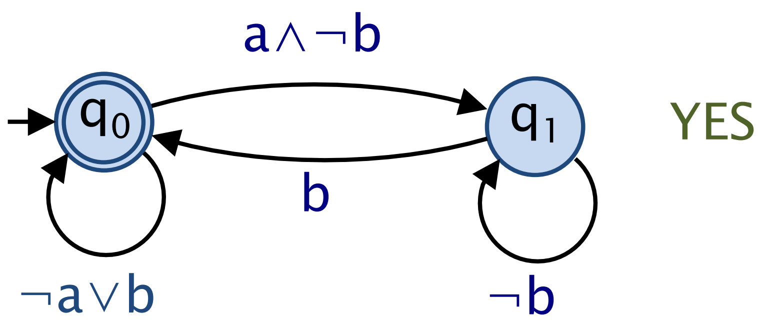

"eventually always a" – $\Diamond\Box a$

# Non-blocking NBAs

- An NBA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ is non-blocking if:
  - every symbol is available in every state
  - i.e. $\delta(q,A) \neq \varnothing$ for all states $q \in Q$ and symbols $A \in \Sigma$
  - so every infinite word has a run through $\mathcal{A}$

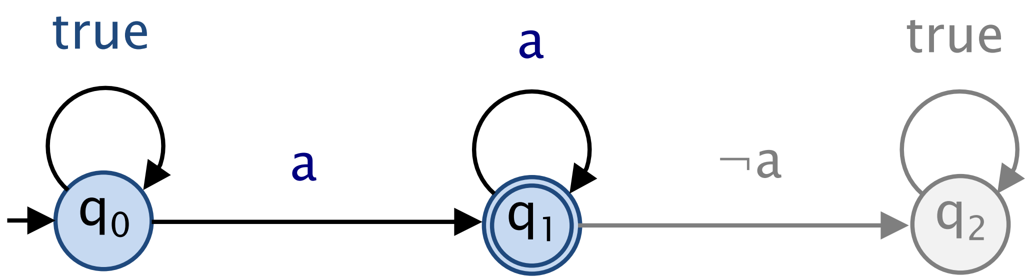


# Non-blocking NBAs

- An NBA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ is non-blocking if:
  - every symbol is available in every state
  - i.e. $\delta(q,A) = \emptyset$ for all states $q \in Q$ and symbols $A \in \Sigma$
  - so every infinite word has a run through $\mathcal{A}$
- We can always convert to a non-blocking NBA
  - by adding a "trap" state



# Product of an LTS and an NBA

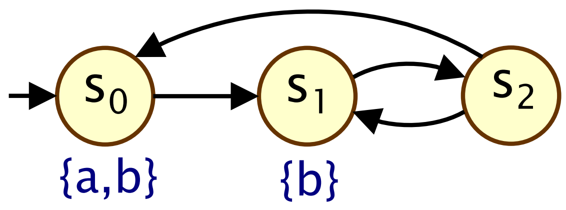
- For an LTS M and a non-blocking NBA $\mathcal{A}$
  - we construct the product of M and $\mathcal{A}$, denoted M $\otimes$ $\mathcal{A}$

- Identical construction to the case of an NFA
  - synchronous parallel composition
  - transitions of NBA $\mathcal{A}$ synchronise with state labels of LTS M
  - allows infinite traces/words that are in <u>both</u> M and $\mathcal{A}$

- Forms the basis of a model checking procedure
  - of ω-regular languages (and thus LTL)

