A Framework for Linear Authorization Logics

Vivek Nigam
Universidade Federal da Paraíba
Based on a LICS’12 paper. An extended version is available on my homepage.
Proof-Carrying Authorization (PCA) [Appel and Felten CCS’99]
Proof-Carrying Authorization (PCA) [Appel and Felten CCS’99]

Alice

Γ

Policy

Bob
Proof-Carrying Authorization (PCA) [Appel and Felten CCS’99]
At the center of PCA lie the policies and the use of formal proofs.

Proof-Carrying Authorization (PCA) [Appel and Felten CCS’99]
Proof-Carrying Authorization (PCA)

$\Gamma$

Policy

Authorization Logics
Proof-Carrying Authorization (PCA)

\[ \Gamma \]

Policy

Authorization Logics

Access control logics for distributed systems [Abadi et al. '93].

Modal Logics:

\[ P \supset K \text{ says } P \]
\[ K \text{ says } (P_1 \supset P_2) \supset K \text{ says } P_1 \supset K \text{ says } P_2 \]
\[ K \text{ says } (K \text{ says } P) \supset K \text{ says } P \]
In many situations, we would like to express effect-based policies.

“\textbf{A principal may have access to a room at most once.}”

“A principal \textbf{may not} withdraw more money than the money available in her bank account.”

\textbf{Linear Authorization Logics} [Garg et al. ESORICS’06]
Our main contributions

We propose a logical framework where different linear authorization logics may live together. We show that in this framework one can express a wider range of policies.

“A principal may use a set of (low-ranked) policy rules, but not a set of (high-ranked) policy rules.”
Our main results

Complexity Results

Provability Problem for LAL
Our main results

<table>
<thead>
<tr>
<th>Complexity Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provability Problem for LAL</td>
</tr>
<tr>
<td>MELL</td>
</tr>
</tbody>
</table>

Notice that for MELL the same problem is still open.
Our main results

<table>
<thead>
<tr>
<th>Provability Problem for LAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MELL</td>
</tr>
<tr>
<td>FOL Balanced Bipoles</td>
</tr>
</tbody>
</table>

Notice that for MELL the same problem is still open.

Propositional Classical auth. logics is also PSPACE-complete
Agenda

- Linear Authorization Logic
  - Undecidability
  - Proof search and MSR
  - PSPACE-completeness
  - Conclusions and Future Work
Linear Logic Basics
Linear Logic Basics

Multiplicative Fragment

\[ \frac{\Gamma, F, G \to H}{\Gamma, F \otimes G \to H} \] _\otimes_L

\[ \frac{\Gamma_1 \to F \quad \Gamma_2 \to G}{\Gamma_1, \Gamma_2 \to F \otimes G} \] _\otimes_R

\[ \frac{\Gamma_1 \to F \quad \Gamma_2, G \to H}{\Gamma_1, \Gamma_2, F \multimap G \to H} \] _\multimap_L

\[ \frac{\Gamma, F \multimap G}{\Gamma \multimap F \multimap G} \] _\multimap_R
Linear Authorization Logics [Garg et al.]
Three Families of Modalities

K says P  K knows P  K has P
Linear Authorization Logics [Garg et al.]

Three Families of Modalities

$K$ says $P$
Three Families of Modalities

\[ \text{\textcolor{red}{K says } P} \]

A lax modality denoting that the principal \( K \) affirms the formula \( P \):

\[
\frac{\Gamma, P \rightarrow K \text{ says } G}{\Gamma, K \text{ says } P \rightarrow K \text{ says } G} \quad \text{say}_{L} \\
\frac{\Gamma \rightarrow P}{\Gamma \rightarrow K \text{ says } P} \quad \text{say}_{R}
\]
Linear Authorization Logics [Garg et al.]

Three Families of Modalities

\[ K \text{ knows } P \]
Three Families of Modalities

Since knowledge is unrestricted, one is allowed to contract as well as weaken it:

\[
\frac{\Gamma \rightarrow G}{\Gamma, K \text{ knows } P \rightarrow G} \quad W
\]

\[
\frac{\Gamma, K \text{ knows } P, K \text{ knows } P \rightarrow G}{\Gamma, K \text{ knows } P \rightarrow G} \quad C
\]
Linear Authorization Logics [Garg et al.]

