Relation-Changing Logics as Fragments of Hybrid Logics

Carlos Areces\(^1\), Raul Fervari\(^1\), Guillaume Hoffmann\(^1\), Mauricio Martel\(^2\)

\(^1\) FaMAF, Universidad Nacional de Córdoba & CONICET, Argentina
\(^2\) Fachbereich Mathematik und Informatik, Universität Bremen, Germany

GandALF 2016 - Catania, Italy
Modal logics from a semantic perspective

- Modal logics are known to describe models.
Modal logics from a semantic perspective

- Modal logics are known to describe models.
- Choose the right paintbrush:
  - $\Diamond \varphi$, $\Diamond \neg \varphi$
  - $E \varphi$
  - $\Diamond \geq n \varphi$
  - $\Diamond^* \varphi$
  - $\ldots$

Now, what about operators that can modify models?

- Change the domain of the model.
- Change the properties of the elements of the domain while we are evaluating a formula.
- Change the accessibility relation of a model while we are evaluating a formula.
Modal logics from a semantic perspective

- Modal logics are known to describe models.
- Choose the right paintbrush:
  - $\Diamond \varphi, \Diamond \neg \varphi$
  - $E \varphi$
  - $\Diamond \geq n \varphi$
  - $\Diamond^* \varphi$
  - $\ldots$
- Now, what about operators that can modify models?
Modal logics from a semantic perspective

- Modal logics are known to **describe** models.
- Choose the right paintbrush:
  - $\diamond \varphi$, $\diamond \neg \varphi$
  - $E\varphi$
  - $\diamond \geq n \varphi$
  - $\diamond \ast \varphi$
  - $\ldots$
- Now, what about operators that can **modify** models?
  - Change the domain of the model.
Modal logics from a semantic perspective

- Modal logics are known to describe models.
- Choose the right paintbrush:
  - $\Diamond \varphi$, $\Diamond \neg \varphi$
  - $E\varphi$
  - $\Diamond \geq n\varphi$
  - $\Diamond^* \varphi$
  - ...

- Now, what about operators that can modify models?
  - Change the domain of the model.
  - Change the properties of the elements of the domain while we are evaluating a formula.
Modal logics from a semantic perspective

- Modal logics are known to describe models.
- Choose the right paintbrush:
  - \( \Diamond \varphi, \Diamond \neg \varphi \)
  - \( E \varphi \)
  - \( \Diamond \geq n \varphi \)
  - \( \Diamond^* \varphi \)
  - \( \ldots \)
- Now, what about operators that can modify models?
  - Change the domain of the model.
  - Change the properties of the elements of the domain while we are evaluating a formula.
  - Change the accessibility relation of a model while we are evaluating a formula.
Logics that can change the model

What about a **swapping** modal operator?

$$\langle \text{sw} \rangle \Diamond \top$$

What happens when you add that to the basic modal logic?
Logics that can change the model

What about a **swapping** modal operator?

\[ ⟨\text{sw}⟩ \Box \top \]

What about

- an edge-deleting modality?
- an edge-adding modality?
Sabotage Modal Logic [van Benthem 05]

\[ M, w \models \langle \text{gsb} \rangle \varphi \text{ iff } \exists \text{ pair } (u, v) \text{ of } M \text{ such that } M_{\{(u,v)\}}, w \models \varphi, \]

where \( M_{\{(u,v)\}} \) is \( M \) without the edge \((u, v)\).

**Note:** \((u, v)\) can be anywhere in the model.
Sabotage Modal Logic [van Benthem 05]

\[ M, w \models \langle \text{gsb} \rangle \varphi \text{ iff } \exists \text{ pair } (u, v) \text{ of } M \text{ such that } M_{\{ (u,v) \}} \setminus \{ (u,v) \}, w \models \varphi, \]

where \( M_{\{ (u,v) \}} \) is \( M \) without the edge \( (u, v) \).

**Note:** \( (u, v) \) can be anywhere in the model.

We are interested in operators that can modify the accessibility relation of a model.
Relation-Changing Logics

Remember the Basic Modal Logic ($\mathcal{ML}$)
- Syntax: propositional language $+$ a modal operator $\Diamond$.
- Semantics of $\Diamond \varphi$: traverse some edge, then evaluate $\varphi$. 

