An Optimal Tableau Decision Procedure for Converse-PDL

Linh Anh Nguyen and Andrzej Szałas
Institute of Informatics
University of Warsaw

KSE’09 – Hanoi, October 2009
Propositional Dynamic Logic (PDL) and its extension with the converse operator (CPDL) are useful for:

- reasoning about programs
- reasoning about structured knowledge:
  - the description logic $\mathcal{ALCI}_{reg}$ is a notational variant of CPDL
  - extensions of $\mathcal{ALCI}_{reg}$ are useful description logics
- artificial intelligence:
  - extending CPDL with regular inclusion axioms yields a powerful logic for expressing epistemic states of agents
Introduction and Motivations

Previous Work

- Fischer & Ladner, 1978:
  - The satisfiability problem of PDL is $\text{ExpTime}$-complete.

- Pratt, 1978:
  - the first practical $\text{ExpTime}$ decision procedure for PDL
  - The formulation is a bit too indirect.

- De Giacomo & Massacci, 2000:
  - a non-optimal $\text{NExpTime}$ decision procedure for CPDL
    - They also described how to transform the algorithm to an $\text{ExpTime}$ version. However, the description is informal and unclear, and the transformation has not been proved sound.

This Work

We give a novel tableau calculus and an optimal ($\text{ExpTime}$) tableau decision procedure based on the calculus for CPDL.
Outline

1. Propositional Dynamic Logic with Converse
   - Kripke Models
   - Syntax and Semantics
   - The Satisfiability Problem

2. A Tableau Calculus for CPDL
   - Tableau Rules
   - Tableaux
   - Soundness and Completeness

3. A Tableau Decision Procedure for CPDL
Propositional Dynamic Logic with Converse

Interpretation

A Kripke Model $\mathcal{M} = \langle \Delta^\mathcal{M}, \cdot^\mathcal{M} \rangle$, where

- $\Delta^\mathcal{M}$ is a set of states
- for each proposition $p \in \Phi_0$, $p^\mathcal{M}$ is a set of states
- for each atomic program $\sigma \in \Pi_0$, $\sigma^\mathcal{M}$ is a set of pairs (input_state, output_state)
Propositional Dynamic Logic with Converse

Formulas and Programs

\[ \varphi ::= \top | \bot | p | \neg \varphi | \varphi \land \varphi | \varphi \lor \varphi | \varphi \rightarrow \varphi | \langle \alpha \rangle \varphi | [\alpha] \varphi \]

\[ \alpha ::= \sigma | \alpha; \alpha | \alpha \cup \alpha | \alpha^* | \alpha^- | \varphi? \]

Semantics

- \( \varphi \) stands for a set of states, for example:
  - \( (\langle \alpha \rangle \varphi)^M \overset{\text{def}}{=} \{ x \in \Delta^M | \exists y (\alpha^M(x, y) \land \varphi^M(y)) \} \)
  - \( ([\alpha] \varphi)^M \overset{\text{def}}{=} \{ x \in \Delta^M | \forall y (\alpha^M(x, y) \rightarrow \varphi^M(y)) \} \)
- \( \alpha \) stands for a set of pairs (input state, output state):
  - \( \alpha; \beta \) stands for a sequential composition of \( \alpha \) and \( \beta \),
  - \( \alpha \cup \beta \) stands for set-theoretical union of \( \alpha \) and \( \beta \),
  - \( \alpha^* \) stands for the reflexive and transitive closure of \( \alpha \),
  - \( \alpha^- \) stands for the converse of \( \alpha \),
  - \( \varphi? \) stands for the test operator.
Propositional Dynamic Logic with Converse

Let $\mathcal{M}$ be a Kripke model and $X$, $\Gamma$ be sets of formulas.

- $X^\mathcal{M} \overset{\text{def}}{=} \bigcup_{\varphi \in X} \varphi^\mathcal{M}$
- $\mathcal{M}$ satisfies $X$ if $X^\mathcal{M} \neq \emptyset$
- $\mathcal{M}$ validates $\Gamma$ if $\Gamma^\mathcal{M} = \Delta^\mathcal{M}$

- $X$ is **satisfiable** w.r.t. the set $\Gamma$ of global assumptions if there exists a Kripke model that satisfies $X$ and validates $\Gamma$.

