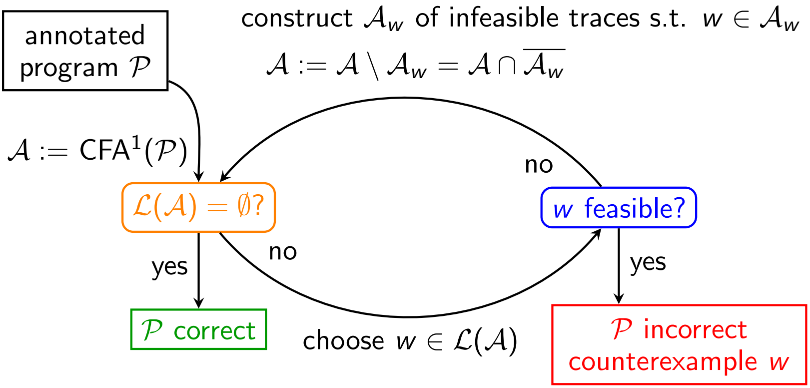
Minimization of Visibly Pushdown Automata Using Partial Max-SAT

Matthias Heizmann, **Christian Schilling**, Daniel Tischner

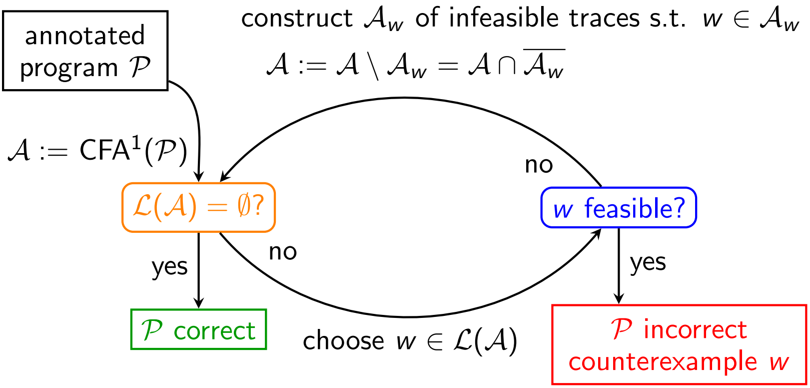
University of Freiburg, Germany

Trace abstraction / Ultimate Automizer



[1] CFA = control flow automaton

Trace abstraction / ULTIMATE AUTOMIZER



- Automaton $\mathcal{A}$ grows **exponentially** in number of iterations unless we apply minimization

[1]CFA = control flow automaton

Visibly pushdown automata (VPA)

- Programs with **procedures**
  Traces also contain calls and returns
- VPA: restricted pushdown automata
  Read words with three types of symbols
  - **internal** – “no stack”
  - **call** – “push current state”
  - **return** – “pop”
- VPA inherit nice properties of finite automata
  - Boolean operations
  - Decidability

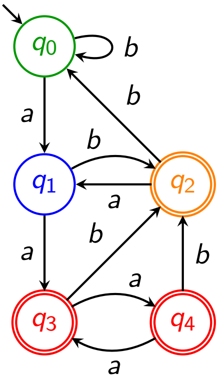
  However, **no minimization**!

Minimization

- Minimization = **reduction** (number of states)
- **Merge** states (according to a congruence)
- Preserve the language

