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:
  - $\Diamond\varphi$, $\Diamond^{-}\varphi$
  - $\mathsf{E}\varphi$
  - $\Diamond_{\geq n}\varphi$
  - $\Diamond^{*}\varphi$
  - ...

Modal logics from a semantic perspective

- Modal logics are known to describe models.
- Choose the right paintbrush:
  - $\Diamond\varphi$, $\Diamond^{-}\varphi$
  - $\mathsf{E}\varphi$
  - $\Diamond_{\geq n}\varphi$
  - $\Diamond^{*}\varphi$
  - ...
- 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:
  - $\Diamond\varphi$, $\Diamond^{-}\varphi$
  - $\mathsf{E}\varphi$
  - $\Diamond_{\geq n}\varphi$
  - $\Diamond^{*}\varphi$
  - ...
- 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:
  - $\Diamond\varphi$, $\Diamond^{-}\varphi$
  - $\mathsf{E}\varphi$
  - $\Diamond_{\geq n}\varphi$
  - $\Diamond^{*}\varphi$
  - ...
- 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:
  - $\Diamond\varphi$, $\Diamond^{-}\varphi$
  - $\mathsf{E}\varphi$
  - $\Diamond_{\geq n}\varphi$
  - $\Diamond^{*}\varphi$
  - ...
- 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.
  - Change the accessibility relation of a model while we are evaluating a formula.

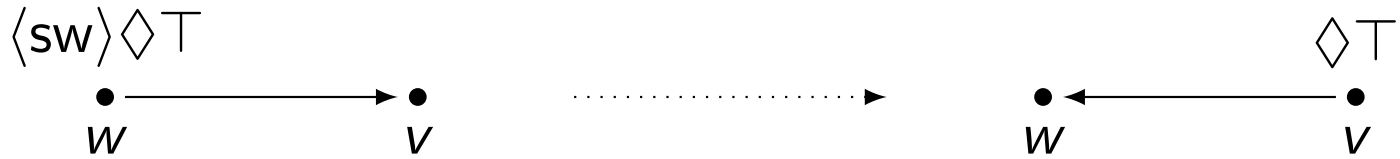
Logics that can change the model

What about a swapping modal operator?

What happens when you add that to the basic modal logic?

Logics that can change the model

What about a swapping modal operator?



What about

- an edge-deleting modality?
- an edge-adding modality?

Sabotage Modal Logic [van Benthem 05]

$\mathcal{M}, w \models \langle \text{gsb} \rangle \varphi$ iff $\exists$ pair (u, v) of $\mathcal{M}$ such that $\mathcal{M}^{-}_{\{(u,v)\}}, w \models \varphi$,

where $\mathcal{M}^{-}_{\{(u,v)\}}$ is $\mathcal{M}$ without the edge (u, v).

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

Sabotage Modal Logic [van Benthem 05]

$\mathcal{M}, w \models \langle \text{gsb} \rangle \varphi$ iff $\exists$ pair (u, v) of $\mathcal{M}$ such that $\mathcal{M}^{-}_{\{(u,v)\}}, w \models \varphi$,

where $\mathcal{M}^{-}_{\{(u,v)\}}$ is $\mathcal{M}$ 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 φ.

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 φ.

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 φ.

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

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

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 φ.

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

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

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 φ.

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

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

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 $\mathcal{ML}$).

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 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 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 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 $\mathcal{FOL}$.

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 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.

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.
- 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 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:

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

- And we also consider the down-arrow binder operator $\downarrow$:

$$\langle W, R, V \rangle, w \models \downarrow n.\varphi \quad \text{iff} \quad \langle W, R, V_n^w \rangle, w \models \varphi,$$

where $V_n^w(n) = \{w\}$ and $V_n^w(m) = V(m)$, when $n \neq m$.

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.

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

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

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

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

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

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

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

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

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

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

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

The translations for Bridge and Swap follow similar ideas.

Comparing Expressive Power

Theorem

$\mathcal{ML}(\blacklozenge_1) < \mathcal{HL}(:, \downarrow)$, *for* $\blacklozenge_1 \in \{\langle \mathsf{sb} \rangle, \langle \mathsf{sw} \rangle\}$.

$\mathcal{ML}(\blacklozenge_2) < \mathcal{HL}(\mathsf{E}, \downarrow)$, *for* $\blacklozenge_2 \in \{\langle \mathsf{gsb} \rangle, \langle \mathsf{gsw} \rangle, \langle \mathsf{br} \rangle, \langle \mathsf{gbr} \rangle\}$.

Comparing Expressive Power

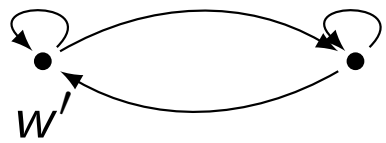
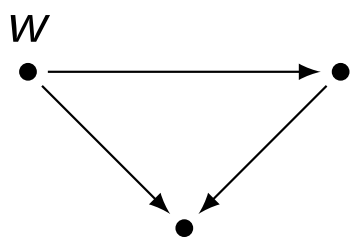
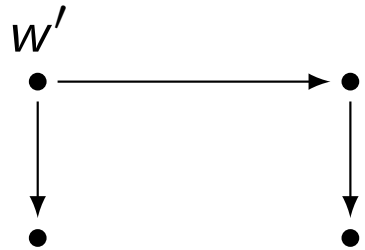
Theorem

$\mathcal{ML}(\blacklozenge_1) < \mathcal{HL}(:, \downarrow)$, *for* $\blacklozenge_1 \in \{\langle \mathsf{sb} \rangle, \langle \mathsf{sw} \rangle\}$.
$\mathcal{ML}(\blacklozenge_2) < \mathcal{HL}(\mathsf{E}, \downarrow)$, *for* $\blacklozenge_2 \in \{\langle \mathsf{gsb} \rangle, \langle \mathsf{gsw} \rangle, \langle \mathsf{br} \rangle, \langle \mathsf{gbr} \rangle\}$.