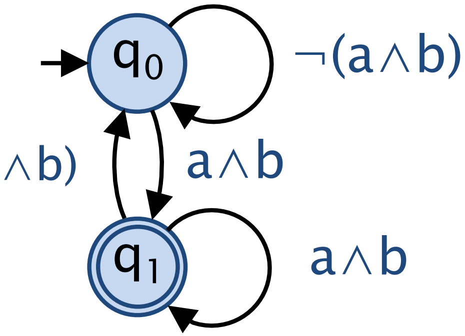
# Example 1 – LTS-NBA product

LTS M

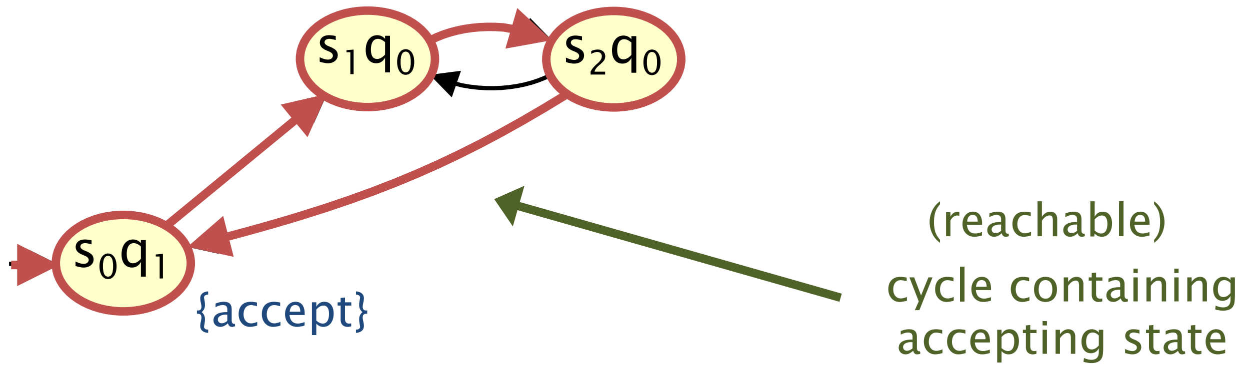



$\psi = \square\diamond(a \wedge b)$

NBA $\mathcal{A}_\psi$




Product $M \otimes \mathcal{A}$




So: there is a path in M which satisfies $\psi$

Note: this is not the procedure for model checking $\psi$ on M

# Model checking LTL

- Given an LTS $M$ and an LTL formula $\psi$
  - $M \models \psi \quad \Leftrightarrow \quad \text{Traces}(M) \subseteq \text{Words}(\psi)$
- Negating the LTL formula (i.e., $\neg\psi$), we have:
  - $M \models \psi \quad \Leftrightarrow \quad \text{Traces}(M) \cap \text{Words}(\neg\psi) = \varnothing$
- Given also an NBA $\mathcal{A}_{\neg\psi}$ representing the formula $\neg\psi$:
  - $M \models \psi \quad \Leftrightarrow \quad \text{Traces}(M) \cap \mathcal{L}_\omega(\mathcal{A}_{\neg\psi}) = \varnothing$
- Constructing the product $M \otimes \mathcal{A}_{\neg\psi}$:
  - $M \models \psi \quad \Leftrightarrow$ there is no accepting path (cycle) in $M \otimes \mathcal{A}_{\neg\psi}$

# LTL model checking

- Model checking LTL formula $\psi$ on LTS M
  - 1. find NBA $\mathcal{A}_{\neg\psi}$ representing the negation $\neg\psi$ of $\psi$
  - 2. build LTS-NBA product $M \otimes \mathcal{A}_{\neg\psi}$
  - 3. is a cycle containing an "accept" state reachable in $M \otimes \mathcal{A}_{\neg\psi}$?
    - "no": $\psi$ <u>is</u> satisfied in M
    - "yes": $\psi$ is <u>not</u> satisfied in M

"persistence"

negation

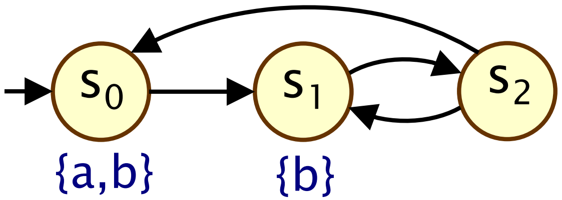
$$\boxed{M \vDash \psi \quad \Leftrightarrow \quad M \otimes \mathcal{A} \vDash \Diamond\Box\neg accept}$$

$$\boxed{M \nvDash \psi \Leftrightarrow \text{some path satisfies } \Box\Diamond accept \text{ in } M \otimes \mathcal{A}_{\neg\psi}}$$