Three Families of Modalities

\[ K \text{ knows } P \]

\[
\frac{\Gamma, P \rightarrow G}{\Gamma, K \text{ knows } P \rightarrow G} \quad \text{knows}_L
\]

\[
\frac{\Psi \rightarrow P}{\Psi \rightarrow K \text{ knows } P} \quad \text{knows}_R
\]

where \( \Psi \) contains only formulas of the form \( K \text{ knows } F \).
Linear Authorization Logics [Garg et al.]

Three Families of Modalities

A restricted modality denoting that the principal $K$ has the consumable resource $P$:

$$\Gamma, P \rightarrow G \quad \text{has}_L \quad \frac{\Psi, \Delta \rightarrow P}{\Psi, \Delta \rightarrow K \text{ has } P} \quad \text{has}_R$$

where $\Psi$ contains only formulas of the form $K$ knows $F$, while $\Delta$ contains only formulas of the form $K$ has $F$. 
Linear Logic with Subexponentials [NM’09, DJS’93]
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical

$!^b, !^r$ and $?^b, ?^r$: 
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical

$!^b, !^r$ and $?^b, ?^r$:

$$!^b F \neq !^r F \quad ?^b F \neq ?^r F$$
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical

\(!^b, !^r\) and \(?^b, ?^r\):

\[!^b F \not\equiv !^r F \quad ?^b F \not\equiv ?^r F\]

All other connectives are canonical.
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical

\( !^b, !^r \) and \( ?^b, ?^r \):

**Subexponentials**  
\( !^b F \neq !^r F \quad ?^b F \neq ?^r F \)

All other connectives are canonical.
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical

\(!^b, !^r \text{ and } ?^b, ?^r:\)  

**Subexponentials**

\(!^b F \not\equiv !^r F \quad ?^b F \not\equiv ?^r F\)

Subexponential Signature

\(\langle I, \leq, U \rangle\)

where \(U \subseteq I\) and is closed under \(\leq\).
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical

\(!^b, !^r\) and \(?^b, ?^r\):

Subexponentials

\(!^b F \neq !^r F\)

\(?^b F \neq ?^r F\)

All other connectives are canonical.

Subexponential Signature

\(\langle I, \leq, U \rangle\)

where \(U \subseteq I\) and is closed under \(\leq\).

Subexponentials with index \(a \in U\) can weaken and contract:

\[
\frac{\Gamma, !^a P, !^a P \rightarrow G}{\Gamma, !^a P \rightarrow G} \quad C \quad \frac{\Gamma \rightarrow G}{\Gamma, !^a P \rightarrow G} \quad W
\]
Linear Logic with Subexponentials [NM’09, DJS’93]

Linear Logic Exponentials are Not Canonical

\(!^b, !^r\) and \(?^b, ?^r\):

Subexponentials

\(!^b F \neq !^r F\)

\(?^b F \neq ?^r F\)

All other connectives are canonical.

Subexponential Signature

\[\langle I, \leq, U \rangle\]

where \(U \subseteq I\) and is closed under \(\leq\).

Subexponentials with index \(a \in U\) can weaken and contract:

\(\Gamma, !^a P, !^a P \rightarrow G\)

\(\Gamma, !^a P \rightarrow G\)

\(\Gamma \rightarrow G\)

\(\Gamma \rightarrow G\)

\(W\)

\(C\)

Introduction Rules

\(\Gamma, !^a P, !^a P \rightarrow G\)

\(\Gamma \rightarrow G\)

\(\Gamma \rightarrow G\)

\(W\)

\(C\)

\(!^x F_1, \ldots !^x F_n \rightarrow G\)

\(!^a G\)

\(!^x F_1, \ldots !^x F_n \rightarrow !^a G\)

\(!^a R\)

\(!^x F_1, \ldots !^x F_n, F \rightarrow ?^x G_{n+1}\)

\(?^x G_{n+1}\)