Proof.

- To prove that $\mathcal{ML}(\blacklozenge_1) < \mathcal{HL}(:, \downarrow)$ it suffices to find two $\blacklozenge_1$-bisimilar models distinguishable by $\mathcal{HL}(:, \downarrow)$
- To prove that $\mathcal{ML}(\blacklozenge_2) < \mathcal{HL}(\mathsf{E}, \downarrow)$ it suffices to find two $\blacklozenge_2$-bisimilar models distinguishable by $\mathcal{HL}(\mathsf{E}, \downarrow)$

Comparing Expressive Power

$\mathcal{M}, w$	$\mathcal{M}', w'$	Bisimilar for
		$\mathcal{ML}(\langle \mathrm{sw} \rangle)$ $\mathcal{ML}(\langle \mathrm{br} \rangle)$ $\mathcal{ML}(\langle \mathrm{gsw} \rangle)$ $\mathcal{ML}(\langle \mathrm{gbr} \rangle)$
		$\mathcal{ML}(\langle \mathrm{sb} \rangle)$ $\mathcal{ML}(\langle \mathrm{gsb} \rangle)$

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:

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.

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.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle \mathrm{sb} \rangle)$ and $\mathcal{ML}(\langle \mathrm{sw} \rangle)$ 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.
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle \mathsf{sb} \rangle)$ and $\mathcal{ML}(\langle \mathsf{sw} \rangle)$ 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.
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over models with a single, transitive tree relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle \mathsf{sb} \rangle)$ and $\mathcal{ML}(\langle \mathsf{sw} \rangle)$ 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.
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over models with a single, transitive tree relation.
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over models with a single, $S5$ relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle \mathsf{sb} \rangle)$ and $\mathcal{ML}(\langle \mathsf{sw} \rangle)$ 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.
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over linear frames (i.e., irreflexive, transitive, and trichotomous frames).
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over models with a single, transitive tree relation.
- $\mathcal{HL}(\mathsf{E},\downarrow)$ over models with a single, $S5$ relation.

Since the translations preserve equivalence, we get:

1. The satisfiability problem for $\mathcal{ML}(\langle \mathsf{sb} \rangle)$ and $\mathcal{ML}(\langle \mathsf{sw} \rangle)$ 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)

$\mathcal{HL}(:,\downarrow) \setminus \Box\downarrow\Box$ 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)

$\mathcal{HL}(:, \downarrow) \setminus \Box\downarrow\Box$ 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 $\blacklozenge \in \{\langle \mathsf{sb} \rangle, \langle \mathsf{sw} \rangle\}$ and $\blacksquare \in \{[\mathsf{sb}], [\mathsf{sw}]\}$. The following patterns are produced by the translations:

RC Pattern	Hybrid Pattern
$\Box$	$\Box$
$\blacksquare$	$\downarrow\Box\downarrow$
$\blacklozenge$	$\downarrow$

Decidable Fragments (Syntactic Restrictions)

$\mathcal{HL}(:,\downarrow) \setminus \Box\downarrow\Box$ 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 $\blacklozenge \in \{\langle \mathsf{sb} \rangle, \langle \mathsf{sw} \rangle\}$ and $\blacksquare \in \{[\mathsf{sb}], [\mathsf{sw}]\}$. The following patterns are produced by the translations:

RC Pattern	Hybrid Pattern
$\Box$	$\Box$
$\blacksquare$	$\downarrow\Box\downarrow$
$\blacklozenge$	$\downarrow$

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

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

where $\blacksquare$ is either $\Box$ or $\blacksquare$.

Implementation in HTab

- 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 $\mathcal{HL}(\mathsf{E}, \downarrow)$.

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 $\mathcal{HL}(\mathsf{E}, \downarrow)$.
- We added a flag `--translate` that interprets the input formula as a relation-changing one. It first translates it to an $\mathcal{HL}(\mathsf{E}, \downarrow)$-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 $\mathcal{HL}(\mathsf{E}, \downarrow)$.
- We added a flag `--translate` that interprets the input formula as a relation-changing one. It first translates it to an $\mathcal{HL}(\mathsf{E}, \downarrow)$-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 PSPACE-complete.
  - Satisfiability is undecidable.

Conclusions

- Relation-changing logics are very expressive:
  - Model checking is PSPACE-complete.
  - Satisfiability is undecidable.
- We defined translations into hybrid logics:

Conclusions

- Relation-changing logics are very expressive:
  - Model checking is PSPACE-complete.
  - Satisfiability is undecidable.
- We defined translations into hybrid logics:
  - They are useful to analyze expressive power.

Conclusions

- Relation-changing logics are very expressive:
  - Model checking is 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.

Conclusions

- Relation-changing logics are very expressive:
  - Model checking is 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.

Conclusions

- Relation-changing logics are very expressive:
  - Model checking is 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 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 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 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!