Now add new dynamic operators (Sabotage, Bridge, and Swap):
- $\langle \text{sb} \rangle \varphi$: traverse some edge, delete it, then evaluate $\varphi$.
- $\langle \text{br} \rangle \varphi$: add a new edge, traverse it, then evaluate $\varphi$.
- $\langle \text{sw} \rangle \varphi$: traverse some edge, turn it around, then evaluate $\varphi$.
- $\langle \text{gsb} \rangle \varphi$: delete some edge anywhere, then evaluate $\varphi$.
- $\langle \text{gbr} \rangle \varphi$: add a new edge anywhere, then evaluate $\varphi$.
- $\langle \text{gsw} \rangle \varphi$: swap an edge anywhere, then evaluate $\varphi$. 

We call this family of logics Relation-Changing Logics.
Relation-Changing Logics

Remember the Basic Modal Logic ($\mathcal{ML}$)

• Syntax: propositional language + a modal operator $\diamond$.

• Semantics of $\diamond \varphi$: traverse some edge, then evaluate $\varphi$.

Now add new dynamic operators (Sabotage, Bridge, and Swap):
Relation-Changing Logics

Remember the Basic Modal Logic ($\mathcal{ML}$)

- Syntax: propositional language + a modal operator $\Diamond$.
- Semantics of $\Diamond \varphi$: traverse some edge, then evaluate $\varphi$.

Now add new dynamic operators (Sabotage, Bridge, and Swap):

- $\langle \text{sb} \rangle \varphi$: traverse some edge, delete it, then evaluate $\varphi$.
- $\langle \text{br} \rangle \varphi$: add a new edge, traverse it, then evaluate $\varphi$.
- $\langle \text{sw} \rangle \varphi$: traverse some edge, turn it around, then evaluate $\varphi$. 
Relation-Changing Logics

Remember the Basic Modal Logic ($\mathcal{ML}$)

- Syntax: propositional language + a modal operator $\Diamond$.
- Semantics of $\Diamond \varphi$: traverse some edge, then evaluate $\varphi$.

Now add new dynamic operators (Sabotage, Bridge, and Swap):

- $\langle \text{sb} \rangle \varphi$: traverse some edge, delete it, then evaluate $\varphi$.
- $\langle \text{br} \rangle \varphi$: add a new edge, traverse it, then evaluate $\varphi$.
- $\langle \text{sw} \rangle \varphi$: traverse some edge, turn it around, then evaluate $\varphi$.
- $\langle \text{gsb} \rangle \varphi$: delete some edge anywhere, then evaluate $\varphi$.
- $\langle \text{gbr} \rangle \varphi$: add a new edge anywhere, then evaluate $\varphi$.
- $\langle \text{gsw} \rangle \varphi$: swap an edge anywhere, then evaluate $\varphi$. 

We call this family of logics Relation-Changing Logics.
Relation-Changing Logics

Remember the Basic Modal Logic ($\mathcal{ML}$)

- Syntax: propositional language + a modal operator $\Diamond$.
- Semantics of $\Diamond \varphi$: traverse some edge, then evaluate $\varphi$.

Now add new dynamic operators (Sabotage, Bridge, and Swap):

- $\langle \text{sb} \rangle \varphi$: traverse some edge, delete it, then evaluate $\varphi$.
- $\langle \text{br} \rangle \varphi$: add a new edge, traverse it, then evaluate $\varphi$.
- $\langle \text{sw} \rangle \varphi$: traverse some edge, turn it around, then evaluate $\varphi$.
- $\langle \text{gsb} \rangle \varphi$: delete some edge anywhere, then evaluate $\varphi$.
- $\langle \text{gbr} \rangle \varphi$: add a new edge anywhere, then evaluate $\varphi$.
- $\langle \text{gsw} \rangle \varphi$: swap an edge anywhere, then evaluate $\varphi$.

We call this family of logics Relation-Changing Logics.
Some results about RCL

- Lack of tree-model property and finite model property (more expressivity than $\mathcal{ML}$).

Model checking is $\text{PSpace}$-complete (via QBF reduction).

The satisfiability problem is undecidable (via spy points and memory logic reduction).

Sound and complete (but non-terminating) tableaux methods; Standard translations into $\text{FOL}$.