\(!^x F_1, \ldots !^x F_n, ?^a F \rightarrow ?^x G_{n+1}\)

\(?^x G_{n+1}\)

\(?^a L\)

where \(a \leq x_i\) for all \(i\).
**Linear Logic with Subexponentials** [NM’09, DJS’93]

**Linear Logic Exponentials are Not Canonical**

\[ !^b, !^r \text{ and } ?^b, ?^r: \]

- \( !^b F \not\equiv !^r F \)
- \( ?^b F \not\equiv ?^r F \)

### Subexponential Signature

\[ \langle I, \leq, U \rangle \]

where \( U \subseteq I \) and is closed under \( \leq \).

Subexponentials with index \( a \in U \) can weaken and contract:

\[
\frac{\Gamma, !^a P, !^a P \rightarrow G}{\Gamma, !^a P \rightarrow G} \quad C \quad \frac{\Gamma \rightarrow G}{\Gamma, !^a P \rightarrow G} \quad W
\]

### Introduction Rules

\[
\frac{!^x_1 F_1, \ldots, !^x_n F_n \rightarrow G}{!^a \rightarrow !^x_1 F_1, \ldots, !^x_n F_n} !^a_R
\]
\[
\frac{!^x_1 F_1, \ldots, !^x_n F_n, F \rightarrow ?^x_{n+1} G}{!^x_1 F_1, \ldots, !^x_n F_n, ?^a F \rightarrow ?^x_{n+1} G} ?^a_L
\]

where \( a \leq x_i \) for all \( i \).

**Theorem:** For any subexponential signature, \( \Sigma \), \( \text{SELL}_\Sigma \) admits cut-elimination.
Encoding Linear Authorization Logics

global

gl
Encoding Linear Authorization Logics

global knows

\[ \cdots \]

\[ \cdots \]

\[ \cdots \]

\[ k_{k1} \]

\[ k_{ki} \]

\[ k_{kn} \]
Encoding Linear Authorization Logics

global \quad \text{knows} \quad \text{has}

\begin{align*}
&k_{k_1} \quad h_{k_1} \\
&\quad \cdots \quad \cdots \\
&g_{l} \quad k_{k_i} \quad h_{k_i} \\
&\quad \cdots \quad \cdots \\
&k_{k_n} \quad h_{k_n}
\end{align*}
Encoding Linear Authorization Logics
Encoding Linear Authorization Logics

\[
\begin{align*}
\text{global} & \quad \text{knows} & \quad \text{has} & \quad \text{linear} & \quad \text{says} \\
\text{gl} & \quad \rightarrow & \quad \rightarrow & \quad \rightarrow & \quad S_{k1} \\
& \quad \rightarrow & \quad \rightarrow & \quad \rightarrow & \quad \cdots \\
& \quad \rightarrow & \quad \rightarrow & \quad \rightarrow & \quad \cdots \\
& \quad \rightarrow & \quad \rightarrow & \quad \rightarrow & \quad S_{kn}
\end{align*}
\]
Encoding Linear Authorization Logics

\[
\begin{align*}
\llbracket F \text{ knows } K \rrbracket_L &= !^k_K \llbracket F \rrbracket_L \\
\llbracket F \text{ has } K \rrbracket_L &= !^h_K \llbracket F \rrbracket_L \\
\llbracket F \text{ knows } K \rrbracket_R &= !^k_K \llbracket F \rrbracket_R \\
\llbracket F \text{ has } K \rrbracket_R &= !^h_K \llbracket F \rrbracket_R
\end{align*}
\]

\[
\begin{align*}
!^{gl}\{\Theta\}, !^k_K \{\Gamma\} &\rightarrow F \\
!^{gl}\{\Theta\}, !^k_K \{\Gamma\} &\rightarrow !^k_K F \\
!^{gl}\{\Theta\}, !^k_K \{\Gamma\}, !^h_K \{\Delta\} &\rightarrow F \\
!^{gl}\{\Theta\}, !^k_K \{\Gamma\}, !^h_K \{\Delta\} &\rightarrow !^h_K F
\end{align*}
\]
Encoding Linear Authorization Logics

