Towards a Model Theory of Ordered Logics
Expressivity and Interpolation

August 23rd 2022, Vienna, MFCS 2022

Bartosz "Bart" Bednarczyk

TU Dresden & University of Wrocław

Reijo Jaakkola

Tampere University

Powered by BeamerikZ

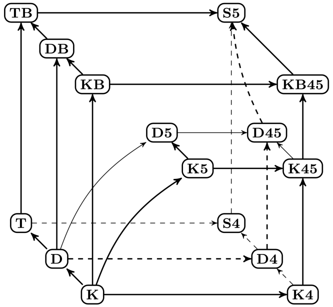
Our motivations

Our motivations

- Modal logic is everywhere:

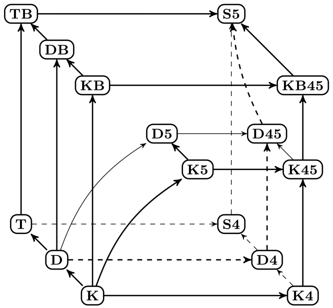
Our motivations

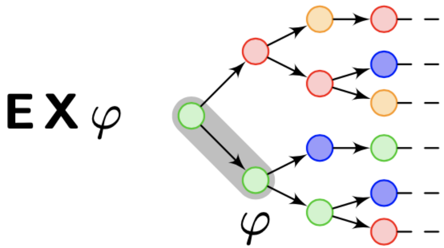
- Modal logic is everywhere: philosophy,



Our motivations

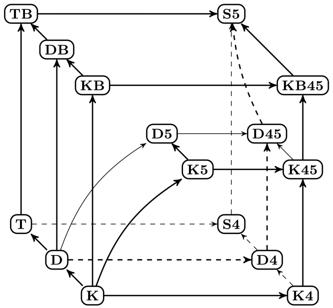
- Modal logic is everywhere: philosophy, temporal logic (CTL),



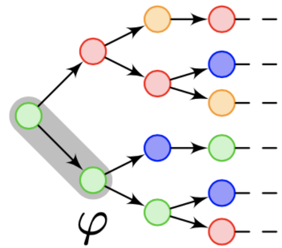


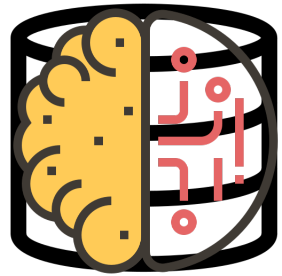
Our motivations

- Modal logic is everywhere: philosophy, temporal logic (CTL), description logic ($\mathcal{ALC}$), and more.



$\mathbf{EX}\,\varphi$



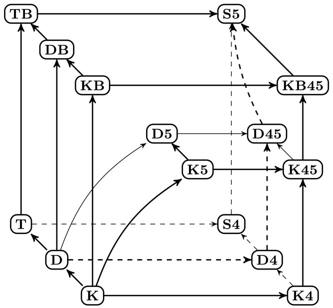


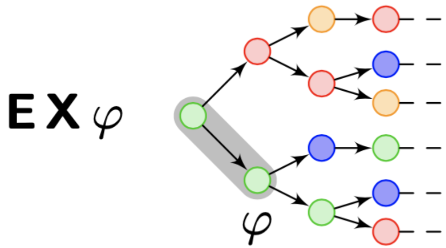
$\models$

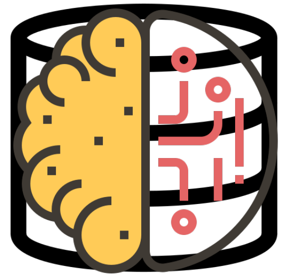
```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

Our motivations

- Modal logic is everywhere: philosophy, temporal logic (CTL), description logic ($\mathcal{ALC}$), and more.







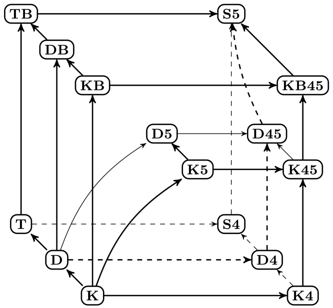
$\models$

```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

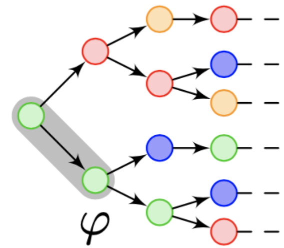
- Modal logic has desirable algorithmic and model theoretic properties.

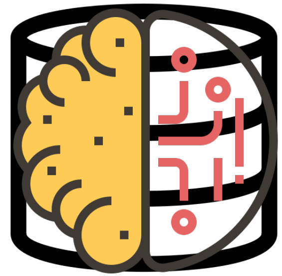
Our motivations

- Modal logic is everywhere: philosophy, temporal logic (CTL), description logic ($\mathcal{ALC}$), and more.



EX φ



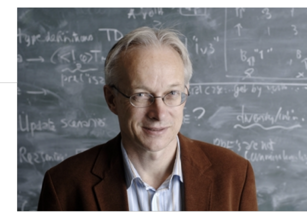


$\models$

```sql
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

- Modal logic has desirable algorithmic and model theoretic properties. More precisely:

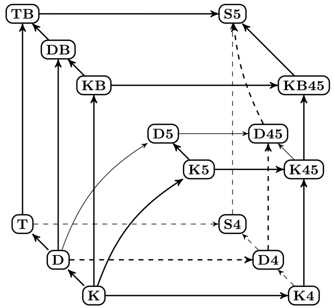
HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM

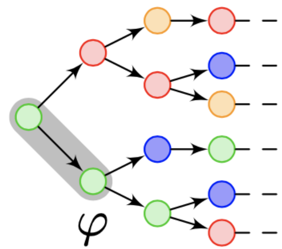
MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

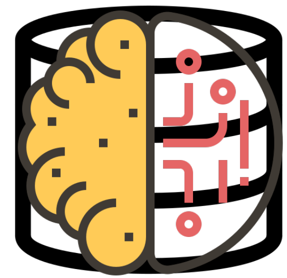
Our motivations

- Modal logic is everywhere: philosophy, temporal logic (CTL), description logic ($\mathcal{ALC}$), and more.



$\mathbf{EX}\,\varphi$





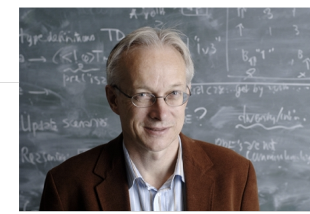
$\models$

```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

- Modal logic has desirable algorithmic and model theoretic properties. More precisely:

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM

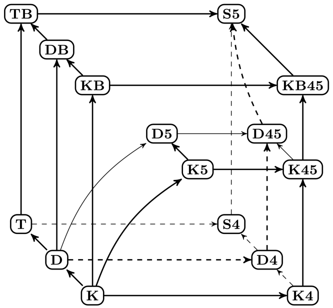
Decidability of SAT



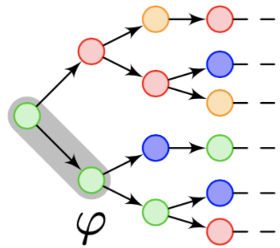
MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

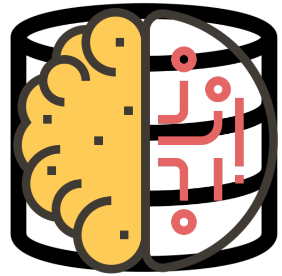
Our motivations

- Modal logic is everywhere: philosophy, temporal logic (CTL), description logic ($\mathcal{ALC}$), and more.



EX φ





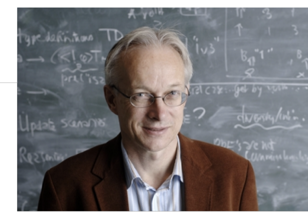
$\models$

```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

- Modal logic has desirable algorithmic and model theoretic properties. More precisely:

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM

Decidability of SAT

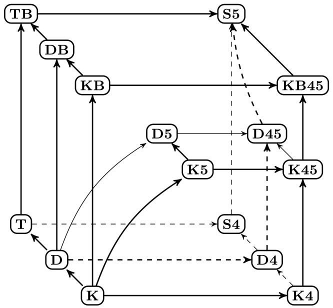


Finite Model Property

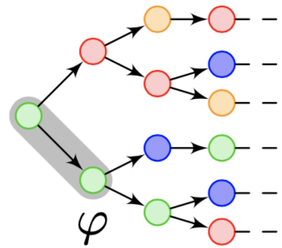
MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

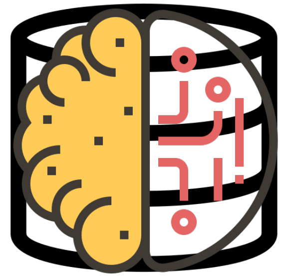
Our motivations

- Modal logic is everywhere: philosophy, temporal logic (CTL), description logic ($\mathcal{ALC}$), and more.



EX φ





$\models$

```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

- Modal logic has desirable algorithmic and model theoretic properties. More precisely:

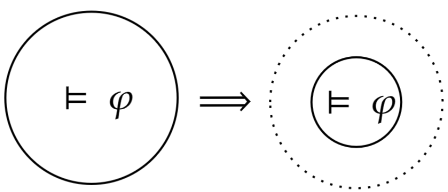
HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM

Decidability of SAT

Finite Model Property

MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

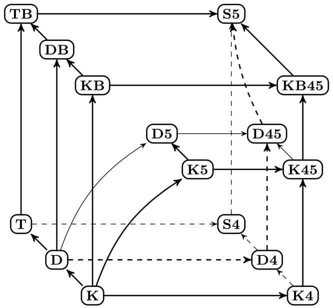
Łoś-Tarski Preservation Thm. (ŁTPT)

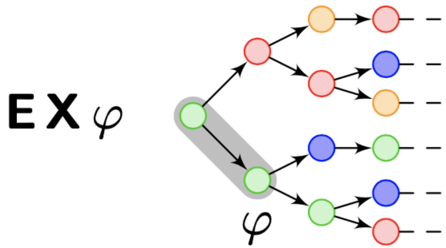


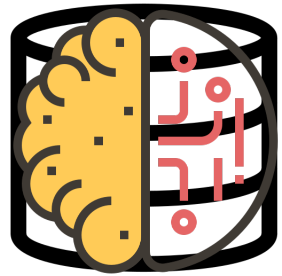
$\Longleftrightarrow \quad \varphi \equiv \forall^* \varphi'$

Our motivations

- Modal logic is everywhere: philosophy, temporal logic (CTL), description logic ($\mathcal{ALC}$), and more.