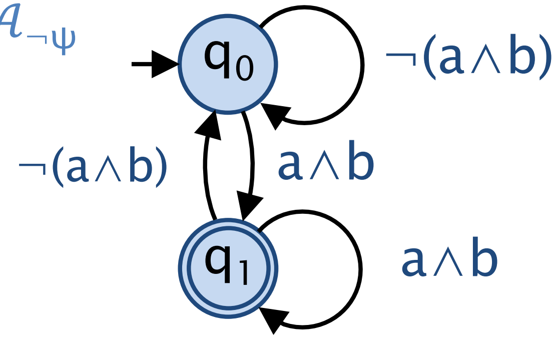
# Example 1 – LTS model checking

Model check $\psi = \Diamond\Box\neg(a \wedge b)$ on LTS M

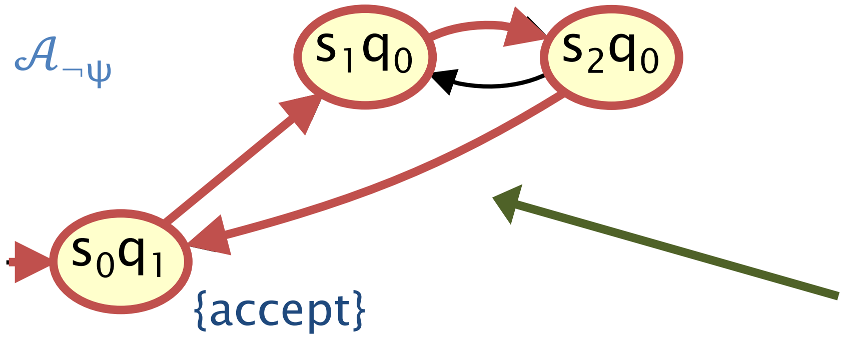
$\neg\psi = \Box\Diamond(a \wedge b)$




NBA $\mathcal{A}_{\neg\psi}$




Product M $\otimes$ $\mathcal{A}_{\neg\psi}$



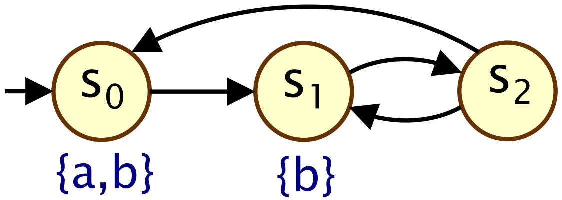

(reachable)
cycle containing accepting state

$M \nvDash \psi$

i.e. $\psi$ not satisfied

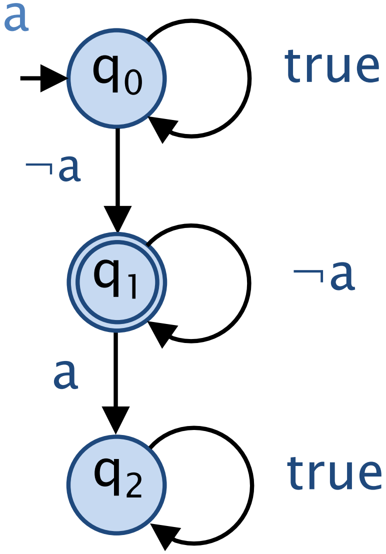
# Example 2 – LTL model checking

Model check $\psi = \Box\Diamond a$ on LTS M

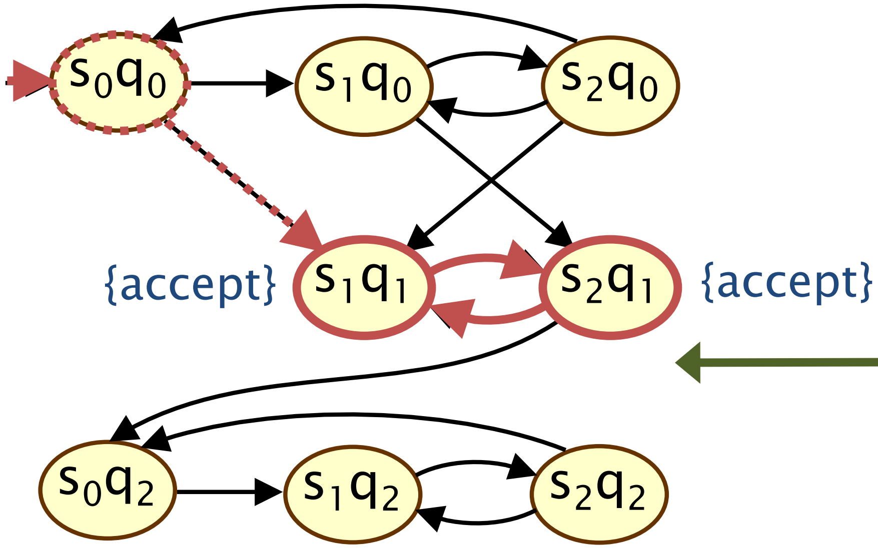



$\neg\psi = \Diamond\Box\neg a$

NBA $\mathcal{A}_{\neg\psi}$




Product M $\otimes$ $\mathcal{A}_{\neg\psi}$




(reachable)

cycle containing accepting state

$M \not\models \psi$

i.e. $\psi$ not satisfied

# Comparison: Linear-time model checking

Model checking regular safety property $P_{safe}$ on LTS M

- 1. NFA $\mathcal{A}$ for bad prefixes of $P_{safe}$
- 2. build product $M \otimes \mathcal{A}$
- 3. is an "accept" state reachable in $M \otimes \mathcal{A}$?
  - "no": $P_{safe}$ is satisfied in M
  - "yes": $P_{safe}$ not satisfied in M

$M \vDash P_{safe} \Leftrightarrow M \otimes \mathcal{A} \vDash \Box\neg accept$

$M \nvDash P_{safe} \Leftrightarrow$ some path satisfies $\Diamond accept$ in $M \otimes \mathcal{A}$

