Complexity Optimal Decision Procedure for PDL with Parallel Composition

Joseph Boudou

IRIT, Toulouse University, France

IJCAR 2016, Coimbra
Propositional Dynamic Logic (PDL)

Syntax

\[ \alpha, \beta ::= a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \]  
\[ \varphi, \psi ::= p \mid \bot \mid \varphi \rightarrow \psi \mid \langle \alpha \rangle \varphi \]  

Semantics

\[ \langle a \rangle p \]
Propositional Dynamic Logic (PDL)

Syntax

\[ \alpha, \beta ::= \ a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \]  

(programs)

\[ \varphi, \psi ::= \ p \mid \bot \mid \varphi \rightarrow \psi \mid \langle \alpha \rangle \varphi \]  

(formulas)

Semantics

\[ [a] q \]
Propositional Dynamic Logic (PDL)

Syntax

\[
\alpha, \beta ::= \ a \ | \ (\alpha ; \beta) \ | \ (\alpha \cup \beta) \ | \ \alpha^* \ | \ \varphi? \\
\varphi, \psi ::= \ p \ | \ \bot \ | \ \varphi \rightarrow \psi \ | \ \langle \alpha \rangle \varphi
\]

(formulas)

(programs)

Semantics

\[\langle a ; b \rangle r\]
Propositional Dynamic Logic (PDL)

Syntax

\[ \begin{align*}
\alpha, \beta & ::= \ a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \\
\varphi, \psi & ::= \ p \mid \bot \mid \varphi \to \psi \mid \langle \alpha \rangle \varphi
\end{align*} \]

(formulas)

(p programs)

Semantics

\[ [a \cup b] q \]
Propositional Dynamic Logic (PDL)

Syntax

\[ \alpha, \beta ::= \ a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \quad \text{(programs)} \]

\[ \varphi, \psi ::= \ p \mid \bot \mid \varphi \rightarrow \psi \mid \langle \alpha \rangle \varphi \quad \text{(formulas)} \]

Semantics

\[ \langle p? \rangle \top \]

Diagram:

```
     q, r
    /   \
   /     \n  a     b
 / \     / \n p a   p \rightarrow p, q
 |   / \ \\
 | /   \ \
 b /     \
 |     /  \\
 |    /     \\
 b \  /       \\
| /           \\
|/             \\
| p, q         
|               
|               
```
Propositional Dynamic Logic (PDL)

Syntax

\[ \alpha, \beta ::= a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \quad \text{(programs)} \]

\[ \varphi, \psi ::= p \mid \bot \mid \varphi \rightarrow \psi \mid \langle \alpha \rangle \varphi \quad \text{(formulas)} \]

Semantics

\[ \langle b^* \rangle r \]
Propositional Dynamic Logic (PDL)

Syntax

\[ \alpha, \beta ::= a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \]  

\[ \varphi, \psi ::= p \mid \bot \mid \varphi \rightarrow \psi \mid \langle \alpha \rangle \varphi \]  

Semantics

\[ \langle (p? ; b)^* \rangle r \]
Propositional Dynamic Logic (PDL)

Properties

- Satisfiability problem is EXPTIME-complete.
- Tree-like model property.

Fischer-Ladner closure

\[ \langle a \rangle p \]

\[ p \]

\[ q, r \]

\[ a \]

\[ b \]

\[ p, q \]

\[ a \]

\[ p, q \]

\[ b \]

\[ p, q \]
Propositional Dynamic Logic (PDL)

Properties

- Satisfiability problem is \(\text{EXPTIME}\)-complete.
- Tree-like model property.

Fischer-Ladner closure

\[ [a] q \]

\begin{align*}
q \\
\end{align*}

\begin{align*}
q, r \\
\end{align*}

\begin{align*}
a \\
b \\
p, q \\
p, q \\
\end{align*}
Propositional Dynamic Logic (PDL)

Properties

- Satisfiability problem is EXPTIME-complete.
- Tree-like model property.

Fischer-Ladner closure

\[
\langle a ; b \rangle r \\
\langle a \rangle \langle b \rangle r \\
\langle b \rangle r
\]
Propositional Dynamic Logic (PDL)

Properties

- Satisfiability problem is EXPTIME-complete.
- Tree-like model property.

Fischer-Ladner closure

\[
[a \cup b]q \\
[a]q \quad [b]q \quad q
\]
Propositional Dynamic Logic (PDL)

Properties

- Satisfiability problem is EXPTIME-complete.
- Tree-like model property.

Fischer-Ladner closure

\[ \langle p? \rangle T \]

\[ p \]

\[ p \]

\[ p, q \]

\[ b \]

\[ a \]

\[ b \]

\[ q, r \]
Propositional Dynamic Logic (PDL)

Properties

- Satisfiability problem is EXPTIME-complete.
- Tree-like model property.