We now provide translations into hybrid logics.
Some results about RCL

- Lack of tree-model property and finite model property (more expressivity than $\mathcal{ML}$).
- Incomparable among them in expressive power (even between local and global cases of same modification).
Some results about RCL

- Lack of tree-model property and finite model property (more expressivity than $\mathcal{ML}$).
- Incomparable among them in expressive power (even between local and global cases of same modification).
- Model checking is $\text{PSPACE}$-complete (via QBF reduction).
Some results about RCL

- Lack of tree-model property and finite model property (more expressivity than $\mathcal{ML}$).
- Incomparable among them in expressive power (even between local and global cases of same modification).
- Model checking is $\text{PSPACE}$-complete (via QBF reduction).
- The satisfiability problem is undecidable (via spy points and memory logic reduction).
Some results about RCL

- Lack of tree-model property and finite model property (more expressivity than $\mathcal{ML}$).
- Incomparable among them in expressive power (even between local and global cases of same modification).
- Model checking is $\text{PSPACE}$-complete (via QBF reduction).
- The satisfiability problem is undecidable (via spy points and memory logic reduction).
- Sound and complete (but non-terminating) tableaux methods; Standard translations into $\mathcal{FOL}$. 

We now provide translations into hybrid logics.
Some results about RCL

- Lack of tree-model property and finite model property (more expressivity than $\mathcal{ML}$).
- Incomparable among them in expressive power (even between local and global cases of same modification).
- Model checking is $\text{PSPACE}$-complete (via QBF reduction).
- The satisfiability problem is undecidable (via spy points and memory logic reduction).
- Sound and complete (but non-terminating) tableaux methods; Standard translations into $\mathcal{FOL}$.

We now provide translations into hybrid logics.
Hybrid Logics

- The basic hybrid logic $\mathcal{HL}$ is obtained by adding a set NOM of nominals to $\mathcal{ML}$. For $n \in \text{NOM}$, its valuation is a singleton set $V(n) = \{w\}$, for some state $w$. 

Mauricio Martel Relation-Changing Logics as Fragments of Hybrid Logics
Hybrid Logics

- The basic hybrid logic $\mathcal{H}L$ is obtained by adding a set $\text{NOM}$ of nominals to $\mathcal{ML}$. For $n \in \text{NOM}$, its valuation is a singleton set $V(n) = \{w\}$, for some state $w$.
- We have a satisfaction operator $n : \varphi$ with the usual semantics:

$$\mathcal{M}, w \models n : \varphi \quad \text{iff} \quad \mathcal{M}, v \models \varphi, \text{ where } V(n) = \{v\}.$$
Hybrid Logics

- The basic hybrid logic $\mathcal{HL}$ is obtained by adding a set $\text{NOM}$ of nominals to $\mathcal{ML}$. For $n \in \text{NOM}$, its valuation is a singleton set $V(n) = \{w\}$, for some state $w$.
- We have a satisfaction operator $n : \varphi$ with the usual semantics:
  \[ M, w \models n : \varphi \iff M, v \models \varphi, \text{ where } V(n) = \{v\}. \]
- And we also consider the down-arrow binder operator $\downarrow$:
  \[ \langle W, R, V \rangle, w \models \downarrow n.\varphi \iff \langle W, R, V^n \rangle, w \models \varphi, \]
  where $V^n(w)(n) = \{w\}$ and $V^n(w)(m) = V(m)$, when $n \neq m$. 

Mauricio Martel
Relation-Changing Logics as Fragments of Hybrid Logics
Translations to Hybrid Logics

- The translations are parametrized over a set of pair of nominals $S \subseteq \text{NOM} \times \text{NOM}$ that simulates the modification of edges.

Sabotage to Hybrid Logic

We define the translation $(\cdot)'_S$ from formulas of $\text{ML}(\langle \text{sb} \rangle)$ to formulas of $\text{HL}(:\downarrow)$ as:

$(\Diamond \phi)'_S = \downarrow n. \Diamond (\bigwedge xy \in S \neg \left( y \land n : x \right) \land (\phi)'_S)$

$(\langle \text{sb} \rangle \phi)'_S = \downarrow n. \Diamond (\bigwedge xy \in S \neg \left( y \land n : x \right) \land \downarrow m. (\phi)'_S \cup nm)$

And for $\text{ML}(\langle \text{gsb} \rangle)$ we translate into $\text{HL}(E, \downarrow)$:

$(\langle \text{gsb} \rangle \phi)'_S = \downarrow k. E \downarrow n. \Diamond (\neg (\bigwedge xy \in S \neg \left( y \land n : x \right) \land \downarrow m. k : (\phi)'_S \cup nm)$

The translations for Bridge and Swap follow similar ideas.
Translations to Hybrid Logics

- The translations are parametrized over a set of pair of nominals $S \subseteq \text{NOM} \times \text{NOM}$ that simulates the modification of edges.