$\models$

```sql
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

- Modal logic has desirable algorithmic and model theoretic properties. More precisely:

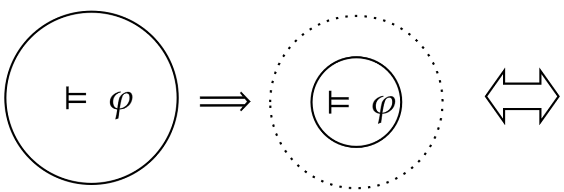
HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM

Decidability of SAT

Finite Model Property

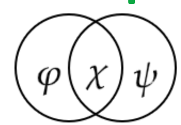
MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

Łoś-Tarski Preservation Thm. (ŁTPT)



$\models \varphi \Longrightarrow \models \varphi \iff \varphi \equiv \forall^* \varphi'$

Craig Interpolation Property (CIP)



$sig(\chi) \subseteq sig(\varphi) \cap sig(\psi)$

$\varphi \models \psi \Longrightarrow \exists \chi \;\; \varphi \models \chi \models \psi$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists \vec{y}\ \alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}), \forall \vec{y}\ \alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})$ – guard must cover free variables of φ.

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists \vec{y}\ \alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}), \forall \vec{y}\ \alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})$ – guard must cover free variables of φ.

Example 1. Some artist admires only beekeepers

$$\exists x\ artst(x) \wedge \forall y\ \ (adm(x, y) \rightarrow bkpr(y))$$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists \vec{y}\ \alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}), \forall \vec{y}\ \alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})$ – guard must cover free variables of φ.

Example 1. Some artist admires only beekeepers

$$\exists x\ artst(x) \wedge \forall y\ \ (adm(x, y) \rightarrow bkpr(y))$$

Example 2. Every artist envies every beekeeper he admires

$$\forall x\ artst(x) \rightarrow \forall y\ [adm(x, y) \rightarrow (bkpr(y) \rightarrow env(x, y))]$$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists\vec{y}\ \alpha(\vec{x},\vec{y})\wedge\varphi(\vec{x},\vec{y}), \forall\vec{y}\ \alpha(\vec{x},\vec{y})\rightarrow\varphi(\vec{x},\vec{y})$ – guard must cover free variables of φ.

Example 1. Some artist admires only beekeepers

$$\exists x\ artst(x) \wedge \forall y\ \ (adm(x,y) \rightarrow bkpr(y))$$

Example 2. Every artist envies every beekeeper he admires

$$\forall x\ artst(x) \rightarrow \forall y\ [adm(x,y) \rightarrow (bkpr(y) \rightarrow env(x,y))]$$

Coexample 3. Every artist admires every beekeeper

$$\forall x\ (artst(x) \rightarrow \forall y\ (bkpr(y) \rightarrow adm(x,y)))$$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists \vec{y}\ \alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}), \forall \vec{y}\ \alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})$ – guard must cover free variables of φ.

Example 1. Some artist admires only beekeepers

$$\exists x\ artst(x) \wedge \forall y\ \ (adm(x, y) \rightarrow bkpr(y))$$

Example 2. Every artist envies every beekeeper he admires

$$\forall x\ artst(x) \rightarrow \forall y\ [adm(x, y) \rightarrow (bkpr(y) \rightarrow env(x, y))]$$

Coexample 3. Every artist admires every beekeeper

$$\forall x\ (artst(x) \rightarrow \forall y\ (bkpr(y) \rightarrow adm(x, y)))$$

SAT

2EXPTIME-complete

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists \vec{y}\ \alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}), \forall \vec{y}\ \alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})$ – guard must cover free variables of φ.

Example 1. Some artist admires only beekeepers

$$\exists x\ artst(x) \wedge \forall y\ \ (adm(x, y) \rightarrow bkpr(y))$$

Example 2. Every artist envies every beekeeper he admires

$$\forall x\ artst(x) \rightarrow \forall y\ [adm(x, y) \rightarrow (bkpr(y) \rightarrow env(x, y))]$$

Coexample 3. Every artist admires every beekeeper

$$\forall x\ (artst(x) \rightarrow \forall y\ (bkpr(y) \rightarrow adm(x, y)))$$

SAT	**FMP**
2ExpTime-complete	

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists\vec{y}\ \alpha(\vec{x},\vec{y})\wedge\varphi(\vec{x},\vec{y}), \forall\vec{y}\ \alpha(\vec{x},\vec{y})\rightarrow\varphi(\vec{x},\vec{y})$ – guard must cover free variables of φ.

Example 1. Some artist admires only beekeepers

$$\exists x\ artst(x) \wedge \forall y\ \ (adm(x,y) \rightarrow bkpr(y))$$

Example 2. Every artist envies every beekeeper he admires

$$\forall x\ artst(x) \rightarrow \forall y\ [adm(x,y) \rightarrow (bkpr(y) \rightarrow env(x,y))]$$

Coexample 3. Every artist admires every beekeeper

$$\forall x\ (artst(x) \rightarrow \forall y\ (bkpr(y) \rightarrow adm(x,y)))$$

SAT	**FMP**	**CIP**
2ExpTime-complete		
		

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The guarded fragment of $\mathcal{FO}$ is obtained by relativising quantifiers by atoms.