\[
\Gamma, P \rightarrow K \text{ says } G \\
\Gamma, K \text{ says } P \rightarrow K \text{ says } G
\]

\[
[[\Gamma]]_L, [[P]]_L \rightarrow ^{\text{lin}} ?^S_k [[G]]_R
\]

\[
[[\Gamma]]_L, ^{\text{lin}} ?^S_k [[P]]_L \rightarrow ^S_k [[G]]_R
\]
Theorem: The sequent $\Gamma \rightarrow F$ is provable in linear authorization logic if and only if the sequent $[[\Gamma]]_L \rightarrow [[F]]_R$ is provable in SELL.
Encoding Linear Authorization Logics

global knows says

\[ \begin{align*}
\text{gl} & \leftarrow k_{k1} \quad \cdots \quad \cdots \\
& \leftarrow k_{ki} \quad \cdots \\
& \leftarrow k_{kn}
\end{align*} \]

\[ \begin{align*}
\text{sR}_{k1} & \\
& \cdots \\
\text{sR}_{ki} & \\
& \cdots \\
\text{sR}_{kn}
\end{align*} \]
Encoding Linear Authorization Logics

global knows says Trigger

\[
\begin{align*}
&\text{gl} \\
&\text{k}_1 \\
&\text{k}_i \\
&\text{k}_n \\
&\text{sR}_1 \\
&\text{sR}_i \\
&\text{sR}_n \\
&\text{el} \\
&\text{eh} \\
&\text{e} \\
&\text{l} \\
&\text{h}
\end{align*}
\]

Lower Ranked Policies
Higher Ranked Policies
Encoding Linear Authorization Logics

- **global**
- **knows**
  - $k_{k1}$
  - $\ldots$
  - $\ldots$
  - $k_{kn}$

- **says**
  - $sR_{k1}$
  - $\ldots$
  - $\ldots$
  - $sR_{kn}$

- **Trigger**
  - $el$
  - $eh$
  - $e$

**Lower Ranked Policies**

**Higher Ranked Policies**

\[
\Gamma \rightarrow F \\
\Gamma \rightarrow !^{el}F \\
!^{el}_R \\
\Gamma, !^{!}(\Gamma_L) \rightarrow !^{el}F \\
n \times W
\]
admin knows (superuser($K_1$)) $\otimes K_1$ says ($K_2$ has $P$) $\rightarrow$ $K_2$ has $P$

admin knows (user($K_1$)) $\otimes \neg^{eh} K_1$ says ($K_2$ has $P$) $\rightarrow$ $K_2$ has $P$
Agenda

- Linear Authorization Logic

**Undecidability**

- Proof search and MSR
- PSPACE-completeness
- Conclusions and Future Work
Undecidability of Multiplicative Linear Authorization Logic

Two counter machine
Two counter machine

Instructions (uniquely labelled)

(Add $r_1$) $a_k$: $r_1 = r_1 + 1$; goto $b_j$
(Add $r_2$) $b_k$: $r_2 = r_2 + 1$; goto $a_j$
(Sub $r_1$) $a_k$: $r_1 = r_1 - 1$; goto $b_j$
(Sub $r_2$) $b_k$: $r_2 = r_2 - 1$; goto $a_j$

(0-test $r_1$) $a_k$: if $r_1 = 0$ then goto $b_{j1}$ else goto $b_{j2}$
(0-test $r_2$) $b_k$: if $r_2 = 0$ then goto $a_{j1}$ else goto $a_{j2}$

(Jump$_1$) $a_k$: goto $b_j$
(Jump$_1$) $b_k$: goto $a_j$
Undecidability of Multiplicative Linear Authorization Logic

Two counter machine

Instructions (uniquely labelled)

(Add $r_1$) $a_k$: $r_1 = r_1 + 1$; goto $b_j$
(Add $r_2$) $b_k$: $r_2 = r_2 + 1$; goto $a_j$
(Sub $r_1$) $a_k$: $r_1 = r_1 - 1$; goto $b_j$
(Sub $r_2$) $b_k$: $r_2 = r_2 - 1$; goto $a_j$

