Computation of Certificate Chains with Alternating Pushdown Systems

Javier Esparza

Institute for Formal Methods in Computer Science

University of Stuttgart

Joint work with Stefan Schwoon and Dejvuth Suwimonteerabuth

A bit of history

1997: Bouajjani, E., Maler study the reachability problem of pushdown systems (PDS), and alternating pushdown systems (APDS).

- Very simple polynomial algorithm for PDS.
  Efficiency improved by E., Hansel, Rossmanith, Schwoon in 2000.
- Straightforward extension leads to exponential algorithm for APDS.

1999: Ellison et al. introduce SPKI/SDSI, an authorization framework.

- Security policy given by a set of certificates or certs $P \longrightarrow G$:
  Principal P delegates his/her rights to a group of principals G.
- Proof of authorization by means of a cert chain from authoriser to authorized principal.

2002: Jha and Reps model SPKI/SDSI as PDSs.

- Computation of cert chains reduced to the reachability problem.

Threshold certs

The SPKI/SDSI standard allows for threshold certs $P \longrightarrow (G_1, \ldots, G_n, k)$:

> Principal P delegates his/her rights to all principals that belong to at least k of the groups $G_1, \ldots, G_n$.

Jha and Reps remark that SPKI/SDSI with threshold certs can be naturally modelled by APDS.

This paper: (1) Improves the efficiency of the exponential algorithm for APDS.

(2) Gives an efficient polynomial algorithm for APDSs modelling SPKI/SDSI with threshold certs.

The talk concentrates on (2): Background on (A)PDS and SPKI/SDS, result, experiments.

Pushdown systems

A pushdown system (PDS) is a triple (P, Γ, Δ), where

- P is a finite set of control locations
- Γ is a finite stack alphabet
- $\Delta \subseteq (P \times \Gamma) \times (P \times \Gamma^*)$ is a finite set of rules

A configuration is a pair $\langle p, v \rangle$, where $p \in P$, $v \in \Gamma^*$

If $\langle p, \gamma \rangle \hookrightarrow \langle p', v \rangle \in \Delta$ then $\langle p, \gamma w \rangle \longrightarrow \langle p', vw \rangle$ for every $w \in \Gamma^*$

Normalization: $|v| \leq 2$

For c, c' configurations: c' is reachable from c if $c \rightarrow c_1 \rightarrow \cdots \rightarrow c_n \rightarrow c'$

Alternating pushdown systems

An alternating pushdown system (APDS) is a triple (P, Γ, Δ), where

- P is a finite set of control locations
- Γ is a finite stack alphabet
- $\Delta \subseteq \mathcal{P}(\,(P \times \Gamma) \times (P \times \Gamma^*)\,)$ is a finite set of rules

A configuration is a pair $\langle p, v \rangle$, where $p \in P$, $v \in \Gamma^*$

If $\langle p, \gamma \rangle \hookrightarrow \{\langle p_1, v_1 \rangle, \ldots, \langle p_n, v_n \rangle\} \in \Delta$ then
$\langle p, \gamma w \rangle \longrightarrow \{\langle p_1, v_1 w \rangle, \ldots, \langle p_n, v_n w \rangle\}$ for every $w \in \Gamma^*$

$\{c_1, \ldots, c_k\} \rightarrow C$ if $c_1 \rightarrow C_1, \ldots, c_k \rightarrow C_k$ and $C = C_1 \cup \ldots \cup C_k$

Normalization: $|v_i| \leq 2$ and $n \leq 2$

For C, C' sets of confs: C' is reachable from C if $C \rightarrow C_1 \rightarrow \ldots \rightarrow C_n \rightarrow C'$