- $\exists \vec{y}\ \alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}), \forall \vec{y}\ \alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})$ – guard must cover free variables of φ.

Example 1. Some artist admires only beekeepers

$$\exists x\ artst(x) \wedge \forall y\ \ (adm(x, y) \rightarrow bkpr(y))$$

Example 2. Every artist envies every beekeeper he admires

$$\forall x\ artst(x) \rightarrow \forall y\ [adm(x, y) \rightarrow (bkpr(y) \rightarrow env(x, y))]$$

Coexample 3. Every artist admires every beekeeper

$$\forall x\ (artst(x) \rightarrow \forall y\ (bkpr(y) \rightarrow adm(x, y)))$$

SAT	FMP	CIP	ŁTPT
2EXPTIME-complete			
			

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1(stud(x_1) \rightarrow \neg\forall x_2(prof(x_2) \rightarrow admires(x_1, x_2)))$$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1(stud(x_1) \rightarrow \neg\forall x_2(prof(x_2) \rightarrow admires(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1(lect(x_1) \rightarrow \neg\exists x_2(prof(x_2) \land \forall x_3(stud(x_3) \rightarrow intro(x_1, x_2, x_3))))$$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1 (stud(x_1) \rightarrow \neg \forall x_2 (prof(x_2) \rightarrow admires(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1 (lect(x_1) \rightarrow \neg \exists x_2 (prof(x_2) \wedge \forall x_3 (stud(x_3) \rightarrow intro(x_1, x_2, x_3))))$$

Coexample 1. $\forall x_1 r(x_1, x_1)$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1(\mathit{stud}(x_1) \to \neg\forall x_2(\mathit{prof}(x_2) \to \mathit{admires}(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1(\mathit{lect}(x_1) \to \neg\exists x_2(\mathit{prof}(x_2) \wedge \forall x_3(\mathit{stud}(x_3) \to \mathit{intro}(x_1, x_2, x_3))))$$

Coexample 1. $\forall x_1 r(x_1, x_1)$

Coexample 2. $\forall x_1 \forall x_2 r(x_1, x_2) \to s(x_2, x_1)$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathrm{L}_{\mathrm{pre}}, \mathrm{L}_{\mathrm{suf}}, \mathrm{L}_{\mathrm{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1(\mathit{stud}(x_1) \to \neg\forall x_2(\mathit{prof}(x_2) \to \mathit{admires}(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1(\mathit{lect}(x_1) \to \neg\exists x_2(\mathit{prof}(x_2) \land \forall x_3(\mathit{stud}(x_3) \to \mathit{intro}(x_1, x_2, x_3))))$$

Coexample 1. $\forall x_1 r(x_1, x_1)$

Coexample 2. $\forall x_1 \forall x_2 r(x_1, x_2) \to s(x_2, x_1)$

Coexample 3. $\forall x_1 \forall x_2 \forall x_3 r(x_1, x_2) \land r(x_2, x_3) \to r(x_1, x_3)$

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\text{pre}}, \mathsf{L}_{\text{suf}}, \mathsf{L}_{\text{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1(stud(x_1) \rightarrow \neg\forall x_2(prof(x_2) \rightarrow admires(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1(lect(x_1) \rightarrow \neg\exists x_2(prof(x_2) \wedge \forall x_3(stud(x_3) \rightarrow intro(x_1, x_2, x_3))))$$

Coexample 1. $\forall x_1 r(x_1, x_1)$

Coexample 2. $\forall x_1 \forall x_2 r(x_1, x_2) \rightarrow s(x_2, x_1)$

Coexample 3. $\forall x_1 \forall x_2 \forall x_3 r(x_1, x_2) \wedge r(x_2, x_3) \rightarrow r(x_1, x_3)$

SAT

PSpace/Tower-complete

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathrm{pre}}, \mathsf{L}_{\mathrm{suf}}, \mathsf{L}_{\mathrm{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1 (stud(x_1) \rightarrow \neg \forall x_2 (prof(x_2) \rightarrow admires(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1 (lect(x_1) \rightarrow \neg \exists x_2 (prof(x_2) \wedge \forall x_3 (stud(x_3) \rightarrow intro(x_1, x_2, x_3))))$$

Coexample 1. $\forall x_1 r(x_1, x_1)$

Coexample 2. $\forall x_1 \forall x_2 r(x_1, x_2) \rightarrow s(x_2, x_1)$

Coexample 3. $\forall x_1 \forall x_2 \forall x_3 r(x_1, x_2) \wedge r(x_2, x_3) \rightarrow r(x_1, x_3)$

SAT **FMP**

PSPACE/TOWER-complete

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathrm{L}_{\mathrm{pre}}, \mathrm{L}_{\mathrm{suf}}, \mathrm{L}_{\mathrm{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1 (stud(x_1) \rightarrow \neg \forall x_2 (prof(x_2) \rightarrow admires(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1 (lect(x_1) \rightarrow \neg \exists x_2 (prof(x_2) \wedge \forall x_3 (stud(x_3) \rightarrow intro(x_1, x_2, x_3))))$$

Coexample 1. $\forall x_1 r(x_1, x_1)$

Coexample 2. $\forall x_1 \forall x_2 r(x_1, x_2) \rightarrow s(x_2, x_1)$

Coexample 3. $\forall x_1 \forall x_2 \forall x_3 r(x_1, x_2) \wedge r(x_2, x_3) \rightarrow r(x_1, x_3)$

SAT	FMP	CIP
PSPACE/TOWER-complete		
		

Two nice logics: $\mathcal{GF}$ [Andreka et al. 1998] and ordered logics [Herzig, Quine, B.]

- The ordered fragments $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ of $\mathcal{FO}$ are obtained by keeping the variables ordered.