(0-test $r_1$) $a_k$: if $r_1 = 0$ then goto $b_{j_1}$ else goto $b_{j_2}$
(0-test $r_2$) $b_k$: if $r_2 = 0$ then goto $a_{j_1}$ else goto $a_{j_2}$

(Jump$_1$) $a_k$: goto $b_j$
(Jump$_1$) $b_k$: goto $a_j$

Computations

$$\langle a_1, n, 0 \rangle \xrightarrow{a_1} \cdots \xrightarrow{b_j} \langle a_i, n_i, m_i \rangle \xrightarrow{a_i} \langle b_k, n_k, m_k \rangle \xrightarrow{b_k} \cdots$$
Undecidability of Multiplicative Linear Authorization Logic

Two counter machine

Instructions (uniquely labelled)

(Add r1) ak: \( r_1 = r_1 + 1; \text{goto } b_j \)
(Add r2) bk: \( r_2 = r_2 + 1; \text{goto } a_j \)
(Sub r1) ak: \( r_1 = r_1 - 1; \text{goto } b_j \)
(Sub r2) bk: \( r_2 = r_2 - 1; \text{goto } a_j \)

(0-test r1) ak: if \( r_1 = 0 \) then goto \( b_{j1} \) else goto \( b_{j2} \)
(0-test r2) bk: if \( r_2 = 0 \) then goto \( a_{j1} \) else goto \( a_{j2} \)

(Jump1) ak: goto \( b_j \)
(Jump1) bk: goto \( a_j \)

Computations

\[ \langle a_1, n, 0 \rangle \xrightarrow{a_1} \cdots \xrightarrow{b_j} \langle a_i, n_i, m_i \rangle \xrightarrow{a_i} \langle b_k, n_k, m_k \rangle \xrightarrow{b_k} \cdots \]

Final State

\[ \langle a_0, 0, 0 \rangle \]
Two counter machine

Instructions (uniquely labelled)

(Add \(r_1\) \(a_k\): \(r_1 = r_1 + 1\); goto \(b_j\))

(Add \(r_2\) \(b_k\): \(r_2 = r_2 + 1\); goto \(a_j\))

(Sub \(r_1\) \(a_k\): \(r_1 = r_1 - 1\); goto \(b_j\))

(Sub \(r_2\) \(b_k\): \(r_2 = r_2 - 1\); goto \(a_j\))

(0-test \(r_1\) \(a_k\): if \(r_1 = 0\) then goto \(b_j_1\)
else goto \(b_j_2\))

(0-test \(r_2\) \(b_k\): if \(r_2 = 0\) then goto \(a_j_1\)
else goto \(a_j_2\))

(Jump_1 \(a_k\): goto \(b_j\))

(Jump_1 \(b_k\): goto \(a_j\))

Computations

\[ \langle a_1, n, 0 \rangle \rightarrow_{a_1} \cdots \rightarrow_{b_j} \langle a_i, n_i, m_i \rangle \rightarrow_{a_i} \langle b_k, n_k, m_k \rangle \rightarrow_{b_k} \cdots \]

Final State

\[ \langle a_0, 0, 0 \rangle \]

The termination problem for two-counter machines is undecidable.
Undecidability of Multiplicative Linear Authorization Logic

Translation
Assume two principals $A$ and $B$, where $A$ is responsible for the register 1 and $B$ for the register 2.
Undecidability of Multiplicative Linear Authorization Logic

Translation

Assume two principals $A$ and $B$, where $A$ is responsible for the register 1 and $B$ for the register 2.

Configurations (similar for $b$-states)

$\langle a_i, n_i, m_i \rangle$

$A$ has $r_1, \ldots, A$ has $r_1, B$ has $r_2, \ldots B$ has $r_2 \longrightarrow A$ has $a_i$