Fischer-Ladner closure

\[
\langle b^* \rangle r \\
\langle b \rangle \langle b^* \rangle r \\
\langle b^* \rangle r
\]
Propositional Dynamic Logic (PDL)

Properties

- Satisfiability problem is EXPTIME-complete.
- Tree-like model property.

Fischer-Ladner closure
Interleaving PDL

Syntax

\[ \alpha, \beta ::= a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \mid (\alpha \mid \beta) \quad \text{(programs)} \]

\[ \varphi, \psi ::= p \mid \bot \mid \varphi \rightarrow \psi \mid \langle \alpha \rangle \varphi \quad \text{(formulas)} \]

Semantics

\[ \mathcal{L}(\alpha \mid \beta) = \mathcal{L}(\alpha) \sqcup \mathcal{L}(\beta) \]

For instance:

\[ \langle a \mid b \rangle \varphi \leftrightarrow \langle (a ; b) \cup (b ; a) \rangle \varphi \]

Complexity

The satisfiability problem is 2EXPTIME-complete.
PDL with intersection

Syntax

\[ \alpha, \beta ::= \ a \mid (\alpha ; \beta) \mid (\alpha \cup \beta) \mid \alpha^* \mid \varphi? \mid (\alpha \cap \beta) \]  
(programs)

\[ \varphi, \psi ::= \ p \mid \bot \mid \varphi \rightarrow \psi \mid \langle \alpha \rangle \varphi \]  
(formulas)

Semantics

We have: \( \langle \alpha \cap \beta \rangle \varphi \rightarrow \langle \alpha \rangle \varphi \land \langle \beta \rangle \varphi \)

Complexity

The satisfiability problem is 2EXPTIME-complete.
Sometimes only parallel programs are executable.

\[
\langle a \parallel b \rangle \top \land [a] \perp \land [b] \perp
\]
PDL with separating Parallel composition (PPDL)

Syntax

\[ \alpha, \beta ::= a | (\alpha ; \beta) | (\alpha \cup \beta) | \alpha^* | \varphi? | (\alpha \parallel \beta) \quad \text{(programs)} \]

\[ \varphi, \psi ::= p | \bot | \varphi \rightarrow \psi | \langle \alpha \rangle \varphi \quad \text{(formulas)} \]

Semantics

A model is a tuple \( M = (W, R, \triangleleft, V) \) where:

- \((W, R, V)\) is a PDL model,
- \(\triangleleft\) is a ternary relation over \(W\).
PDL with deterministic separating Parallel composition ($\text{PPDL}^{\text{det}}$)

**Definition**
A model is $\triangleleft$-deterministic iff there is at most one way to merge any pair of states:

$$\text{if } w_1 \triangleleft (x, y) \text{ and } w_2 \triangleleft (x, y) \text{ then } w_1 = w_2$$

**Rationale**
- There is a partial binary operator $\bullet$ such that $w \triangleleft (x, y) \iff w = x \bullet y$.
- Usual constraint in modal logics with a binary modality (arrow logics, ambient logics, separation logic...).
The Petri net example
Adaptation of the Fischer-Ladner closure

\[ [\alpha \parallel \beta] \varphi \]

\[ [\alpha] \ldots \alpha \]

\[ [\beta] \ldots \beta \]
Adaptation of the Fischer-Ladner closure
Adaptation of the Fischer-Ladner closure

\[ [\alpha] L_1 \quad \alpha \quad L_1 \]

\[ [\alpha \parallel \beta] \varphi \]

\[ [\beta] R_1 \quad \beta \quad R_1 \]
Adaptation of the Fischer-Ladner closure

\[ [\alpha || \beta] \varphi \]

\[ [\alpha] L_1 \quad \alpha \quad L_1 \quad \varphi \]

\[ [\beta] R_1 \quad \beta \quad R_1 \]
The neat model property

\[ \alpha \parallel \beta \]

\[ \gamma \parallel \delta \]
Hintikka sets
Maximal consistent subsets of the Fischer-Ladner closure.

Plugs
Triples \((H_1, H_2, H_3)\) of Hintikka sets s.t. \(L_i \in H_2\) and \(R_i \in H_3\) for some \(i \in \{1, 2\}\).

Sockets
Sets of zero, one or two plugs.

States of the pseudo-models
Pairs \((H, S)\) where \(H\) is a Hintikka set and \(S\) is a socket.

\[(H_1, S_1) \triangleleft ((H_2, S_2), (H_3, S_3))\] iff \((H_1, H_2, H_3) \in S_2 = S_3\]
Conclusion

Theoretical result
The addition of deterministic separating parallel compositions to PDL does not increase the complexity of the satisfiability problem.

Future works
- Design an optimal and implementable decision procedure.
- Add commutativity and associativity.