Minimization of finite automata

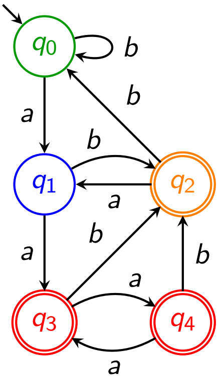
$$(a+b)^*a\,(a+b)$$



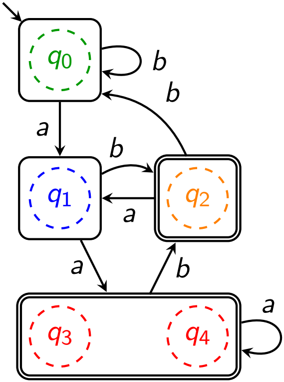
non-minimal DFA

Minimization of finite automata

$$(a+b)^* a\,(a+b)$$



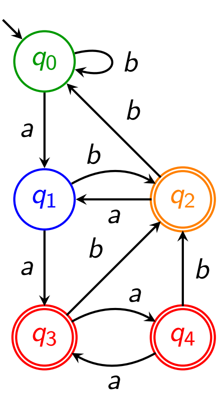
non-minimal DFA



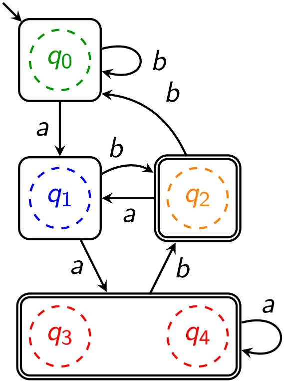
minimal DFA / merge-minimal NFA

Minimization of finite automata

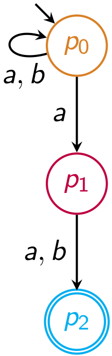
$$(a + b)^* a\,(a + b)$$



non-minimal DFA

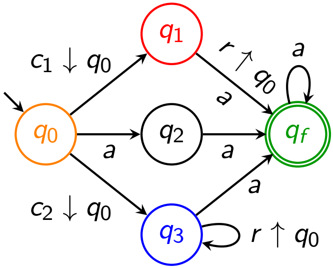


minimal DFA / merge-minimal NFA



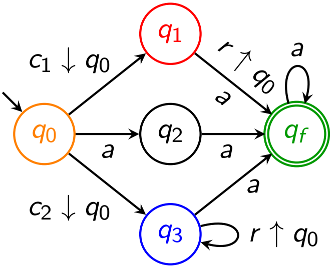
minimal NFA

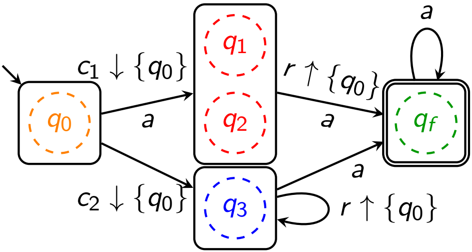
Minimization of VPA



1. Observation:

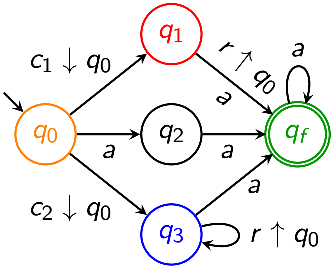
Return transitions can sometimes be ignored

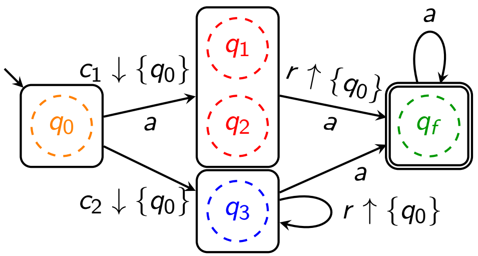


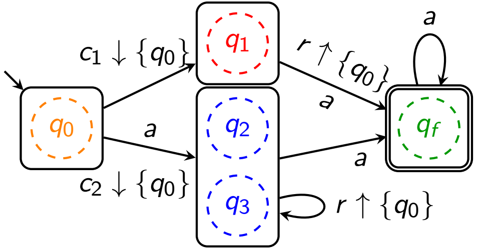


Minimization of VPA

1. Observation:
Return transitions can sometimes be ignored

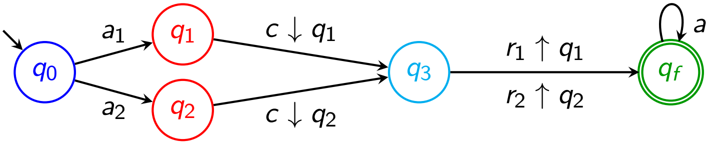


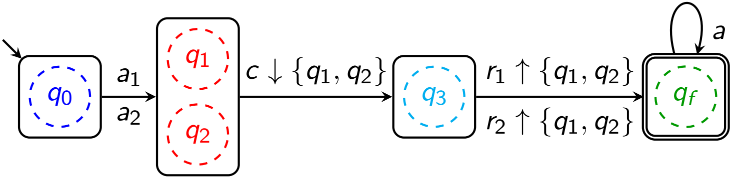




2. Observation:
Ignoring return transitions can destroy transitivity

Minimization of VPA



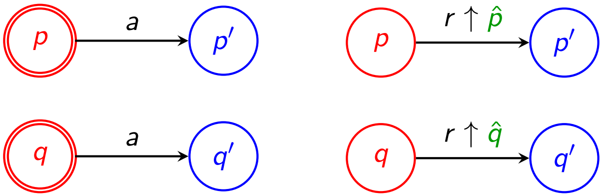


3. Observation:

Merging call predecessors changes the stack alphabet

Congruence for minimization

- Two states are **equivalent** if they
  - are **both accepting** or **both non-accepting**
  - **reach equivalent states** under the same symbol
    - and **equivalent stack symbols** (for returns)