- In atoms we can use only pref/suf/inf ixes of the sequences of already quantified variables.

Example 1. No student admires every professor

$$\forall x_1 (stud(x_1) \rightarrow \neg \forall x_2 (prof(x_2) \rightarrow admires(x_1, x_2)))$$

Example 2. No lecturer introduces any professor to every student

$$\forall x_1 (lect(x_1) \rightarrow \neg \exists x_2 (prof(x_2) \wedge \forall x_3 (stud(x_3) \rightarrow intro(x_1, x_2, x_3))))$$

Coexample 1. $\forall x_1 r(x_1, x_1)$

Coexample 2. $\forall x_1 \forall x_2 r(x_1, x_2) \rightarrow s(x_2, x_1)$

Coexample 3. $\forall x_1 \forall x_2 \forall x_3 r(x_1, x_2) \wedge r(x_2, x_3) \rightarrow r(x_1, x_3)$

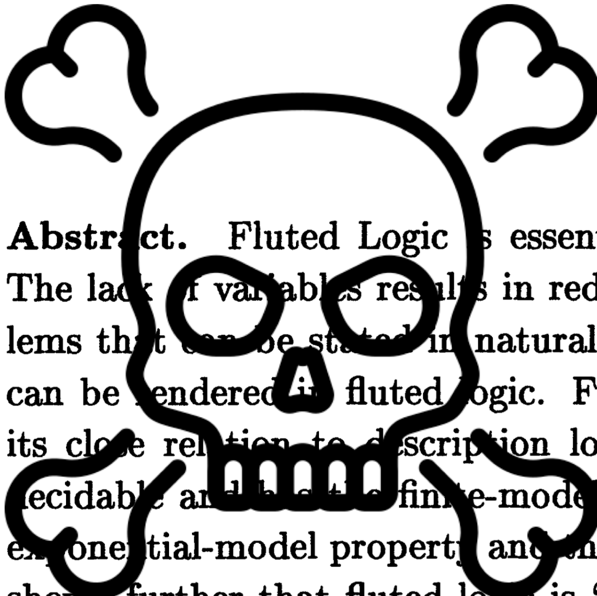
SAT	FMP	CIP	ŁTPT
PSPACE/TOWER-complete			

On the infamous work of Purdy

On the infamous work of Purdy

William C. Purdy

Complexity and Nicety of Fluted Logic



Abstract. Fluted Logic is essentially first-order predicate logic deprived of variables. The lack of variables results in reduced expressiveness. Nevertheless, many logical problems that can be stated in natural language, such as the famous Schubert's Steamroller, can be rendered in fluted logic. Further evidence of the expressiveness of fluted logic is its close relation to description logics. Already it has been shown that fluted logic is decidable and has the finite-model property. This paper shows that fluted logic has the exponential-model property and that deciding satisfiability is NEXPTIME-complete. It is shown further that fluted logic is 'nice', that is, it shares with first-order predicate logic the interpolation property and model preservation properties.

On the infamous work of Purdy

William C. Purdy

Complexity and Nicety of Fluted Logic

Abstract. Fluted Logic is essentially first-order predicate logic deprived of variables. The lack of variables results in reduced expressiveness. Nevertheless, many logical problems that can be stated in natural language, such as the famous Schubert's Steamroller, can be rendered in fluted logic. Further evidence of the expressiveness of fluted logic is its close relation to description logics. Already it has been shown that fluted logic is decidable and has the finite-model property. This paper shows that fluted logic has the exponential-model property and that deciding satisfiability is NEXPTIME-complete. It is shown further that fluted logic is 'nice', that is, it shares with first-order predicate logic the interpolation property and model preservation properties.

Error 1. SAT of $\mathsf{L}_{\mathsf{suf}}$ is Tower-hard (discovered by Ian Pratt-Hartmann)

On the infamous work of Purdy

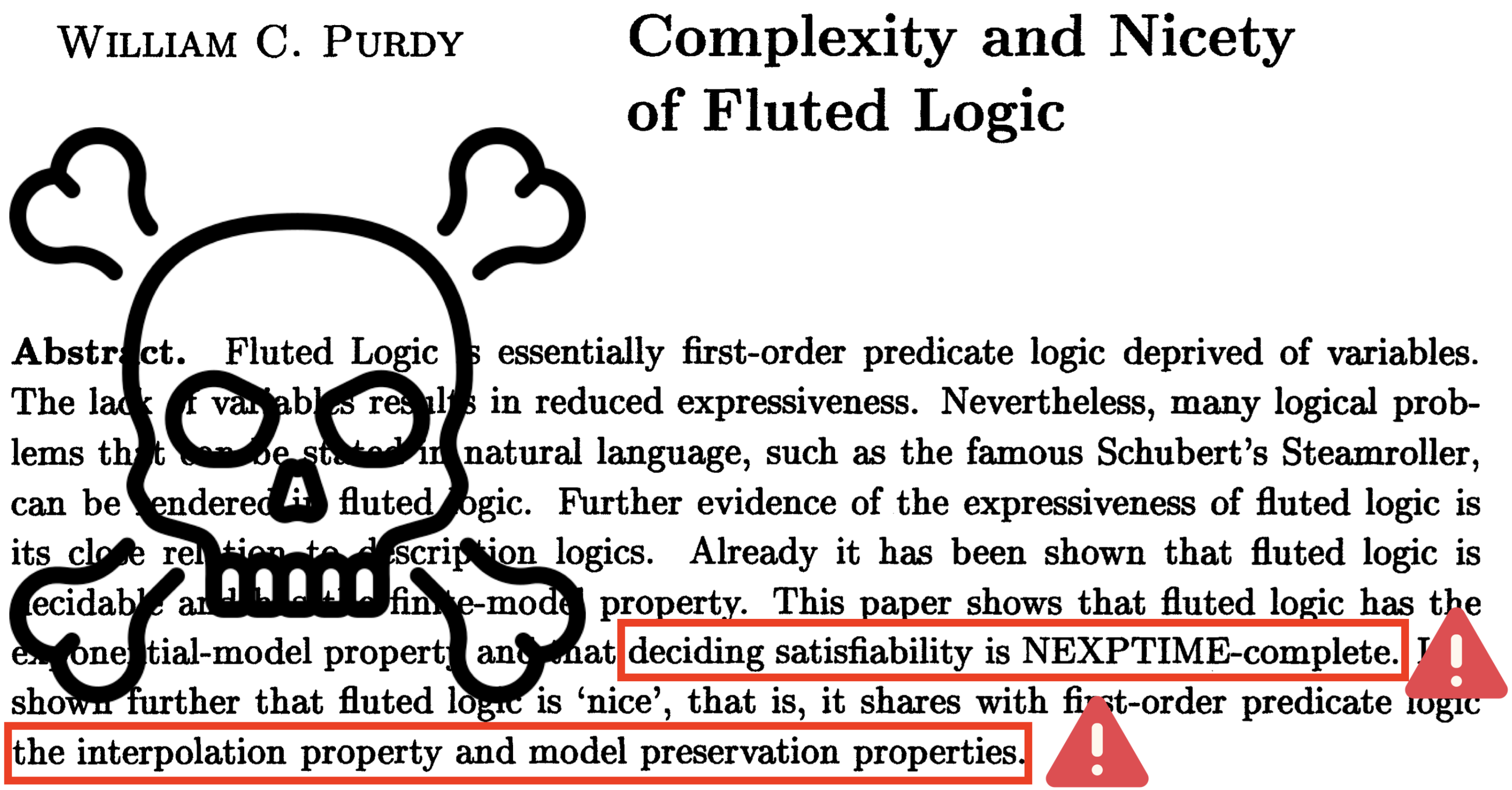
WILLIAM C. PURDY

Complexity and Nicety of Fluted Logic

Abstract. Fluted Logic is essentially first-order predicate logic deprived of variables. The lack of variables results in reduced expressiveness. Nevertheless, many logical problems that can be stated in natural language, such as the famous Schubert's Steamroller, can be rendered in fluted logic. Further evidence of the expressiveness of fluted logic is its close relation to description logics. Already it has been shown that fluted logic is decidable and has the finite-model property. This paper shows that fluted logic has the exponential-model property and that deciding satisfiability is NEXPTIME-complete. It is shown further that fluted logic is 'nice', that is, it shares with first-order predicate logic the interpolation property and model preservation properties.

Error 1. SAT of L_{suf} is TOWER-hard (discovered by Ian Pratt-Hartmann)

Error 2. L_{suf} does not enjoy CIP (this work!)


William C. Purdy

Complexity and Nicety of Fluted Logic

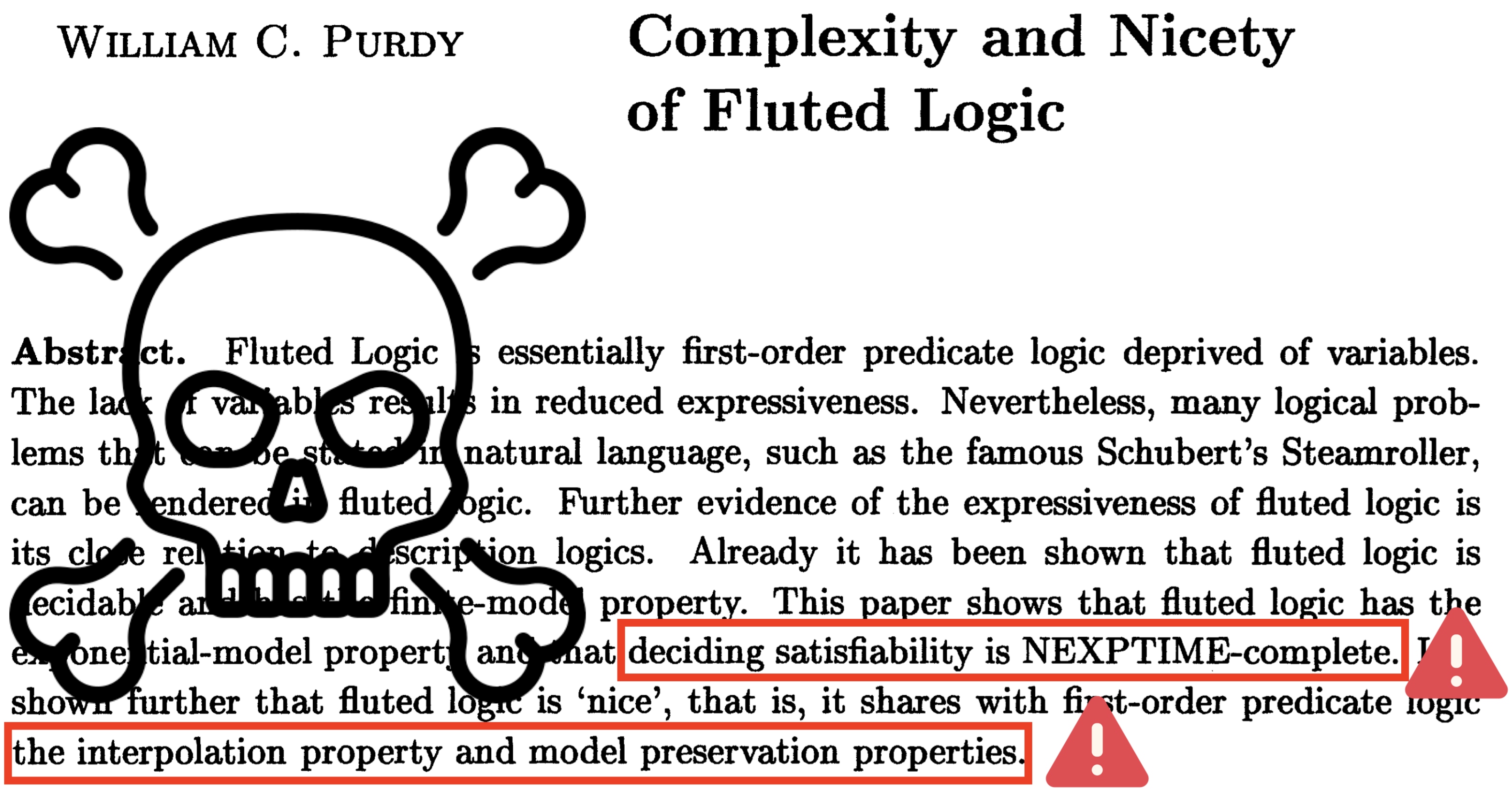
Abstract. Fluted Logic is essentially first-order predicate logic deprived of variables. The lack of variables results in reduced expressiveness. Nevertheless, many logical problems that can be stated in natural language, such as the famous Schubert's Steamroller, can be rendered in fluted logic. Further evidence of the expressiveness of fluted logic is its close relation to description logics. Already it has been shown that fluted logic is decidable and has the finite-model property. This paper shows that fluted logic has the exponential-model property and that deciding satisfiability is NEXPTIME-complete. It is shown further that fluted logic is 'nice', that is, it shares with first-order predicate logic the interpolation property and model preservation properties.

Error 1. SAT of $\mathrm{L}_{\mathrm{suf}}$ is $\textsc{Tower}$-hard (discovered by Ian Pratt-Hartmann)

Error 2. $\mathrm{L}_{\mathrm{suf}}$ does not enjoy CIP (this work!)

Fact: ŁPTP proof of Purdy lack of mathematical arguments.