Sabotage to Hybrid Logic

We define the translation $(\varphi)_S'$ from formulas of $\mathcal{ML}(\langle \text{sb} \rangle)$ to formulas of $\mathcal{HL}(:, \downarrow)$ as:

$$
(\Diamond \varphi)_S' = \downarrow n. (\bigwedge_{xy \in S} \neg (y \land n:x) \land (\varphi)_S')
$$

$$
(\langle \text{sb} \rangle \varphi)_S' = \downarrow n. (\bigwedge_{xy \in S} \neg (y \land n:x) \land \downarrow m. (\varphi)'_{S \cup nm})
$$
Translations to Hybrid Logics

- The translations are parametrized over a set of pair of nominals $S \subseteq \text{NOM} \times \text{NOM}$ that simulates the modification of edges.

Sabotage to Hybrid Logic

We define the translation $(\text{'}_S)$ from formulas of $\mathcal{ML}(\langle \text{sb} \rangle)$ to formulas of $\mathcal{HL}(\cdot, \downarrow)$ as:

\[
(\Diamond \varphi)'_S = \downarrow n. (\bigwedge_{xy \in S} \neg(y \land n:x) \land (\varphi)'_S)
\]

\[
(\langle \text{sb} \rangle \varphi)'_S = \downarrow n. (\bigwedge_{xy \in S} \neg(y \land n:x) \land \downarrow m.(\varphi)'_{S \cup nm})
\]

And for $\mathcal{ML}(\langle \text{gsb} \rangle)$ we translate into $\mathcal{HL}(E, \downarrow)$:

\[
(\langle \text{gsb} \rangle \varphi)'_S = \downarrow k. E \downarrow n. (\bigwedge_{xy \in S} \neg(y \land n:x) \land \downarrow m.k:(\varphi)'_{S \cup nm})
\]
Translations to Hybrid Logics

- The translations are parametrized over a set of pair of nominals \( S \subseteq \text{NOM} \times \text{NOM} \) that simulates the modification of edges.

Sabotage to Hybrid Logic

We define the translation \((\cdot)'_S\) from formulas of \(\mathcal{ML}(\langle sb \rangle)\) to formulas of \(\mathcal{HL}(\langle \cdot \rangle, \downarrow)\) as:

\[
(\Box \varphi)'_S = \downarrow n.\Diamond (\bigwedge_{xy \in S} \neg (y \land n:x) \land (\varphi)'_S)
\]

\[
(\langle sb \rangle \varphi)'_S = \downarrow n.\Diamond (\bigwedge_{xy \in S} \neg (y \land n:x) \land \downarrow m.(\varphi)'_{S \cup nm})
\]

And for \(\mathcal{ML}(\langle gsb \rangle)\) we translate into \(\mathcal{HL}(E, \downarrow)\):

\[
(\langle gsb \rangle \varphi)'_S = \downarrow k.E\downarrow n.\Diamond (\bigwedge_{xy \in S} \neg (y \land n:x) \land \downarrow m.k:(\varphi)'_{S \cup nm})
\]

The translations for Bridge and Swap follow similar ideas.
Comparing Expressive Power

Theorem

\[ \mathcal{ML}(\Diamond_1) < \mathcal{HL}(:, \downarrow), \text{ for } \Diamond_1 \in \{\langle sb\rangle, \langle sw\rangle\}. \]
\[ \mathcal{ML}(\Diamond_2) < \mathcal{HL}(E, \downarrow), \text{ for } \Diamond_2 \in \{\langle gsb\rangle, \langle gsw\rangle, \langle br\rangle, \langle gbr\rangle\}. \]
Comparing Expressive Power

Theorem

$\mathcal{ML}(\Diamond_1) < \mathcal{HL}(:, \downarrow)$, for $\Diamond_1 \in \{\langle sb \rangle, \langle sw \rangle\}$.

