SS 2017
Software Verification
LTL model checking continued

Prof. Dr. Holger Schlingloff $^{1,2}$
Dr. Esteban Pavese $^1$

(1) Institut für Informatik der Humboldt Universität
(2) Fraunhofer Institut für offene Kommunikationssysteme FOKUS
Recap

- Can you give a recursive characterization of $G^*(p \lor q)$?
- Consider an operator $B^+$ such that
  $$(\phi B^+\psi) \iff X(\phi \lor (\neg\psi \land (\phi B^+\psi)))$$
  Can you express $B^+$ in terms of $U^+$ and/or $W^+$?

- Can you give recursive characterizations of $AG^+p$ and $EG^+p$?

- Can you give a Büchi-automaton for „finitely many a“?
- Can you give a deterministic Büchi-automaton for it?
- Can you prove that?
Back to the homeworks...

• **Theorem:** Büchi automata are more expressive than deterministic Büchi automata.

• **Proof:** Show that there is no deterministic BA accepting the language $L$ of all $\omega$-words which contain only finitely many $a$ (Alphabet $\{a, b\}$)

• **Lemma:** For every dBA $A = (S, R, s_0, S_F, S_R)$ and every infinite input word $w=w_1w_2w_3...$ accepted by $A$ there is a unique sequence of states $\sigma_w : \mathbb{N}_0 \to S$, such that $\sigma_w(0)=s_0$ and $(\sigma_w(i), w_{i+1}, \sigma_w(i+1)) \in R$, and for infinitely many indices $i$ it holds that $\sigma_w(i) \in S_R$
Proof: BA not determinizable

Widerspruchsbeweis: Ann. es gibt ein BA $A = (S, \delta, \omega, s_0, F, \mathcal{R})$ mit $\mathcal{R} = \{a, b, \ldots\}$ und endlich vielen $x$.

Es gibt Sequenz $\omega = \omega_1 \omega_2 \ldots$ der $a_k$. Für ein $s \in \mathcal{S}$ gibt es unendlich viele Indexe $i$ mit $s_i = s$.

Sei in der Liste solche $i$.

Betrachte Wort $\omega = b a b a b a b \ldots = b^i a^i b^i$.

Es gibt für ein $s \in \mathcal{S}$ unendlich viele Indices $i$ mit $s_i = s$.

Sei $\omega$ das kleinste solche $i$ mit $i > i_0$.

Betrachte Wort $\omega = b^i a^i b^i a^i$.

Sei $|\mathcal{S}| = m$.

Betrachte Wort $\omega_{i_0} = b^i a^i b^i a^i b^i a^i \ldots = b^i a^i$.

Beim Akzeptieren dieses Wortes kommt mindestens ein $s \in \mathcal{S}$ in "Anfangstel" mehrfach vor: $s_{i_0} = s_{i_1} = s_{i_2} = \ldots = s_{i_m} = s_{i_m}$.

Betrachte Wort $\omega_0 = \omega_{i_0}^{i_0} (\omega_{i_0}^{i_0} \omega_{i_0}^{i_0})^\omega$.

$\omega_0$ enthält unendlich viele $a$'s.

$\omega_0$ wird von $A$ akzeptiert.
Recap (continued)

- Can you give a recursive characterization of $G^*(p \lor q)$?
- Consider an operator $B^+$ such that
  $$(\phi B^+ \psi) \leftrightarrow X(\phi \lor (\neg \psi \land (\phi B^+ \psi)))$$
  Can you express $B^+$ in terms of $U^+$ and/or $W^+$?

- Can you give a Büchi-automaton for "finitely many $a$"?
- Can you give a deterministic Büchi-automaton for it?

- What is an atom in LTL modelchecking?
- What is the atom graph?
- How is it built?
Kripke structures

- Usually, model $M$ is an LTS, FSM, Kripke-structure, or such
- $M \models \phi$ is read as „for all execution sequences $\sigma$ of $M$ it holds that $\sigma \models \phi$“
- How to check all execution sequences?
  - $\Rightarrow$ depth-first search!

- given formula $\phi$, define $SF(\phi)$ to be the set of all subformulas of $\phi$
  (for reasons which will become clear later, we say that $X\phi$, $X\psi$ and $X(\phi U \psi)$ are subformulae of $(\phi U \psi)$)
- $m \subseteq SF(\phi)$ is propositionally consistent, if
  - not $m \models \bot$ (propositionally), e.g., not $(\psi \in m$ and $\neg \psi \in m)$
  - $\phi U \psi \in m$ iff $X\psi \in m$ or $X\phi \in m$ and $X(\phi U \psi) \in m$
- $atom a = (w, m)$, $w \subseteq P$ interpretation, $m \subseteq SF(\phi)$ prop. cons., $p \in m$ iff $p \in w$
- define an $atom graph$ as „$M \times \phi$“
Atom Graph

- an initial atom is any $a_0=(w_0,m_0)$, where $w_0$ is any initial state of $M$ and $m_0$ is any propositionally consistent set s.t. $\phi \in m_0$
- $(w,m) \rightarrow (w',m')$ if
  - $(w,w') \in \Delta$
  - $\forall \psi \in m$ iff $\psi \in m'$
- atom graph can be constructed depth-first
- Example $\phi = G^+(p \lor q) = \neg (T U^+ \neg (p \lor q))$
  $SF(\phi) =$
- $M =$
Eventualities

- if \( m \) contains \((\phi U^+ \psi)\), some \( m' \) containing \( \psi \) must be reachable
- „reachable“ means „in the same strongly connected component“ (SCC)
- self-fulfilling SCC: for any \( \alpha = (w,m) \) and \((\phi U^+ \psi) \in m\) there is a reachable \( \alpha' = (w',m') \) and \( \psi \in m' \)

\[ \Rightarrow \text{we have to decompose the atom graph into SCCs} \]

- Tarjan’s algorithm is a clever solution to this
- linear complexity (enumerates SCCs as they are encountered)
- overall complexity: \( |M| \times 2^{|\phi|} \)
- meaning: The model must be traversed only once
Tarjan’s algorithm

Tarjan's strongly connected components algorithm

From Wikipedia, the free encyclopedia

**Tarjan's algorithm** is an algorithm in graph theory for finding the strongly connected components of a graph. It runs in linear time, matching the time bound for alternative methods including Kosaraju's algorithm and the path-based strong component algorithm. Tarjan's Algorithm is named for its discoverer, Robert Tarjan.\(^1\)

**Overview**

The algorithm takes a directed graph as input, and produces a partition of the graph's vertices into the graph's strongly connected components. Each

---