On the infamous work of Purdy



WILLIAM C. PURDY

Complexity and Nicety of Fluted Logic

Abstract. Fluted Logic is essentially first-order predicate logic deprived of variables. The lack of variables results in reduced expressiveness. Nevertheless, many logical problems that can be stated in natural language, such as the famous Schubert's Steamroller, can be rendered in fluted logic. Further evidence of the expressiveness of fluted logic is its close relation to description logics. Already it has been shown that fluted logic is decidable and has the finite-model property. This paper shows that fluted logic has the exponential-model property and that deciding satisfiability is NEXPTIME-complete. It is shown further that fluted logic is 'nice', that is, it shares with first-order predicate logic the interpolation property and model preservation properties.

Error 1. SAT of L_{suf} is TOWER-hard (discovered by Ian Pratt-Hartmann)

Error 2. L_{suf} does not enjoy CIP (this work!)

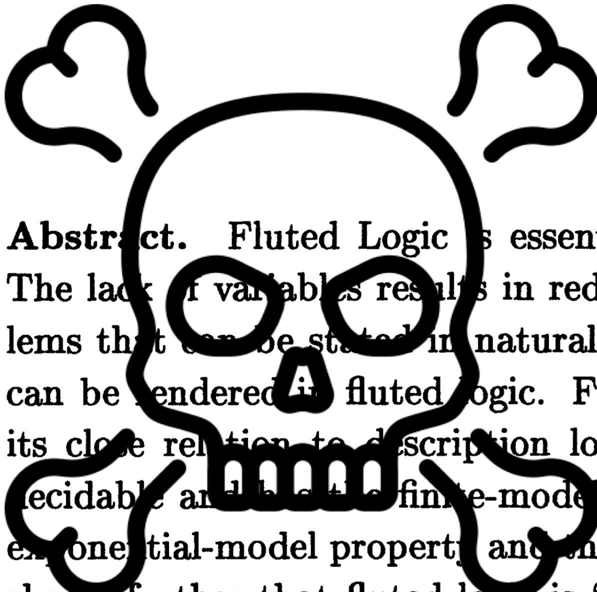
Fact: ŁPTP proof of Purdy lack of mathematical arguments.

Conclusion:

On the infamous work of Purdy

WILLIAM C. PURDY

Complexity and Nicety of Fluted Logic



Abstract. Fluted Logic is essentially first-order predicate logic deprived of variables. The lack of variables results in reduced expressiveness. Nevertheless, many logical problems that can be stated in natural language, such as the famous Schubert's Steamroller, can be rendered in fluted logic. Further evidence of the expressiveness of fluted logic is its close relation to description logics. Already it has been shown that fluted logic is decidable and has the finite-model property. This paper shows that fluted logic has the exponential-model property and that deciding satisfiability is NEXPTIME-complete. It is shown further that fluted logic is 'nice', that is, it shares with first-order predicate logic the interpolation property and model preservation properties.

Error 1. SAT of $\mathrm{L}_{\mathrm{suf}}$ is TOWER-hard (discovered by Ian Pratt-Hartmann)

Error 2. $\mathrm{L}_{\mathrm{suf}}$ does not enjoy CIP (this work!)

Fact: ŁPTP proof of Purdy lack of mathematical arguments.

Conclusion:

We need to study ordered logics more!

Our contribution (Part I)

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.

Our contribution (Part I)

We study $L_{pre}, L_{suf}, L_{inf}$ and their guarded subfragments $G_{pre}, G_{suf}, G_{inf}$.

1. We introduced a suitable notion of bisimulation.

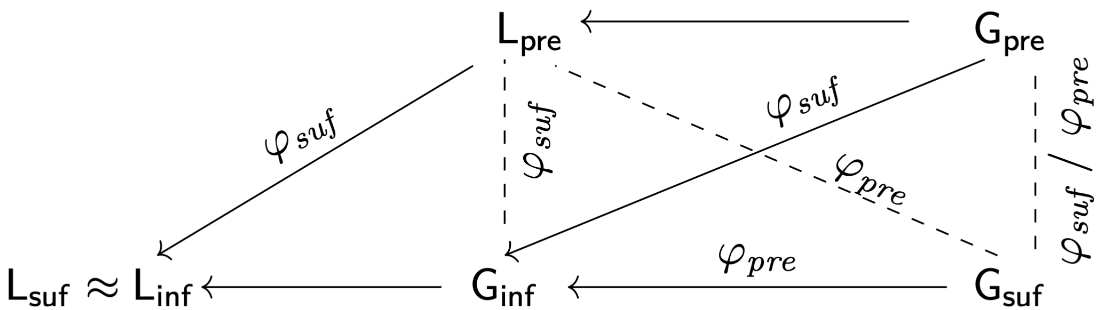
2. Comparison of relative expressive powers

Our contribution (Part I)

We study $\mathsf{L}_{pre}, \mathsf{L}_{suf}, \mathsf{L}_{inf}$ and their guarded subfragments $\mathsf{G}_{pre}, \mathsf{G}_{suf}, \mathsf{G}_{inf}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers



Solid: more expr. Dashed: incomp.

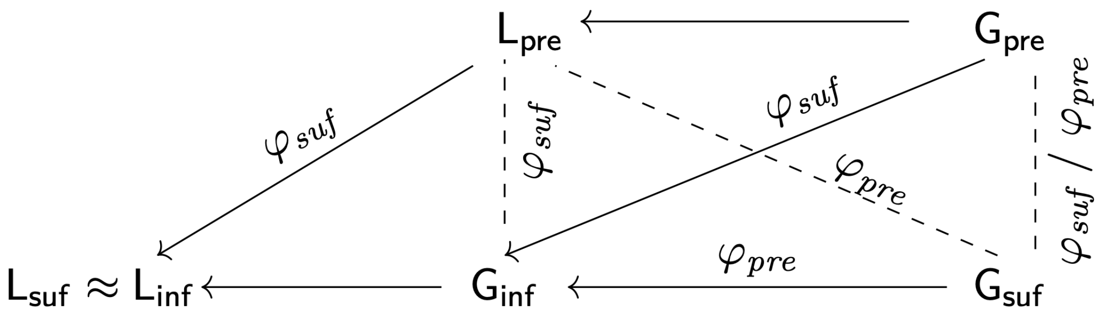
$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{T}(x_2, x_3)$$

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.
2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$$

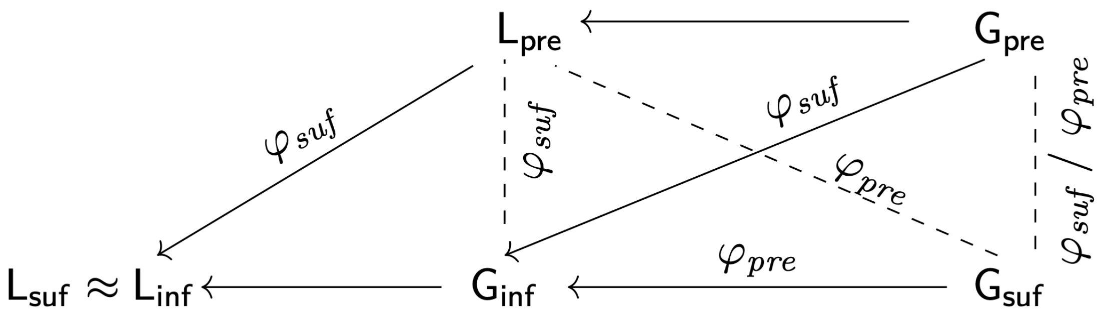
$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$$

Our contribution (Part I)

We study $\mathsf{L}_{pre}, \mathsf{L}_{suf}, \mathsf{L}_{inf}$ and their guarded subfragments $\mathsf{G}_{pre}, \mathsf{G}_{suf}, \mathsf{G}_{inf}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$$

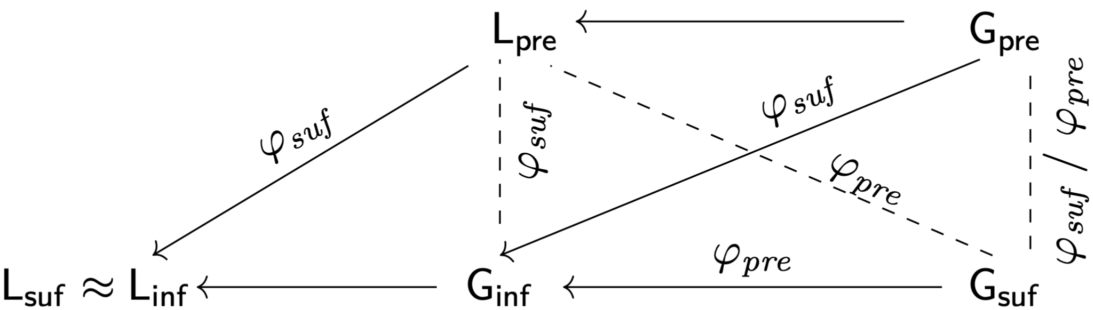
3. L_{suf} and L_{inf} do not enjoy CIP.

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3 \; \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3 \; \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$$

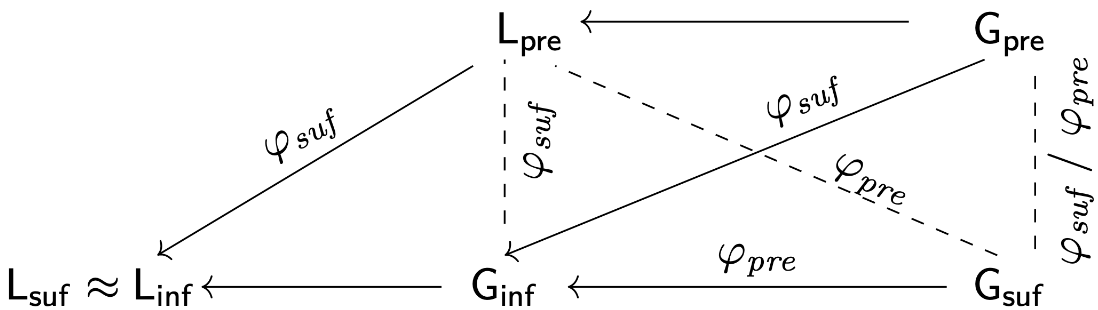
3. $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$ do not enjoy CIP. A very simple counterexample:

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3 \ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3 \ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$$

3. $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$ do not enjoy CIP. A very simple counterexample:

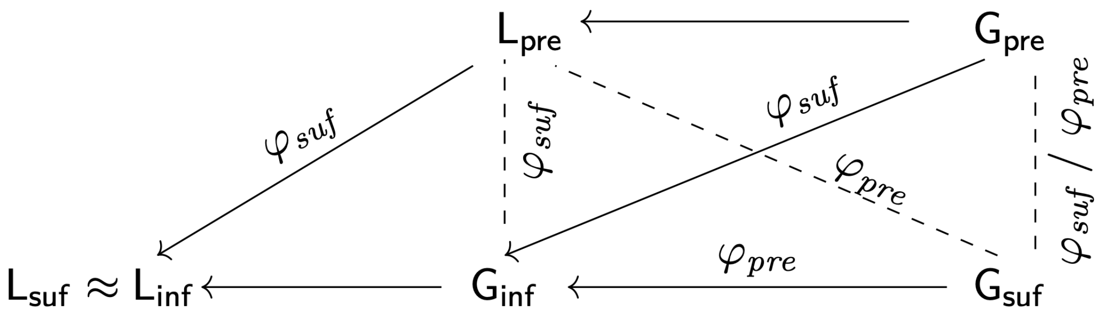
$$\varphi := \forall x_1 \forall x_2 \forall x_3 \ [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \to (\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3))]$$

Our contribution (Part I)

We study $\mathsf{L}_{pre}, \mathsf{L}_{suf}, \mathsf{L}_{inf}$ and their guarded subfragments $\mathsf{G}_{pre}, \mathsf{G}_{suf}, \mathsf{G}_{inf}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/\sim_{\mathsf{L}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\, \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\, \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$$