**Problem**

$X$ and $\Gamma$ are finite sets of formulas.

Is $X$ satisfiable w.r.t. the set $\Gamma$ of global assumptions?
A **tableau** is an “and-or” graph with global caching.  
A **node** of such a graph contains:
- a **label**, which is a set of formulas
- additional information.

An “and-or” graph is expanded using tableau rules.

A **tableau rule** is either an “and”-rule or an “or”-rule.
"Or"-Rules and "And"-Rules

"Or"-Rules

\[ \frac{Y}{Z_1 | \ldots | Z_k} \quad (k \geq 1) \]

- If \( Y \) is satisfiable w.r.t. \( \Gamma \) then so is some \( Z_i \).
- Expanding a node with label \( Y \) using the above "or"-rule makes the node become an "or"-node with \( k \) successors with label \( Z_1, \ldots, Z_k \), respectively.

In our calculus, "or"-rules are static rules.
"And"-Rules

\[
\begin{align*}
Y \\
Z_1 & \& \ldots & \& Z_k
\end{align*}
\]

or

\[
\begin{align*}
Y \\
& \& \{Z_i \text{ such that } \ldots \}
\end{align*}
\]

- If \( Y \) is satisfiable w.r.t. \( \Gamma \) then all \( Z_i \) are also satisfiable w.r.t. \( \Gamma \), possibly at different states.

- Expanding a node with label \( Y \) using the above "and"-rule makes the node become an "and"-node with \( k \) successors with label \( Z_1, \ldots, Z_k \), respectively.

The only "and"-rule of our calculus is a **transitional rule**.
Formulas are in negation-and-converse normal form (NCNF). We write $\overline{\varphi}$ for the NCNF of $\neg \varphi$.

\[
\begin{align*}
(\bot_0) & \quad \frac{Y, \bot}{\bot} \\
(\bot) & \quad \frac{Y, p, \neg p}{\bot} \\
(^\wedge) & \quad \frac{Y, \varphi \wedge \psi}{Y, \varphi, \psi} \\
(^\vee) & \quad \frac{Y, \varphi \vee \psi}{Y, \varphi \upharpoonright Y, \psi}
\end{align*}
\]
### Rules of Tableau Calculus $C_{\text{CPDL}}$ (2)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Premise</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>($\square;.$)</td>
<td>$Y, [\alpha; \beta] \varphi$</td>
<td>$Y, [\alpha][\beta] \varphi$</td>
</tr>
<tr>
<td>($\square \cup$)</td>
<td>$Y, [\alpha \cup \beta] \varphi$</td>
<td>$Y, [\alpha] \varphi, [\beta] \varphi$</td>
</tr>
<tr>
<td>($\square ?$)</td>
<td>$Y, [\psi?] \varphi$</td>
<td>$Y, \overline{\psi}</td>
</tr>
<tr>
<td>($\square *$)</td>
<td>$Y, [\alpha^*] \varphi$</td>
<td>$Y, \varphi, [\alpha][\alpha^*] \varphi$</td>
</tr>
<tr>
<td>($\diamond;.$)</td>
<td>$Y, \langle \alpha; \beta \rangle \varphi$</td>
<td>$Y, \langle \alpha \rangle \langle \beta \rangle \varphi$</td>
</tr>
<tr>
<td>($\diamond \cup$)</td>
<td>$Y, \langle \alpha \cup \beta \rangle \varphi$</td>
<td>$Y, \langle \alpha \rangle \varphi</td>
</tr>
<tr>
<td>($\diamond ?$)</td>
<td>$Y, \langle \psi? \rangle \varphi$</td>
<td>$Y, \psi, \varphi$</td>
</tr>
<tr>
<td>($\diamond *$)</td>
<td>$Y, \langle \alpha^* \rangle \varphi$</td>
<td>$Y, \varphi</td>
</tr>
</tbody>
</table>
We use
• ς to denote a program of the form σ or σ−, for σ ∈ Π₀
• an auxiliary modal operator □ς as a “blocked” of [ς].