$n_i$ copies $m_i$ copies
ADD₁:  \((A \text{ has } r₁ \rightarrow B \text{ says } b_j) \rightarrow A \text{ says } a_k\)
ADD₂:  \((B \text{ has } r₂ \rightarrow A \text{ says } a_j) \rightarrow B \text{ says } b_k\)
SUB₁:  \((A \text{ has } r₁ \otimes B \text{ says } b_j) \rightarrow A \text{ says } a_k\)
SUB₂:  \((B \text{ has } r₂ \otimes A \text{ says } a_j) \rightarrow B \text{ says } b_k\)
0-IF₁:  \(B \text{ has } (B \text{ says } b_{j₁}) \rightarrow A \text{ says } a_k\)
0-IF₂:  \(A \text{ has } (A \text{ says } a_{j₁}) \rightarrow B \text{ says } b_k\)
0-ELSE₁:  \((A \text{ has } r₁ \rightarrow B \text{ says } b_{j₂}) \otimes A \text{ has } r₁ \rightarrow A \text{ says } a_k\)
0-ELSE₂:  \((B \text{ has } r₂ \rightarrow A \text{ says } a_{j₂}) \otimes B \text{ has } r₂ \rightarrow B \text{ says } b_k\)
JUMP₁:  \(B \text{ says } b_j \rightarrow A \text{ says } a_k\)
JUMP₂:  \(A \text{ says } a_j \rightarrow B \text{ says } b_k\)
FINAL:  \(A \text{ has } \top \otimes B \text{ has } \top \rightarrow A \text{ says } a₀\)
Completeness

$$\text{ADD}_1: (A \text{ has } r_1 \rightarrow B \text{ says } b_j) \rightarrow A \text{ says } a_k$$
Undecidability of Multiplicative Linear Authorization Logic

Completeness

ADD$_1$: $(A$ has $r_1 \rightarrow B$ says $b_j) \rightarrow A$ says $a_k$

Backchaining

$$
\frac{
A \text{ says } a_k \rightarrow A \text{ says } a_k
}{
\Gamma \rightarrow A \text{ says } a_k
}
\quad
\frac{
\Gamma, A \text{ has } r_1 \rightarrow B \text{ says } b_j
}{
\Gamma \rightarrow A \text{ has } r_1 \rightarrow B \text{ says } b_j
}
\quad
\frac{
\Gamma \rightarrow A \text{ has } r_1 \rightarrow B \text{ says } b_j
}{
\neg R
}
\quad
\text{ADD}_1
$$
Undecidability of Multiplicative Linear Authorization Logic

Completeness

0-IF$_1$: $B$ has ($B$ says $b_{j_1}$) $\vdash A$ says $a_k$
Undecidability of Multiplicative Linear Authorization Logic

Completeness

0-IF\(_1\): \(B\) has \((B\ says\ b_{j_1})\) \(\rightarrow\) \(A\ says\ a_k\)

Backchaining

\[
\begin{align*}
&\frac{A\ says\ a_k \rightarrow A\ says\ a_k}{\Gamma \rightarrow A\ says\ a_k} \\
&\frac{\frac{\frac{\Gamma \rightarrow B\ says\ b_{j_1}}{\Gamma \rightarrow B\ has\ (B\ says\ b_{j_1})}}{I}}{\frac{\frac{\frac{\frac{\Gamma \rightarrow B\ has\ (B\ says\ b_{j_1})}{\Gamma \rightarrow \ has^R}}{0-IF\_1}}{\Gamma \rightarrow A\ says\ a_k}}}
\end{align*}
\]
Soundness

For soundness, we need more invariants on how \textit{says} formulas move while splitting the context.

\textbf{Lemma:} Sequents of the form below are not provable:

\[ !^g\{\Theta \}_M, C \text{ says } q_i, D \text{ says } q_j, \Gamma \rightarrow E \text{ says } q_k \]

\textbf{Lemma:} If the sequent of the following form is provable:

\[ !^g\{\Theta \}_M, D \text{ says } q_j, \Gamma \rightarrow C \text{ says } q_k, \]

then

\[ \langle q_k, m, n \rangle \rightarrow^* \langle q_j, 0, 0 \rangle \]

\textit{without} any transition using the if case of zero instructions.
Main Result

**Theorem**  The encoding of two counter machines is sound and complete.

**Corollary**  The propositional multiplicative fragment for linear authorization logics with two principals and no function symbols is **undecidable**.
Agenda

- Linear Authorization Logic
- Undecidability

Proof search and MSR

- PSPACE-completeness
- Conclusions and Future Work
Can we interpret policies as multiset rewrite rules?
Can we interpret policies as multiset rewrite rules?