Congruence for minimization

- Two states are **equivalent** if they
  - are **both accepting** or **both non-accepting**
  - **reach equivalent states** under the same symbol
    - and **equivalent stack symbols** (for returns)

- How to compute such a relation?
  - Encode existence as **Boolean formula**
  - Any satisfying assignment represents a congruence

Encoding

- Boolean variables $X_{\{p,q\}}$ for any two states p, q
  - p and q can be merged if $X_{\{p,q\}}$ is true
- Constraints enforce that the relation
  - is an **equivalence relation**
  - is compatible with **acceptance condition**
  - is a **congruence for transition relation**

Equivalence relation

- Reflexivity

$$X_{\{q,q\}} \tag{1}$$

- Symmetry

encoded in variables

- Transitivity

$$X_{\{q_1,q_2\}} \wedge X_{\{q_2,q_3\}} \rightarrow X_{\{q_1,q_3\}} \tag{2}$$

Compatibility with acceptance condition

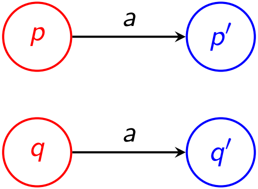
- Accepting state $p \in F$ must not be merged with non-accepting state $q \notin F$

$$\neg \mathsf{X}_{\{p,q\}} \tag{3}$$

Congruence for transition relation

- States are only merged if their successors are merged
  - Internal and call transitions

$$X_{\{p,q\}} \rightarrow X_{\{p',q'\}} \quad (4.1)$$



Congruence for transition relation

- States are only merged if their successors are merged
  - Return transitions

$$X_{\{p,q\}} \wedge X_{\{\hat{p},\hat{q}\}} \rightarrow X_{\{p',q'\}} \tag{4.2}$$

$$p \xrightarrow{r \uparrow \hat{p}} p'$$

$$q \xrightarrow{r \uparrow \hat{q}} q'$$

  - Only required for reachable $\hat{p}, \hat{q}$

Are we done yet?

- Assignment

$$\mathsf{X}_{\{q,q\}} \mapsto \textbf{true} \qquad \mathsf{X}_{\{p,q\}} \mapsto \textbf{false} \ (p \neq q)$$

corresponds to original $\textsc{Vpa}$ – so sad!

PMax-SAT encoding

- Partial maximum satisfiability (PMax-SAT)
  - Clauses are either **hard** or **soft**
  - Assignment must satisfy
    - all **hard** clauses
    - as many **soft** clauses as possible

PMax-SAT encoding

- Partial maximum satisfiability (PMax-SAT)
  - Clauses are either **hard** or **soft**
  - Assignment must satisfy
    - all **hard** clauses
    - as many **soft** clauses as possible
- Consider all clauses so far as **hard** clauses
- Add **soft** clauses

$$X_{\{p,q\}} \tag{5}$$

  Rationale: Merge as many states as possible
- Solution corresponds to a **local optimum**

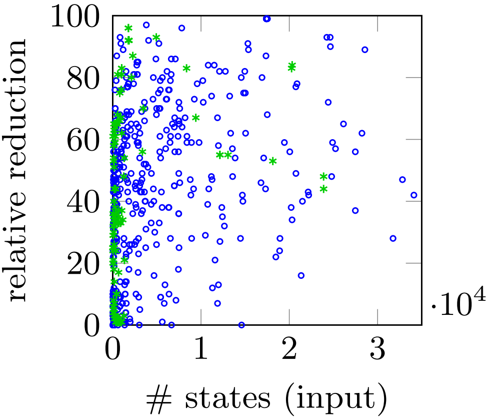
Integration in ULTIMATE AUTOMIZER

- 165 programs from SV-COMP 2016
- Resource limit: 300 s / 4 GiB

minimization used?	# solved	∅ time total	∅ time min.	∅ removal
no	66	16,085	-	-
yes	same 66	15,564	2,649	3,077
	+ 12	101,985	61,384	8,472

times given in ms

Automata from Ultimate Automizer



○ deterministic Vpa   ∗ nondeterministic Vpa

596 data points

Recap

- Algorithm for reducing VPA by merging states
- Reduction to **synthesis** of language-preserving **congruence**
- Reduction to solving a **Boolean optimization problem**