$\mathcal{ML}(\Diamond_2) < \mathcal{HL}(E, \downarrow)$, for $\Diamond_2 \in \{\langle gsb \rangle, \langle gsw \rangle, \langle br \rangle, \langle gbr \rangle\}$.

Proof.

- To prove that $\mathcal{ML}(\Diamond_1) < \mathcal{HL}(:, \downarrow)$ it suffices to find two $\Diamond_1$-bisimilar models distinguishable by $\mathcal{HL}(:, \downarrow)$.
- To prove that $\mathcal{ML}(\Diamond_2) < \mathcal{HL}(E, \downarrow)$ it suffices to find two $\Diamond_2$-bisimilar models distinguishable by $\mathcal{HL}(E, \downarrow)$.
Comparing Expressive Power

<table>
<thead>
<tr>
<th>$\mathcal{M}, w$</th>
<th>$\mathcal{M}', w'$</th>
<th>Bisimilar for</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="" alt="Diagram 1" /></td>
<td><img src="" alt="Diagram 2" /></td>
<td>$\mathcal{ML}(\langle \text{sw} \rangle)$, $\mathcal{ML}(\langle \text{br} \rangle)$, $\mathcal{ML}(\langle \text{gsw} \rangle)$, $\mathcal{ML}(\langle \text{gbr} \rangle)$</td>
</tr>
<tr>
<td><img src="" alt="Diagram 3" /></td>
<td><img src="" alt="Diagram 4" /></td>
<td>$\mathcal{ML}(\langle \text{sb} \rangle)$, $\mathcal{ML}(\langle \text{gsb} \rangle)$</td>
</tr>
</tbody>
</table>

The formula $\downarrow n. \Box n$ can distinguish the models in the first row.

The formula $\downarrow n. \Diamond \downarrow m. n: \Diamond \Diamond m$ can distinguish the models in the second row.
Decidable Fragments (Semantic Restrictions)

The hybrid logic fragments considered in the translations are known to be Decidable over the indicated classes:

- $\text{HL} (\wedge, \downarrow)$ over models with a single relation of bounded width.
- $\text{HL} (E, \downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\text{HL} (E, \downarrow)$ over models with a single, transitive tree relation.
- $\text{HL} (E, \downarrow)$ over models with a single, $S_5$ relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\text{ML} (\langle \text{sb} \rangle)$ and $\text{ML} (\langle \text{sw} \rangle)$ over models of bounded width is Decidable.
2. The satisfiability problem for all relation-changing logics over linear, transitive trees, and $S_5$ frames is Decidable.
Decidable Fragments (Semantic Restrictions)

The hybrid logic fragments considered in the translations are known to be decidable over the indicated classes:

- $\mathcal{H}L(\cdot, \downarrow)$ over models with a single relation of bounded width.
- $\mathcal{H}L(E, \downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{H}L(E, \downarrow)$ over models with a single, transitive tree relation.
- $\mathcal{H}L(E, \downarrow)$ over models with a single, $S5$ relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $ML(\langle sb \rangle)$ and $ML(\langle sw \rangle)$ over models of bounded width is decidable.
2. The satisfiability problem for all relation-changing logics over linear, transitive trees, and $S5$ frames is decidable.
Decidable Fragments (Semantic Restrictions)

The hybrid logic fragments considered in the translations are known to be decidable over the indicated classes:

- $\mathcal{HL}(,\downarrow)$ over models with a single relation of bounded width.
- $\mathcal{HL}(E,\downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{HL}(E,\downarrow)$ over models with a single, transitive tree relation.
- $\mathcal{HL}(E,\downarrow)$ over models with a single, $S_5$ relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle sb \rangle)$ and $\mathcal{ML}(\langle sw \rangle)$ over models of bounded width is decidable.
Decidable Fragments (Semantic Restrictions)

The hybrid logic fragments considered in the translations are known to be **decidable** over the indicated classes:

- $\mathcal{H}L(\cdot, \downarrow)$ over models with a single relation of **bounded width**.
- $\mathcal{H}L(E, \downarrow)$ over **linear frames** (i.e., irreflexive, transitive, and trichotomous frames).