States

\[ T ::= K \text{ says } A \mid K \text{ has } A \mid K \text{ says } T \mid K \text{ has } T \]
Can we interpret policies as multiset rewrite rules?

States

\[ T ::= K \text{ says } A \mid K \text{ has } A \mid K \text{ says } T \mid K \text{ has } T \]

No knowledge as one can easily use it to encode the existential Horn implication problem, which is undecidable.
Can we interpret policies as multiset rewrite rules?

States

\[ T ::= K \text{ says } A \mid K \text{ has } A \mid K \text{ says } T \mid K \text{ has } T \]

Policy Rules (Bipoles)

\[ \forall \vec{y}[!^eT_1 \otimes \cdots \otimes !^eT_m] \rightarrow \exists \vec{x}.[T'_1 \otimes \cdots \otimes T'_n] \]

Pre-condition

Post-condition

Fresh Values
Encoding Linear Authorization Logics

global knows

\[ k_{k1} \quad \cdots \quad \cdots \quad k_{kn} \]

\[ g_l \rightarrow k_{ki} \quad \cdots \quad \cdots \quad k_{kn} \]

\[ sR_{k1} \quad \cdots \quad sR_{ki} \quad \cdots \quad sR_{kn} \]

\[ l \rightarrow el \quad eh \rightarrow h \]

Lower Ranked Policies

Higher Ranked Policies
Can we interpret policies as multiset rewrite rules?

States

\[ T ::= K \text{ says } A \mid K \text{ has } A \mid K \text{ says } T \mid K \text{ has } T \]

Policy Rules (Bipoles)

\[ \forall \vec{y} [!^e T_1 \otimes \cdots \otimes !^e T_m] \longrightarrow \exists \vec{x}. [T_1' \otimes \cdots \otimes T_n'] \]

Pre-condition

\[ \exists \vec{x}. [T_1' \otimes \cdots \otimes T_n'] \]

Post-condition

Fresh Values

Simple proofs!

\[ T_1'' \rightarrow T_1 \quad \cdots \quad T_m'' \rightarrow T_m \quad !^h\{\Gamma_H}, \mathcal{T}, T_1', \ldots, T_k' \rightarrow G \]

\[ !^h\{\Gamma_H}, \mathcal{T}, T_1'', T_2'', \ldots, T_m'' \rightarrow G \]
Can we interpret policies as multiset rewrite rules?

Simple proofs!

\[ T''_1 \rightarrow T_1 \quad \cdots \quad T''_m \rightarrow T_m \quad \vdash^h \{ \Gamma_H \}, \mathcal{T}, T'_1, \ldots, T'_k \rightarrow G \]

\[ \vdash^h \{ \Gamma_H \}, \mathcal{T}, T''_1, T''_2, \ldots, T''_m \rightarrow G \]

**Lemma:** Checking whether a sequent of the form \( T \rightarrow T' \) is provable is in NP. It is bounded by the number of modalities in \( T \) and \( T' \).
Can we interpret policies as multiset rewrite rules?

**States**

\[ T ::= K \text{ says } A \mid K \text{ has } A \mid K \text{ says } T \mid K \text{ has } T \]

**Goals**

\[ !^e T_G \otimes \top \]
Can we interpret policies as multiset rewrite rules?

States

\[ T ::= K \text{ says } A \mid K \text{ has } A \mid K \text{ says } T \mid K \text{ has } T \]

Goals

\[ !^e T_G \otimes \top \]

Simple proofs!

\[ T'' \rightarrow T_G \quad \frac{!^h\{\Gamma_H\}, \mathcal{T} \rightarrow \top}{!^h\{\Gamma_H\}, \mathcal{T}, T'' \rightarrow !^e T_G \otimes \top} \]
Can we interpret policies as multiset rewrite rules?