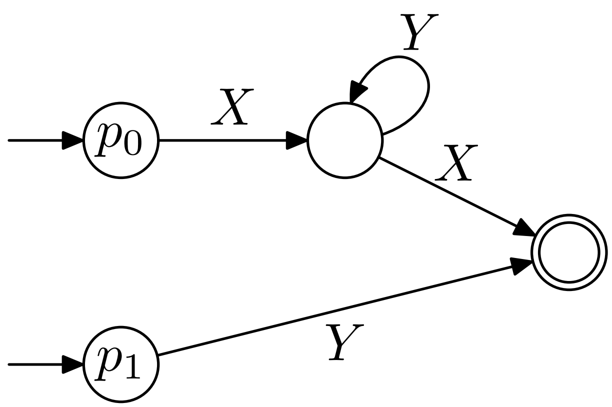
BEM's reachability algorithm for PDSs

Key problem: given a set of configurations C, compute the set of its predecessors/successors

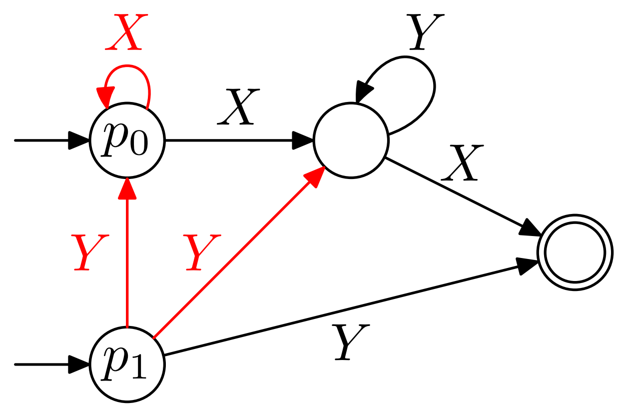
Symbolic representation: use finite automata to represent regular sets of configurations

Polynomial saturation algorithms working on the symbolic representation

$P = \{p_0, p_1\}$, $\Gamma = \{X, Y\}$, $\Delta = \{p_0 X \hookrightarrow p_0, p_1 Y \hookrightarrow p_0, p_1 Y \hookrightarrow p_1 Y X\}$



$\langle p_0, XY^*X\rangle \cup \langle p_1, Y\rangle$



$\text{pre}^*(\langle p_0, XY^*X\rangle \cup \langle p_1, Y\rangle)$

Extension to APDS

Symbolic representation using alternating finite automata

- Transitions of the form $q \xrightarrow{a} Q$
- w accepted if $\{q_0\} \xrightarrow{w} Q$ for some $Q \subseteq F$

Exponential saturation algorithms working on the symbolic representation

Observe:

- a finite automaton with n states and m alphabet letters can have at most $n \cdot m \cdot n = n^2 \cdot m$ transitions.
- an alternating automaton can have $n \cdot 2^n \cdot m$ transitions.

Theorem: The reachability problem for APDS is EXPTIME-complete

Simple SPKI/SDSI

A set of principals identified by their public keys: K_{Alice}, K_{CS}, K_{Uni}

A set of names to describe rôles: `prof`, `student`

A set of name certs to describe relations:

$$K_{CS}\ \mathtt{prof} \longrightarrow K_{Alice} \qquad \text{(Alice is CS prof)}$$
$$K_{Uni}\ \mathtt{prof} \longrightarrow K_{CS}\ \mathtt{prof} \qquad \text{(All CS profs are uni. profs)}$$

A set of authorization certs to grant or delegate rights.

Read $K_P \blacksquare$ as set of rights that P owns

Read $K_P \square$ as set of rights that P owns and can delegate

$$K_{Money}\ \square \rightarrow K_{Uni}\ \mathtt{prof}\ \blacksquare \qquad \text{(Delegation)}$$
$$K_{Money}\ \square \rightarrow K_{Uni}\ \mathtt{prof}\ \square \qquad \text{(``Recursive'' delegation)}$$

Normalization: at most 2 names in right-hand-side

Proof of authorization by cert chains:

$$K_{Money}\,\square \;\Longrightarrow\; K_{Uni}\,\mathtt{prof}\,\blacksquare \;\Longrightarrow\; K_{CS}\,\mathtt{prof}\,\blacksquare \;\Longrightarrow\; K_{Alice}\,\blacksquare$$

