14. Model Checking

Computer-Aided Verification

Dave Parker

University of Birmingham

2017/18

Assignments

- Assignment 3 (model checking and automata)
  - out now; due 12 noon next Thur (1 Mar)
  - tutorials in week 9 (8/9 Mar)

- Assignment 5 (“extended” version only)
  - due Thur of week 10 (15 March)
  - released early; out now

Module syllabus

- Modelling sequential and parallel systems
  - labelled transitions systems, parallel composition
- Temporal logic
  - LTL, CTL and CTL*, etc.
- Model checking
  - CTL model checking algorithms
  - automata-theoretic model checking (LTL)
- Verification tools: SPIN
- Advanced verification techniques
  - bounded model checking via propositional satisfiability
  - (symbolic execution), (symbolic model checking)
- Quantitative verification
  - (real-time systems), probabilistic systems

Overview

- Model checking
  - strengths & weaknesses
  - example applications
- Counterexamples
  - evidence of property refutation
  - or witness to desired behaviour
- Complexity & scalability
  - model size is crucial, many approaches to tackle this

LTL model checking: Summary

- Model checking algorithm
  - construct NBA $\mathcal{A}_{\neg\psi}$ for negation of formula ψ to be verified
  - check for reachable "accept" cycles in product $M \otimes \mathcal{A}_{\neg\psi}$

- LTL-to-automaton translation
  - various algorithms, tools exist (not covered on this module)

- 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 (DFS)

Complexity of LTL model checking

- The time complexity of LTL model checking
  - for LTS M and LTL formula ψ
- 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 ψ
- Worst-case execution:
  - there are LTL formulas ψ 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)

CTL & LTL model checking

- Model checking – key ideas
  - LTSs to model nondeterministic/concurrent systems
  - temporal logics to formally define behaviour
  - relationships between logic & automata

- CTL/LTL model checking – differences
  - CTL: recursive descent + backwards model search
  - LTL: automata-based + cycle detection
  - CTL model checking simple and lower complexity

- CTL/LTL model checking – common themes
  - reduce a hard problem to an instance of a simpler one
  - reduce checking of “good” executions to a search for a “bad” one
  - reduction to basic graph algorithms

Model checking: Pros & cons

- Strengths
  - exhaustive analysis, sound mathematical underpinning
  - fully automated, tool support, limited expertise required
  - general verification approach, broadly applicable
  - partial system verification (property-based)
  - diagnostic information (counterexamples) in case of errors

- Weaknesses
  - scalability (state-space explosion)
  - verifies only the stated requirements, not total correctness
  - verifies a model of the system, not the actual system
  - developing appropriately abstract models may require expertise

Verification in practice

- (see Canvas page for links to papers)

- SLAM: model checking for device drivers in Windows
  - more generally: checking for client violation of APIs
  - simple examples: spinlock must be locked/unlocked in strict alternation; a file can be read only after it is opened.
  - uses “counterexample-guided abstraction refinement”

- Example application domains
  - NASA Martian Rover control software
  - safety and dependability of satellite control software
  - breaking/fixing the Needham-Schroeder public-key protocol
  - Facebook: static analysis of code errors in a rapid release cycle

Counterexamples

Example

- Recall this simple concurrent program:

process Inc = **while** true **do if** $x < 200$ **then** $x := x + 1$ **od**

process Dec = **while** true **do if** $x > 0$ **then** $x := x - 1$ **od**

process Reset = **while** true **do if** $x = 200$ **then** $x := 0$ **od**

- Property specification:
  - variable x always remains in the range $\{0,1,\ldots,200\}$
  - i.e., the invariant $\Box$safe where safe means $0 \leq x \leq 200$
- Property is false (not satisfied)
  - evidenced by a path which reaches a state where $x = -1$

Example – counterexample

- Counterexample trace produced by the model checker:

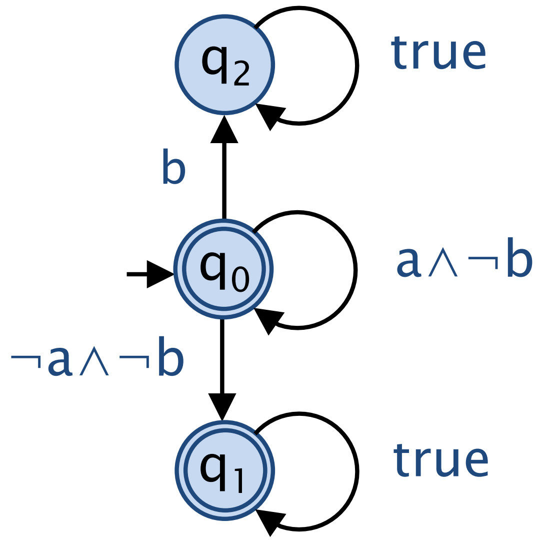
```
..............
605:  proc 1 (Inc)                                            [((x<200))]
606:  proc 1 (Inc)                                            [x = (x+1)]
607:  proc 2 (Dec)                                            [((x > 0))]
608:  proc 1 (Inc)                                            [(1)]
609:  proc 3 (Reset) line 13 "pan_in" (state 2)               [((x==200))]
610:  proc 3 (Reset) line 13 "pan_in" (state 3)               [x=0]
611:  proc 3 (Reset) line 13 "pan_in" (state 1)               [(1)]
612:  proc 2 (Dec) line 5 "pan_in" (state 3)                  [x = (x-1)]
613:  proc 2 (Dec) line 5 "pan_in" (state 1)                  [(1)]
spin: line 17 "pan_in", Error: assertion violated spin: text of failed
assertion: assert(((x>=0)&&(x<=200)))
```

Counterexamples

- Counterexample for $M \not\models \psi$ where ψ is an LTL formula
  - path π of M that refutes ψ (i.e. indicates why ψ is false)
  - in practice: sufficiently long prefix of π showing why π refutes ψ
- Examples
  - counterexample for $\Box a$?
    - finite path ending in $\neg a$
  - counterexample for $\bigcirc a$?
    - 2-state path ending in $\neg a$
  - counterexample for $\Diamond a$?
    - finite prefix of $\neg a$ states followed by single cycle of $\neg a$ states
  - counterexample for $a \: U \: b$?
    - finite path of $a \wedge \neg b$ states ending in $\neg a \wedge \neg b$
    - <u>or</u> finite prefix of $a \wedge \neg b$ states followed by cycle of $a \wedge \neg b$ states

Diversion: LTL model checking of a U b

- Formula to be verified
  - $\psi = a \mathbin{U} b$
- Negation
  - $\neg\psi = \neg(a \mathbin{U} b)$
- Automaton
  - NBA $\mathcal{A}_{\neg\psi}$



Counterexamples

- Counterexample for M $\not\models \psi$ where ψ is an LTL formula
  - path π of M that refutes ψ (i.e. indicates why ψ is false)
  - in practice: sufficiently long prefix of π showing why π refutes ψ
- Examples
  - counterexample for $\square a$?
    - finite path ending in $\neg a$
  - counterexample for $\bigcirc a$?
    - 2-state path ending in $\neg a$
  - counterexample for $\Diamond a$?
    - finite prefix of $\neg a$ states followed by single cycle of $\neg a$ states
  - counterexample for a U b ?
    - finite path of $a \wedge \neg b$ states ending in $\neg a \wedge \neg b$
    - <u>or</u> finite prefix of $a \wedge \neg b$ states followed by cycle of $a \wedge \neg b$ states
  - counterexample for arbitrary LTL formula?
    - finite prefix plus cycle, extracted from LTS-NBA product

Counterexamples (and witnesses)

- Counterexample for $M \not\models \phi$ where ϕ is an CTL formula
  - depends on path quantifier $\forall / \exists$

- For formulae of the form $\forall\psi$ (e.g., $\phi = \forall\Box a$)
  - same as for LTL, just discussed
  - (ignoring nested operators)

- For formulae of the form $\exists\psi$ (e.g., $\phi = \exists\Box a$)
  - there may be no convenient form of counterexamples
  - but is often to natural to provide witnesses when $M \models \phi$
  - a witness is a path giving an example of how ψ can be true
  - a witness for ψ is a counterexample for $\neg\psi$