3. L_{suf} and L_{inf} do not enjoy CIP. A very simple counterexample:

$$\varphi := \forall x_1 \forall x_2 \forall x_3\, [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \to (\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3))] \wedge \forall x_1 \forall x_2\, [(\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3)) \to \mathrm{R}(x_1, x_2)]$$

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.

$$
\begin{tikzcd}
 & \mathsf{L}_{\mathsf{pre}} \arrow[dl, "\varphi_{suf}"'] \arrow[d, dash, dashed, "\varphi_{suf}"] \arrow[dr, dash, dashed, "\varphi_{pre}"] & \mathsf{G}_{\mathsf{pre}} \arrow[l] \arrow[dl, "\varphi_{suf}"'] \arrow[d, dash, dashed, "\varphi_{suf} / \varphi_{pre}"] \\
\mathsf{L}_{\mathsf{suf}} \approx \mathsf{L}_{\mathsf{inf}} & \mathsf{G}_{\mathsf{inf}} \arrow[l] & \mathsf{G}_{\mathsf{suf}} \arrow[l, "\varphi_{pre}"']
\end{tikzcd}
$$

Solid: more expr. Dashed: incomp.

$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$

$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$

3. $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$ do not enjoy CIP. A very simple counterexample:

$$\varphi := \forall x_1 \forall x_2 \forall x_3\ [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \to (\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3))] \wedge \forall x_1 \forall x_2\ [(\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3)) \to \mathrm{R}(x_1, x_2)]$$

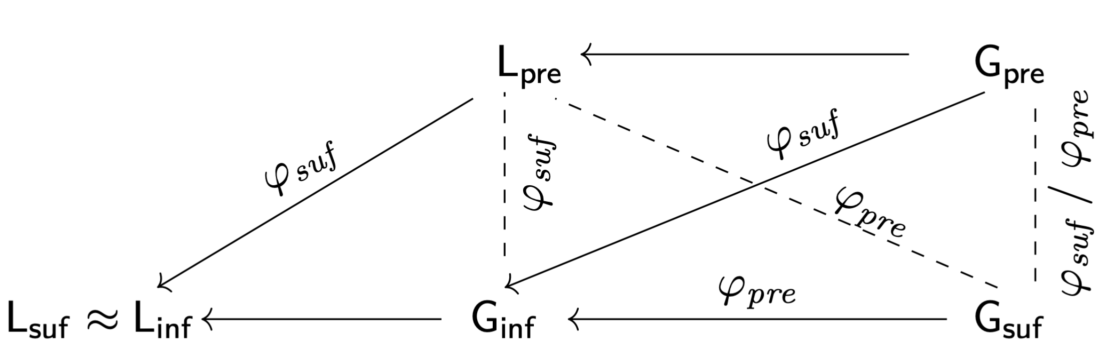
$$\psi := \exists x_1 \exists x_2 \exists x_3 [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \wedge \mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)]$$

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/\sim_{\mathsf{L}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$

$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$

3. $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$ do not enjoy CIP. A very simple counterexample:

$$\varphi := \forall x_1 \forall x_2 \forall x_3\ [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \to (\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3))]\ \wedge \forall x_1 \forall x_2\ [(\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3)) \to \mathrm{R}(x_1, x_2)]$$

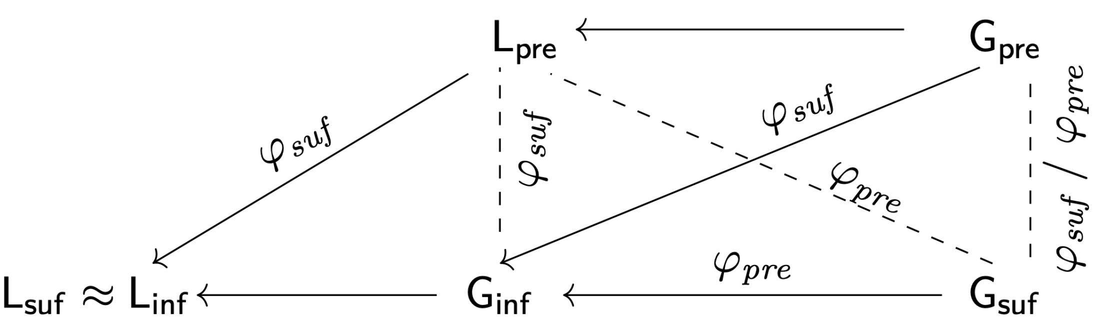
$$\psi := \exists x_1 \exists x_2 \exists x_3 [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \wedge \mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)]\ \wedge \forall x_1 \forall x_2\ [(\mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)) \to \neg \mathrm{R}(x_1, x_2)]$$

Our contribution (Part I)

We study $\mathsf{L}_{pre}, \mathsf{L}_{suf}, \mathsf{L}_{inf}$ and their guarded subfragments $\mathsf{G}_{pre}, \mathsf{G}_{suf}, \mathsf{G}_{inf}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{T}(x_2, x_3)$$

3. L_{suf} and L_{inf} do not enjoy CIP. A very simple counterexample:

$$\varphi := \forall x_1 \forall x_2 \forall x_3\ [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \rightarrow (\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3))] \wedge \forall x_1 \forall x_2\ [(\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3)) \rightarrow \mathrm{R}(x_1, x_2)]$$

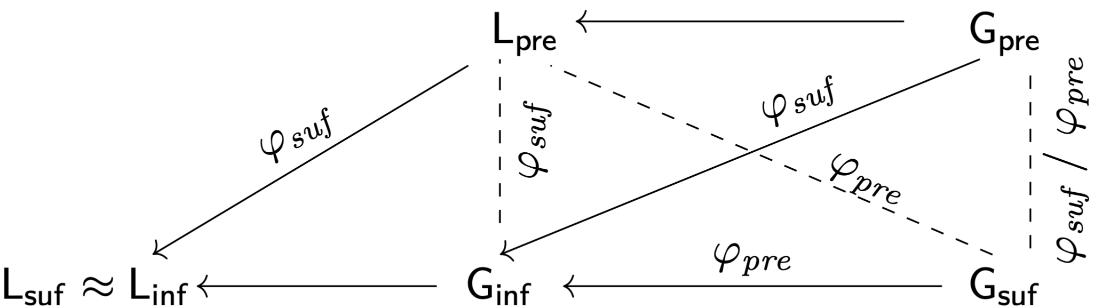
$$\psi := \exists x_1 \exists x_2 \exists x_3 [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \wedge \mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)] \wedge \forall x_1 \forall x_2\ [(\mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)) \rightarrow \neg \mathrm{R}(x_1, x_2)]$$

$\varphi \models \neg\psi$ (why?)

Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.
2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \to \mathrm{T}(x_2, x_3)$$

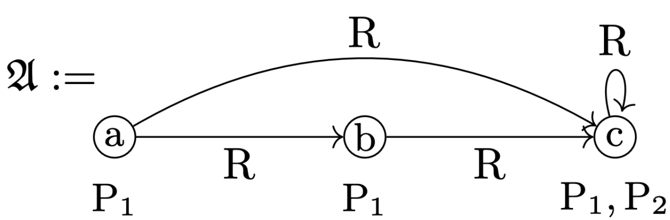
3. $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$ do not enjoy CIP. A very simple counterexample:

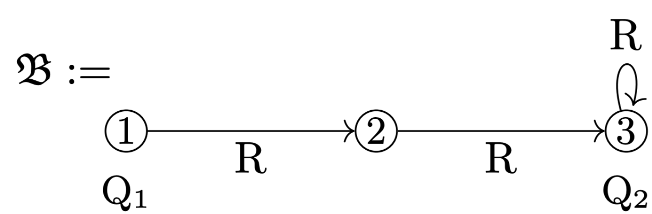
$$\varphi := \forall x_1 \forall x_2 \forall x_3\ [\mathrm{R}(x_1, x_2) \land \mathrm{R}(x_2, x_3) \to (\mathrm{P}_1(x_1) \land \mathrm{P}_2(x_3))] \land \forall x_1 \forall x_2\ [(\mathrm{P}_1(x_1) \land \mathrm{P}_2(x_3)) \to \mathrm{R}(x_1, x_2)]$$

$$\psi := \exists x_1 \exists x_2 \exists x_3 [\mathrm{R}(x_1, x_2) \land \mathrm{R}(x_2, x_3) \land \mathrm{Q}_1(x_1) \land \mathrm{Q}_2(x_2)] \land \forall x_1 \forall x_2\ [(\mathrm{Q}_1(x_1) \land \mathrm{Q}_2(x_2)) \to \neg \mathrm{R}(x_1, x_2)]$$

$\varphi \models \neg\psi$ (why?)

but $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ are



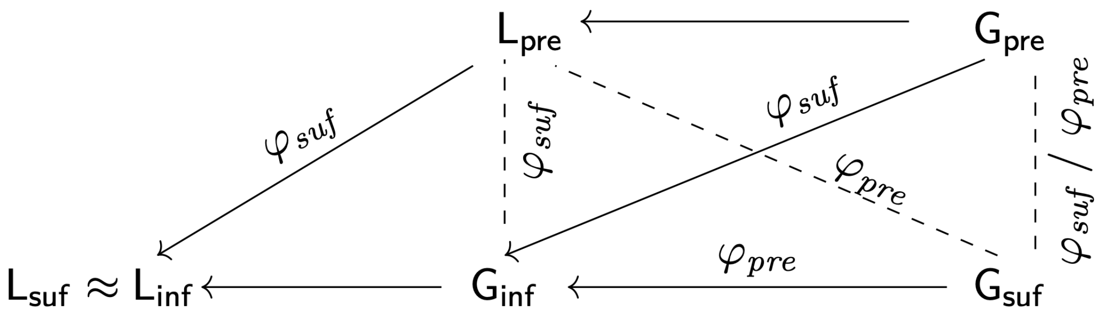


Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{S}(x_1, x_2)$

$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{T}(x_2, x_3)$

3. $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$ do not enjoy CIP. A very simple counterexample:

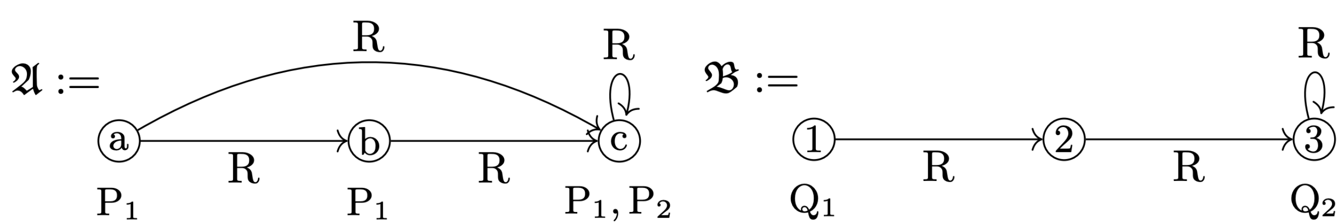