For dealing with converse:

\[
\text{(cut)} \quad \frac{Y}{Y, \varphi | Y, □ς (ς−) \overline{\varphi}}
\]

if
• ϕ belongs to the closure of \(Y \cup \Gamma\)
• and some other conditions hold
The only "and"-rule / transitional rule is:

\[
Y \\
\&\{ (\{\varphi\} \cup \{\psi \text{ s.t. } [\varsigma]\psi \in Y \text{ or } \Box\varsigma\psi \in Y\}\cup \Gamma) \text{ s.t. } \langle\varsigma\rangle\varphi \in Y \}
\]

Example

An instance of this rule w.r.t. \(\Gamma = \{s\}\):

\[
\langle\sigma\rangle p, \langle\sigma\rangle q, [\sigma]r \\
p, r, s \& q, r, s
\]
Restrictions on Applicability of Rules of $\mathcal{C}_{\text{CPDL}}$

Priorities on Rules

- $(\bot_0), (\bot)$
- the other static rules
- the transitional rule

Another Restriction

Purpose:

In any sequence of applications of static rules a formula of the form $\varphi \land \psi$, $\varphi \lor \psi$, $[\alpha; \beta] \varphi$, $[\alpha \cup \beta] \varphi$, $[\psi?] \varphi$, or $[\alpha^*] \varphi$ is reduced at most once.

Technique:

Attach some additional information to each node so that the check can be done locally.
Tableaux, cont’d

Purpose: Check whether $X$ is satisfiable w.r.t. $\Gamma$.

A tableau for $(X, \Gamma)$ is an “and-or” graph such that

- The label of the initial node (the root) is $X \cup \Gamma$.
- The nodes are expanded using the tableau rules accordingly to the restrictions of applicability,
  using global caching (so that each node has unique contents).
Soundness and Completeness of $C_{CPDL}$

Marking of an “and-or” graph $G$

a subgraph $G'$ of $G$ such that:

- the root of $G$ is the root of $G'$,
- if $v$ is a node of $G'$ and is an “or”-node of $G$ then at least one edge $(v, w)$ of $G$ is an edge of $G'$
- if $v$ is a node of $G'$ and is an “and”-node of $G$ then every edge $(v, w)$ of $G$ is an edge of $G'$
- if $(v, w)$ is an edge of $G'$ then $v$ and $w$ are nodes of $G'$. 
Consistent marking of an “and-or” graph $G$

A marking $G'$ of $G$ such that:

- **local consistency**: $G'$ contains no node with label $\{\bot\}$
- **global consistency**: for every node $v$ of $G'$, every formula of the form $\langle \alpha \rangle \varphi$ of the label of $v$ has a $\Diamond$-realization in $G'$ (i.e., is fulfilled).
Theorem

Let $X$ and $\Gamma$ be finite sets of formulas in NCNF, and $G$ be an “and-or” graph for $(X, \Gamma)$. Then $X$ is satisfiable w.r.t. the set $\Gamma$ of global assumptions iff $G$ has a consistent marking.
Purpose: Check whether $X$ is satisfiable w.r.t. $\Gamma$.

The Basic Algorithm

- Construct an “and-or” graph $G$ for $(X, \Gamma)$.
- Try to construct a consistent marking $G'$ of $G$ by starting from $G$ and repeatedly eliminating nodes that violate the local consistency property or the global consistency property.

To find nodes that violate the global consistency property:
- construct the graph of traces of $G'$;
- analyze “productiveness” in this graph of traces.
- If such a $G'$ exists then answer “yes”, else answer “no”.
Properties

- The basic algorithm runs in exponential time (optimal).
- Optimizations can be incorporated.
Comparison

Our decision procedure for CPDL substantially differs from the procedure given by De Giacomo and Massacci:

- Our decision procedure has optimal complexity $\text{ExpTime}$, while the formal decision procedure of [De Giacomo and Massacci] has non-optimal complexity $\text{NExpTime}$.
- Our decision procedure uses traditional (unlabeled) tableau rules, while the decision procedure of [De Giacomo and Massacci] uses labeled (prefixed) tableau rules.
- Our cut rule is of the kind of “guessing the future”, while [De Giacomo and Massacci] uses a cut rule of the kind “look behind”.
- The applicability of our cut rule is much more restricted than that of the cut rule of [De Giacomo and Massacci].