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{M}L(\langle sb \rangle)$ and $\mathcal{M}L(\langle sw \rangle)$ over models of **bounded width** is decidable.
Decidable Fragments (Semantic Restrictions)

The hybrid logic fragments considered in the translations are known to be decidable over the indicated classes:

- $\mathcal{HL}(\mathord:, \downarrow)$ over models with a single relation of bounded width.
- $\mathcal{HL}(E, \downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{HL}(E, \downarrow)$ over models with a single, transitive tree relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle sb \rangle)$ and $\mathcal{ML}(\langle sw \rangle)$ over models of bounded width is decidable.
Decidable Fragments (Semantic Restrictions)

The hybrid logic fragments considered in the translations are known to be decidable over the indicated classes:

- $\mathcal{HL}(:, \downarrow)$ over models with a single relation of bounded width.
- $\mathcal{HL}(E, \downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{HL}(E, \downarrow)$ over models with a single, transitive tree relation.
- $\mathcal{HL}(E, \downarrow)$ over models with a single, $S5$ relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle sb \rangle)$ and $\mathcal{ML}(\langle sw \rangle)$ over models of bounded width is decidable.
Decidable Fragments (Semantic Restrictions)

The hybrid logic fragments considered in the translations are known to be decidable over the indicated classes:

- $\mathcal{HL}(:, \downarrow)$ over models with a single relation of bounded width.
- $\mathcal{HL}(E, \downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{HL}(E, \downarrow)$ over models with a single, transitive tree relation.
- $\mathcal{HL}(E, \downarrow)$ over models with a single, $S5$ relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle sb \rangle)$ and $\mathcal{ML}(\langle sw \rangle)$ over models of bounded width is decidable.
2. The satisfiability problem for all relation-changing logics over linear, transitive trees, and $S5$ frames is decidable.
Decidable Fragments (Syntactic Restrictions)

$\mathcal{H}\mathcal{L}(\,;\,\downarrow) \setminus \Box\downarrow\Box$ is the fragment obtained by removing formulas that contain a nesting of $\Box$, $\downarrow$, and again $\Box$. This fragment is known to be decidable [B. ten Cate & M. Franceschet 05].
Decidable Fragments (Syntactic Restrictions)

$\mathcal{HL}(;,,\downarrow) \setminus \square\downarrow\square$ is the fragment obtained by removing formulas that contain a nesting of $\square$, $\downarrow$, and again $\square$. This fragment is known to be decidable [B. ten Cate & M. Franceschet 05].

Let $\diamond \in \{\langle\text{sb}\rangle, \langle\text{sw}\rangle\}$ and $\blacksquare \in \{[\text{sb}], [\text{sw}]\}$. The following patterns are produced by the translations:

<table>
<thead>
<tr>
<th>RC Pattern</th>
<th>Hybrid Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\square$</td>
<td>$\square$</td>
</tr>
<tr>
<td>$\blacksquare$</td>
<td>$\downarrow\square\downarrow$</td>
</tr>
<tr>
<td>$\diamond$</td>
<td>$\downarrow$</td>
</tr>
</tbody>
</table>
Decidable Fragments (Syntactic Restrictions)

$\mathcal{HL}(\cdot, \downarrow) \setminus \Box \downarrow \Box$ is the fragment obtained by removing formulas that contain a nesting of $\Box$, $\downarrow$, and again $\Box$. This fragment is known to be decidable [B. ten Cate & M. Franceschet 05].

Let $\diamond \in \{\langle sb \rangle, \langle sw \rangle\}$ and $\blacksquare \in \{[sb], [sw]\}$. The following patterns are produced by the translations:

<table>
<thead>
<tr>
<th>RC Pattern</th>
<th>Hybrid Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Box$</td>
<td>$\Box$</td>
</tr>
<tr>
<td>$\blacksquare$</td>
<td>$\downarrow \Box \downarrow$</td>
</tr>
<tr>
<td>$\diamond$</td>
<td>$\downarrow$</td>
</tr>
</tbody>
</table>

The following fragments are decidable on the class of all models:

1. $\mathcal{ML}(\langle sb \rangle) \setminus \{\blacksquare \blacksquare, \blacksquare \Box, \Box \blacksquare, \Box \diamond \Box\}$
2. $\mathcal{ML}(\langle sw \rangle) \setminus \{\blacksquare \blacksquare, \blacksquare \Box, \Box \blacksquare, \Box \diamond \Box\}$

where $\blacksquare$ is either $\Box$ or $\blacksquare$. 
We implemented the translations as a new feature of the tableaux-based theorem prover HTab [G. Hoffmann & C. Areces 09].
Implementation in HTab