$$\varphi := \forall x_1 \forall x_2 \forall x_3\ [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \rightarrow (\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3))]\ \wedge \forall x_1 \forall x_2\ [(\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3)) \rightarrow \mathrm{R}(x_1, x_2)]$$

$$\psi := \exists x_1 \exists x_2 \exists x_3 [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \wedge \mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)]\ \wedge \forall x_1 \forall x_2\ [(\mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)) \rightarrow \neg \mathrm{R}(x_1, x_2)]$$

$\varphi \models \neg\psi$ (why?)

but $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ are

$\mathsf{L}_{\mathsf{inf}}[\{\mathrm{R}\}]$-bisimilar!

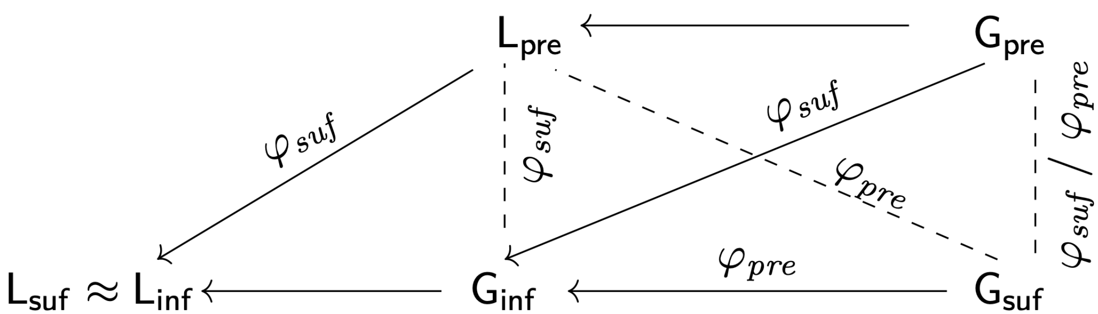


Our contribution (Part I)

We study $\mathsf{L}_{\mathsf{pre}}, \mathsf{L}_{\mathsf{suf}}, \mathsf{L}_{\mathsf{inf}}$ and their guarded subfragments $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$.

1. We introduced a suitable notion of bisimulation.

2. Comparison of relative expressive powers + Van-Benthem Style Theorems, i.e. $\mathcal{FO}/{\sim_{\mathsf{L}}} = \mathsf{L}$.



Solid: more expr. Dashed: incomp.

$$\varphi_{pre} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{S}(x_1, x_2)$$

$$\varphi_{suf} := \forall x_1 \forall x_2 \forall x_3\ \mathrm{R}(x_1, x_2, x_3) \rightarrow \mathrm{T}(x_2, x_3)$$

3. $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$ do not enjoy CIP. A very simple counterexample:

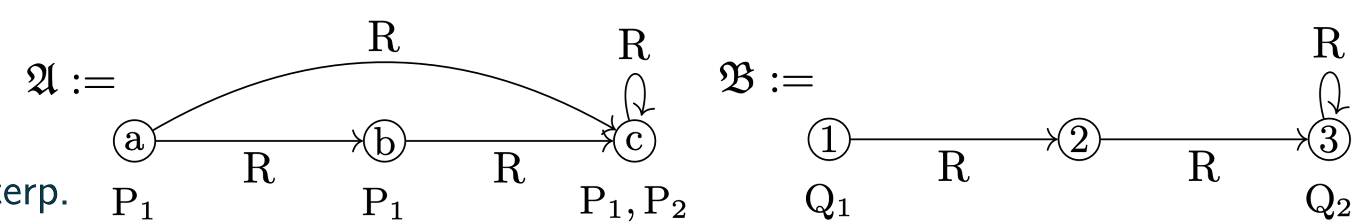
$$\varphi := \forall x_1 \forall x_2 \forall x_3\ [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \rightarrow (\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3))] \wedge \forall x_1 \forall x_2\ [(\mathrm{P}_1(x_1) \wedge \mathrm{P}_2(x_3)) \rightarrow \mathrm{R}(x_1, x_2)]$$

$$\psi := \exists x_1 \exists x_2 \exists x_3 [\mathrm{R}(x_1, x_2) \wedge \mathrm{R}(x_2, x_3) \wedge \mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)] \wedge \forall x_1 \forall x_2\ [(\mathrm{Q}_1(x_1) \wedge \mathrm{Q}_2(x_2)) \rightarrow \neg \mathrm{R}(x_1, x_2)]$$

$\varphi \models \neg\psi$ (why?)

but $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ are

$\mathsf{L}_{\mathsf{inf}}[\{\mathrm{R}\}]$-bisimilar! $\Rightarrow$ no $\mathsf{L}_{\mathsf{inf}}$-interp.



Our contribution (Part II)

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t.

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \land \psi$ then

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take φ, ψ with models $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ and

Our contribution (Part II)

4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take φ, ψ with models $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ and amalgamate $\mathfrak{A}, \mathfrak{B}$ into a single $\mathfrak{U} \models \varphi \wedge \psi$.

Our contribution (Part II)

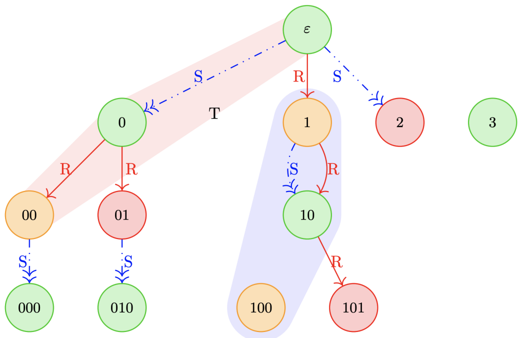
4. We prove that L_{pre} and $G_{pre}, G_{suf}, G_{inf}$ enjoy CIP.

Proof method: interpolation via amalgamation

φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in$ L there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take φ, ψ with models $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ and amalgamate $\mathfrak{A}, \mathfrak{B}$ into a single $\mathfrak{U} \models \varphi \wedge \psi$.



Tree models

Our contribution (Part II)

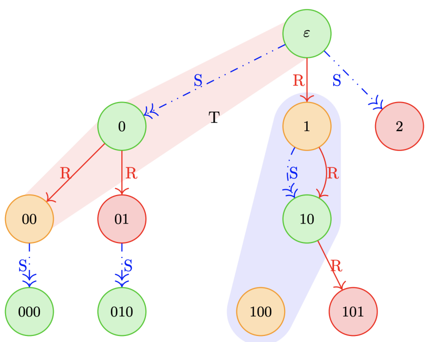
4. We prove that $\mathsf{L}_{\mathrm{pre}}$ and $\mathsf{G}_{\mathrm{pre}}, \mathsf{G}_{\mathrm{suf}}, \mathsf{G}_{\mathrm{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

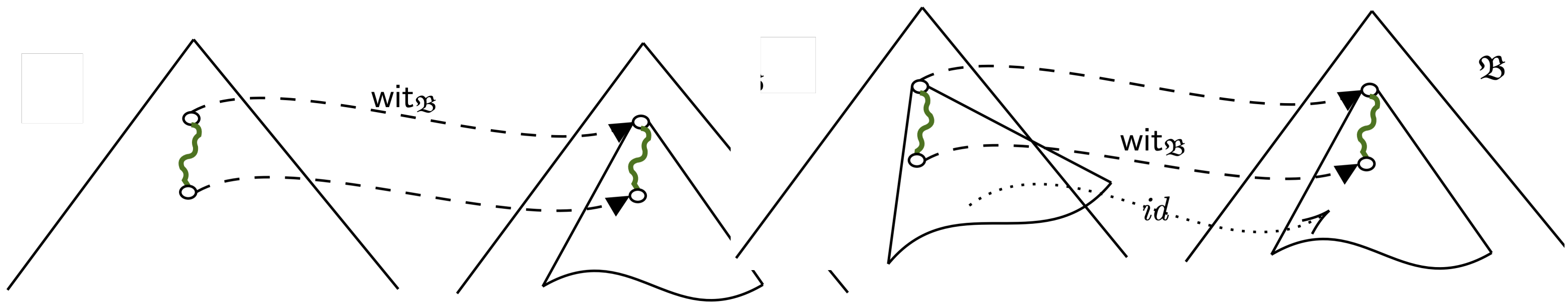
φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take φ, ψ with models $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ and amalgamate $\mathfrak{A}, \mathfrak{B}$ into a single $\mathfrak{U} \models \varphi \wedge \psi$.



Tree models



A novel "complete and repair" model construction method

Our contribution (Part II)

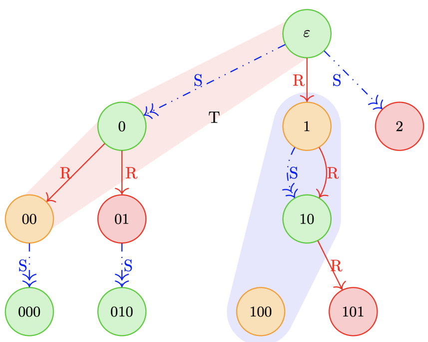
4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

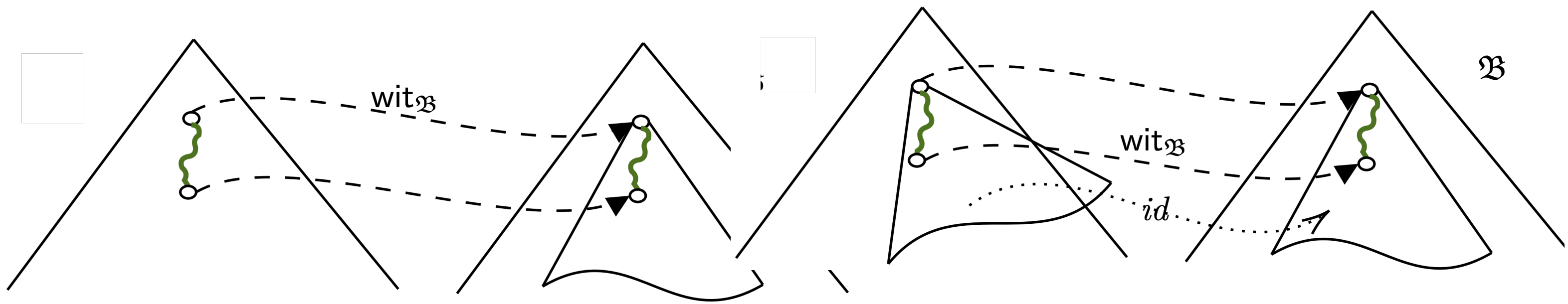
φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take φ, ψ with models $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ and amalgamate $\mathfrak{A}, \mathfrak{B}$ into a single $\mathfrak{U} \models \varphi \wedge \psi$.



Tree models



A novel "complete and repair" model construction method

Works beyond forward $\mathcal{GF}$. A good meta-heuristic.

Our contribution (Part II)

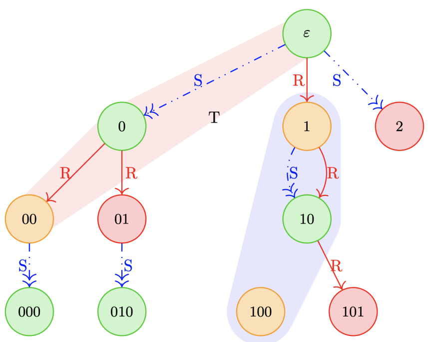
4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

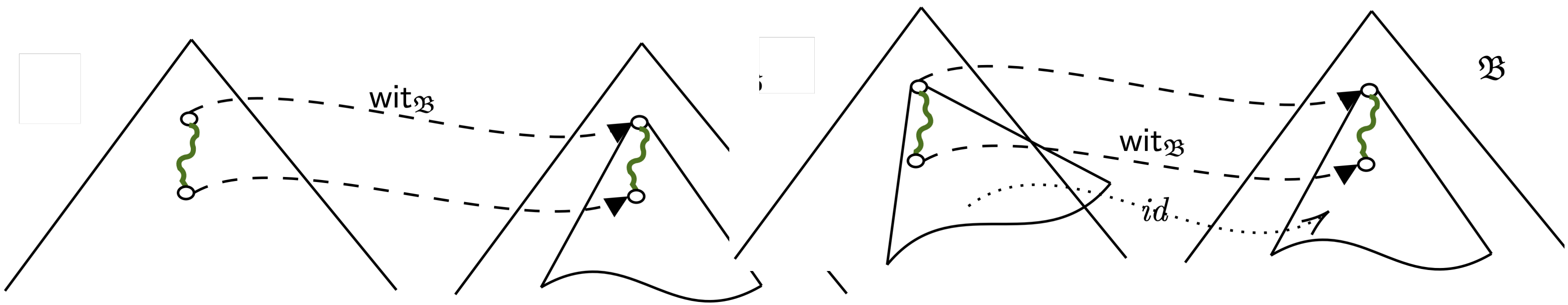
φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take φ, ψ with models $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ and amalgamate $\mathfrak{A}, \mathfrak{B}$ into a single $\mathfrak{U} \models \varphi \wedge \psi$.



Tree models



A novel "complete and repair" model construction method

Works beyond forward $\mathcal{GF}$. A good meta-heuristic.

5. Some initial results on the model checking problem:

Our contribution (Part II)

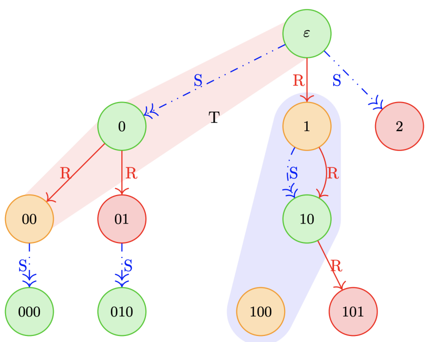
4. We prove that $\mathsf{L}_{\mathsf{pre}}$ and $\mathsf{G}_{\mathsf{pre}}, \mathsf{G}_{\mathsf{suf}}, \mathsf{G}_{\mathsf{inf}}$ enjoy CIP.

Proof method: interpolation via amalgamation

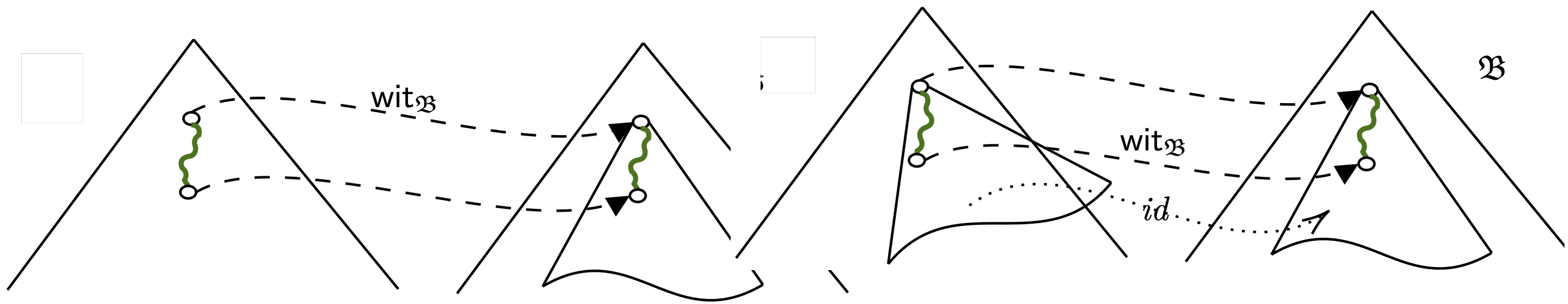
φ, ψ are L-jointly-consistent iff there are L-bisimilar (over common vocab.) models $\mathfrak{A} \models \varphi$ and $B \models \psi$.

Lemma: If for all L-jointly-consistent $\varphi, \psi \in \mathsf{L}$ there is an amalgam $\mathfrak{U}$ s.t. $\mathfrak{U} \models \varphi \wedge \psi$ then L has CIP.

So we take φ, ψ with models $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \psi$ and amalgamate $\mathfrak{A}, \mathfrak{B}$ into a single $\mathfrak{U} \models \varphi \wedge \psi$.



Tree models



A novel "complete and repair" model construction method

Works beyond forward $\mathcal{GF}$. A good meta-heuristic.

5. Some initial results on the model checking problem: PSPACE-c for $\mathsf{L}_{\mathsf{inf}}$, in PTIME for $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{pre}}$.

Conclusions

Conclusions

1. Status of L_{suf} and L_{inf}

Conclusions

1. Status of $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$

SAT

Conclusions

1. Status of L_{suf} and L_{inf}

SAT **FMP**

Conclusions

1. Status of $\mathrm{L}_{\mathrm{suf}}$ and $\mathrm{L}_{\mathrm{inf}}$

SAT	FMP	CIP
✓	✓	✗

Conclusions

1. Status of $\mathrm{L}_{\mathrm{suf}}$ and $\mathrm{L}_{\mathrm{inf}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

Conclusions

1. Status of L_{suf} and L_{inf}

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of L_{pre}

Conclusions

1. Status of L_{suf} and L_{inf}

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of L_{pre}

SAT
✓

Conclusions

1. Status of $\mathrm{L}_{\mathrm{suf}}$ and $\mathrm{L}_{\mathrm{inf}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of $\mathrm{L}_{\mathrm{pre}}$

SAT	FMP
✓	✓

Conclusions

1. Status of L_{suf} and L_{inf}

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of L_{pre}

SAT	FMP	CIP
✓	✓	✓

Conclusions

1. Status of $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of $\mathsf{L}_{\mathsf{pre}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

Conclusions

1. Status of L_{suf} and L_{inf}

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of L_{pre}

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

3. Status of G_{inf}, G_{suf} and G_{pre}

Conclusions

1. Status of $\mathrm{L}_{\mathrm{suf}}$ and $\mathrm{L}_{\mathrm{inf}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of $\mathrm{L}_{\mathrm{pre}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

3. Status of $\mathrm{G}_{\mathrm{inf}}$, $\mathrm{G}_{\mathrm{suf}}$ and $\mathrm{G}_{\mathrm{pre}}$

SAT
✓

Conclusions

1. Status of $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of $\mathsf{L}_{\mathsf{pre}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

3. Status of $\mathsf{G}_{\mathsf{inf}}$, $\mathsf{G}_{\mathsf{suf}}$ and $\mathsf{G}_{\mathsf{pre}}$

SAT	FMP
✓	✓

Conclusions

1. Status of $\mathsf{L}_{\mathsf{suf}}$ and $\mathsf{L}_{\mathsf{inf}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of $\mathsf{L}_{\mathsf{pre}}$

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

3. Status of $\mathsf{G}_{\mathsf{inf}}$, $\mathsf{G}_{\mathsf{suf}}$ and $\mathsf{G}_{\mathsf{pre}}$

SAT	FMP	CIP
✓	✓	✓

Conclusions

1. Status of L_{suf} and L_{inf}

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of L_{pre}

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

3. Status of G_{inf}, G_{suf} and G_{pre}

SAT	FMP	CIP	ŁTPT
✓	✓	✓	

Conclusions

1. Status of L_{suf} and L_{inf}

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of L_{pre}

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

3. Status of G_{inf}, G_{suf} and G_{pre}

SAT	FMP	CIP	ŁTPT
✓	✓	✓	

Open problems: interpolant existence problem, Łoś-Tarski for any equality-free fragment

Conclusions

1. Status of L_{suf} and L_{inf}

SAT	FMP	CIP	ŁTPT
✓	✓	✗	?

2. Status of L_{pre}

SAT	FMP	CIP	ŁTPT
✓	✓	✓	?

3. Status of G_{inf}, G_{suf} and G_{pre}

SAT	FMP	CIP	ŁTPT
✓	✓	✓	

Open problems: interpolant existence problem, Łoś-Tarski for any equality-free fragment

Icons that appear in the paper were downloaded from flaticon.com. No changes have been made.