Strong analogy between simple SPKI/SDSI systems and pushdown automata:

principals' keys	$\leadsto$	states
names + $\{\blacksquare, \square\}$	$\leadsto$	stack symbols
certs	$\leadsto$	rules
certificate chains	$\leadsto$	computations

Principal K_P has access to resource K_R equivalent to:

$K_P\,\blacksquare$ or $K_P\,\square$ are reachable from $K_R\,\square$

SPKI/SDSI with threshold certificates

Threshold authorization certificates (part of the SPKI/SDSI standard):
Delegate rights to profs that belong to at least k faculties.

$$K_{Money}\,\square \rightarrow (K_{F_1}\,\texttt{prof}\,\square, K_{F_2}\,\texttt{prof}\,\blacksquare, \ldots, K_{F_n}\,\texttt{profs}\,\square, k)$$

Threshold name certificates (not part of the standard):
Declare students that study at least k CS-subjects as CS-students:

$$K_{CS}\,\texttt{student} \rightarrow (K_{sub1}\,\texttt{student}, \ldots, K_{sub_n}\,\texttt{student}, k)$$

Certificates for $k = n$ correspond to alternating pushdown rules

$$K_{Money}\,\square \rightarrow \{\, K_{F_1}\,\texttt{prof}\,\square, K_{F_2}\,\texttt{prof}\,\blacksquare, \ \ldots\ , K_{F_n}\,\texttt{prof}\,\square \,\}$$

Normalization: $n = k = 2$ (possible blowup).

Complexity of the authorization problem I

Theorem:

Let n, c_0, c_1, c_2 as before.

Let c_{ta} be the number of threshold authorization certs

Let c_{tn} be the number of threshold name certs

The authorization problem for SPKI/SDSI with both threshold authorization and threshold name certs is EXPTIME-complete and can be solved in time

$$O(c_0 + c_{ta} + 2^n c_1 + 4^n (n c_2 + c_{tn}))$$

> Ellison et al., 1999: "*The reason that a threshold subject may not appear in a name cert is . . . which would almost surely be too convoluted to be usable in practice.*"

Complexity of the authorization problem II

Theorem: The authorization problem for SPKI/SDSI with only threshold authorization certs can be solved in time

$$O(c_0 + c_{ta} + n\,c_1 + n^2\,c_2)$$

Idea of the proof: In this case the saturation algorithm cannot add any alternating rules to the initial alternating automaton.

Coincides with best known algorithm for simple SPKI/SDSI when $c_{ta} = 0$.

Implementation and experiments

Algorithm implemented on top of the NEXUS platform for context-aware systems

- NEXUS provides middleware to obtain context data (e.g. geographical neighbours) about mobile objects registered at the platform

Scenario: Trade fair

- Visitors move around the exhibition halls
- Mobile phones used to obtain visitors' locations
- Company X launches a promotion: customers of X can freely download ringtones if they visit X's stand, and can authorize another person of their choice.

  For this, X's manager only needs to add the certificate

$$K_{ring} \,\square \rightarrow \{Stand_X \;\texttt{visitor}\; \square, K_X \;\texttt{customer}\; \square\}$$

Experiments:

- Some thousands of visitors and about 100 hierarchically organized stands
- Database queries and data transmision simulated by opening and closing files
- Between 25 and 500 milliseconds to grant/reject acess for realistic values of the parameters.

Summary

Algorithm for reachability in APDS with detailed complexity analysis

Application to SPKI/SDSI's threshold certs

Ef£cient polynomial algorithm for SPKI/SDSI's standard

Theoretical support for design choice to leave threshold name certs out

Implementation on top of a platform for context-aware systems

Promising experimental results

Also in the paper: more efficient algorithm for computing attractors in PDS games (inspired by Cachat's work).