- We implemented the translations as a new feature of the tableaux-based theorem prover HTab [G. Hoffmann & C. Areces 09].
- HTab originally handles the hybrid logic $\mathcal{HL}(E, \downarrow)$.
Implementation in HTab

- We implemented the translations as a new feature of the tableaux-based theorem prover HTab [G. Hoffmann & C. Areces 09].
- HTab originally handles the hybrid logic $\mathcal{HL}(E, \downarrow)$.
- We added a flag `--translate` that interprets the input formula as a relation-changing one. It first translates it to an $\mathcal{HL}(E, \downarrow)$-formula and then runs its internal hybrid tableaux calculus on the translation.
Implementation in HTab

- We implemented the translations as a new feature of the tableaux-based theorem prover HTab [G. Hoffmann & C. Areces 09].
- HTab originally handles the hybrid logic $\mathcal{HL}(E, \downarrow)$.
- We added a flag --translate that interprets the input formula as a relation-changing one. It first translates it to an $\mathcal{HL}(E, \downarrow)$-formula and then runs its internal hybrid tableaux calculus on the translation.
- This implementation is useful to check the correctness of the translations and for checking that RC formulas build models in the expected way.
Conclusions

- Relation-changing logics are very expressive:
  - Model checking is $\text{PSPACE}$-complete.
  - Satisfiability is undecidable.

- We defined translations into hybrid logics:
  - They are useful to analyze expressive power.
  - They allow us to identify some decidable fragments.
  - We provided an implementation in HTab.

- Further work using hybrid logic techniques:
  - Find axiomatizations.
  - Compute interpolants.

Thanks!
Conclusions

• Relation-changing logics are very expressive:
  • Model checking is \text{PSPACE}-complete.
  • Satisfiability is undecidable.
• We defined translations into hybrid logics:
Conclusions

• Relation-changing logics are very expressive:
  • Model checking is \textit{PSPACE}-complete.
  • Satisfiability is undecidable.

• We defined translations into hybrid logics:
  • They are useful to analyze \textit{expressive power}.
Conclusions

- Relation-changing logics are very expressive:
  - Model checking is $\text{PSPACE}$-complete.
  - Satisfiability is undecidable.
- We defined translations into hybrid logics:
  - They are useful to analyze expressive power.
  - They allow us to identify some decidable fragments.

Thanks!
Conclusions

- Relation-changing logics are very expressive:
  - Model checking is \( \text{PSPACE} \)-complete.
  - Satisfiability is undecidable.
- We defined translations into hybrid logics:
  - They are useful to analyze expressive power.
  - They allow us to identify some decidable fragments.
  - We provided an implementation in HTab.
Conclusions

- Relation-changing logics are very expressive:
  - Model checking is $\text{PSPACE}$-complete.
  - Satisfiability is undecidable.
- We defined translations into hybrid logics:
  - They are useful to analyze expressive power.
  - They allow us to identify some decidable fragments.
  - We provided and implementation in HTab.
- Further work using hybrid logic techniques:
Conclusions

• Relation-changing logics are very expressive:
  • Model checking is $\text{PSPACE}$-complete.
  • Satisfiability is undecidable.

• We defined translations into hybrid logics:
  • They are useful to analyze expressive power.
  • They allow us to identify some decidable fragments.
  • We provided and implementation in HTab.

• Further work using hybrid logic techniques:
  • Find axiomatizations.
Conclusions

- Relation-changing logics are very expressive:
  - Model checking is \( \text{PSPACE} \)-complete.
  - Satisfiability is undecidable.
- We defined translations into hybrid logics:
  - They are useful to analyze expressive power.
  - They allow us to identify some decidable fragments.
  - We provided and implementation in HTab.
- Further work using hybrid logic techniques:
  - Find axiomatizations.
  - Compute interpolants.
Conclusions

• Relation-changing logics are very expressive:
  • Model checking is $\text{PSPACE}$-complete.
  • Satisfiability is undecidable.

• We defined translations into hybrid logics:
  • They are useful to analyze expressive power.
  • They allow us to identify some decidable fragments.
  • We provided and implementation in HTab.

• Further work using hybrid logic techniques:
  • Find axiomatizations.
  • Compute interpolants.

Thanks!