**States**

\[ T ::= K \text{ says } A \mid K \text{ has } A \mid K \text{ says } T \mid K \text{ has } T \]

**Goals**

\[ !^e T_G \otimes T \]

Simple proofs!

\[
\begin{align*}
T'' & \rightarrow T_G \\
!^h \{\Gamma_H\}, T & \rightarrow T_R \\
!^h \{\Gamma_H\}, T', T'' & \rightarrow !^e T_G \otimes T
\end{align*}
\]

**Theorem:** Proof search using only derivations of the forms above is sound and complete.
Can we interpret policies as rewrite rules?

Principals

A

B

C
Can we interpret policies as rewrite rules?

Principals | Tables | New Tables
--- | --- | ---
A | | |
B | | |
C | | |
Agenda

- Linear Authorization Logic
- Undecidability
- Proof search and MSR
- **PSPACE-completeness**
- Conclusions and Future Work
Restriction based on [Kanovich, Rowe, Scedrov]
Restriction based on [Kanovich, Rowe, Scedrov]

Balanced Bipoles

$$\forall \vec{y}^{\exists}[!^eT_1 \otimes \cdots \otimes !^eT_m] \rightarrow \exists \vec{y}.[T'_1 \otimes \cdots \otimes T'_n]$$

$$n = m$$
Restriction based on [Kanovich, Rowe, Scedrov]

Balanced Bipoles

\[ \forall \vec{y}[^e T_1 \otimes \cdots \otimes ^e T_m] \rightarrow \exists \vec{y}.[T'_1 \otimes \cdots \otimes T'_n] \]

\[ n = m \]

\[
\begin{align*}
T''_1 & \rightarrow T_1 \\
\cdots & \\
T''_m & \rightarrow T_m \\
\hastype{!}{\{ \Gamma_H, \mathcal{T}, T'_1, \ldots, T'_n \}} & \rightarrow G
\end{align*}
\]

\[
\begin{align*}
\hastype{!}{\{ \Gamma_H, \mathcal{T}, T''_1, T''_2, \ldots, T''_m \}} & \rightarrow G
\end{align*}
\]

Number of \( T \)-formulas to the left-hand-side of sequents is always the same.
**Parameters** based on [Kanovich, Ban Kirigin, Nigam, and Scedrov]

- $\mathcal{L}$ is finite first-order alphabet without function symbols with $J$ predicate symbols and $D$ constant symbols;
- $k$ is an upper bound on the arity of predicate symbols;
- $\mathcal{P}$ is a finite set of **balanced bipoles** specifying the policy rules;
- $\mathcal{T}$ is a multiset of exactly $m$ $T$-formulas specifying the initial contents of the sequent.
- $G$ is $G$-formula appearing at the right-hand-side of the sequent.

**Problem**

The sequent $!^{h}\{\mathcal{P}\}, \mathcal{T} \rightarrow G$ is provable or not in SELL

**Theorem**: There is an algorithm that determines whether a sequent $!^{h}\{\mathcal{P}\}, \mathcal{T} \rightarrow G$ is provable or not and runs in $\text{PSPACE}$ with respect to the parameters above.
PSPACE-completeness

**PSPACE lower bound**

Easy sound and complete encoding of a Turing Machine that accepts in space $n$.

**PSPACE upper bound**

**Lemma**: Checking whether a sequent of the form $T \longrightarrow T'$ is provable is in NP. It is bounded by the number of modalities in $T$ and $T'$.

**Lemma**: The upper bound $M$ on the number of modalities in a $T$-formula appearing in a sequent $S$ is the same as the upper bound in any one of its cut-free proofs.

**Lemma**: There are at most $MJ(D + 2mk)^k$ different $T$-formulas.

**Theorem**: There is an algorithm that determines whether a sequent $!^h\{\mathcal{P}\}, T \longrightarrow G$ is provable or not and runs in PSPACE with respect to the parameters above.
Conclusions and Future Work

We proposed a logical framework for linear authorization logics.

We showed that the MELL fragment of LAL is undecidable.

We proposed a novel first-order fragment of LAL for which provability is PSPACE-complete.

Future Work

Investigate the use of subexponentials on formulas appearing in the postcondition of rules. [CONCUR’13]

Decidable fragments when using knows modalities.
Questions