Model checking LTL formula $\psi$ on LTS M

- 1. NBA $\mathcal{A}_{\neg\psi}$ for $\neg\psi$
- 2. build product $M \otimes \mathcal{A}_{\neg\psi}$
- 3. is an "accept" cycle reachable in $M \otimes \mathcal{A}_{\neg\psi}$?
  - "no": $\psi$ is satisfied in M
  - "yes": $\psi$ not satisfied in M

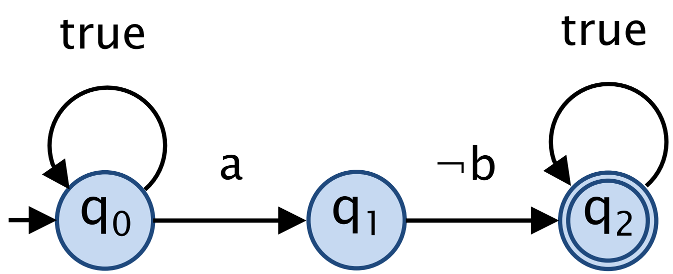
$M \vDash \psi \Leftrightarrow M \otimes \mathcal{A} \vDash \Diamond\Box\neg accept$

$M \nvDash \psi \Leftrightarrow$ some path satisfies $\Box\Diamond accept$ in $M \otimes \mathcal{A}_{\neg\psi}$

# Regular safety properties

- Regular safety properties
  - are a subclass of ω-regular properties
  - and many can be represented as LTL
  - so LTL model checking also works there (but is more costly)

- Recall the example: $\square(a \rightarrow \bigcirc b)$
  - an NBA for its negation is…



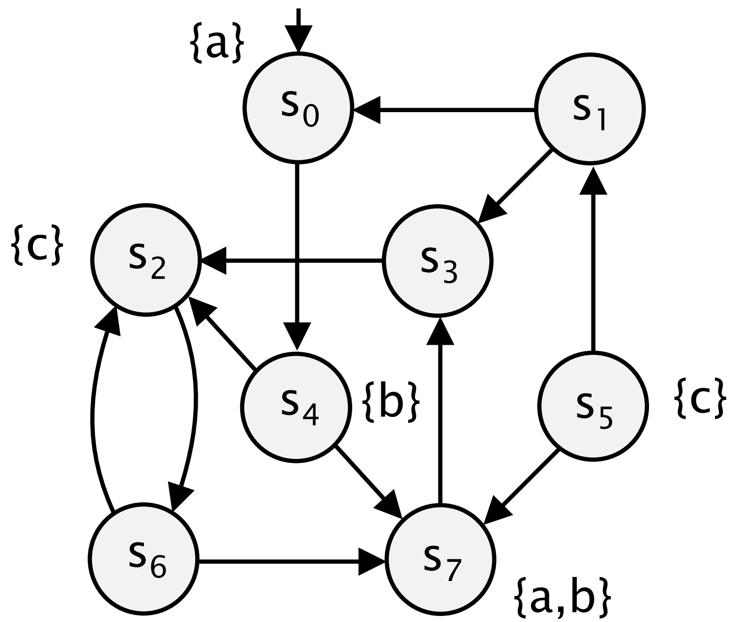
# LTL model checking: Algorithms

- LTL-to-automaton translation
  - various algorithms, tools exist (not covered on this module)
  - (See [BK08] Section 5.2)
- Cycle detection – various options
  - 1. search for reachable non-trivial SCCs containing "accept"
  - 2. find all "accept" states, perform DFS to find back edges
  - 3. nested depth-first search
  - (See [BK08] Section 4.4.2)

# Assm 2 Qu 2

- Recall this example from Assm 2…

- LTS M




- M $\models \Box\Diamond((a \wedge \neg b) \vee \neg c)$
- M $\not\models \Box\Diamond(a \wedge b)$

# Complexity of LTL model checking

- The time complexity of LTL model checking
  - for LTS $M$ and LTL formula $\psi$
- is: $O(|M| \cdot 2^{|\psi|})$
  - i.e. linear in model and exponential in formula size
  - where $|M|$ = number of states + number of transitions in $M$
  - and $|\psi|$ = number of operators in $\psi$
- Worst-case execution:
  - there are LTL formulas $\psi$ whose NBA $\mathcal{A}_{\neg\psi}$ is of size $O(2^{|\psi|})$
  - the product to be analysed is $|M| \cdot |\mathcal{A}_{\neg\psi}|$
  - checking for cycles can be done in linear time (nested DFS)

# Summary

- LTL model checking procedure
  - convert negation of formula to an equivalent NBA
  - construct LTS–NBA product
  - look for cycles containing accept state in the product

- Same basic idea as for regular safety properties
  - (negation + automaton + product + search)

- Next time
  - counterexamples, fairness, complexity, state space explosion, …