# Guarded Fragments Meet Dynamic Logic The Story of Regular Guards

Bartosz Jan Bednarczyk bartek@cs.uni.wroc.pl Technische Universität Wien & Uniwersytet Wrocławski



TECHNISCHE UNIVERSITÄT WIEN Vienna | Austria



• Modal logic is everywhere:

• Modal logic is everywhere: philosophy,



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



• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.



SELECT CandID FROM Candidate WHERE Major = "Computer Science"

• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.



Syntax & Semantics via First-Order Structures

• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

 $\begin{array}{c|c} \varphi,\varphi'::=p & | \neg\varphi & | \varphi \land \varphi' & | \varphi \lor \varphi' & | \varphi \rightarrow \varphi' & | \varphi \leftrightarrow \varphi' & | \langle R \rangle \varphi & | [R] \varphi \\ \\ \mathfrak{A}:= & \begin{array}{c} p,q & R \rightarrow 4 \\ \hline 1 & R,S \\ \hline 3 & T \rightarrow 5 \end{array} \end{array}$ 

• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







3

SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures



• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

 $\begin{array}{c|c} \varphi,\varphi'::=p & | \neg\varphi & | \varphi \land \varphi' & | \varphi \lor \varphi' & | \varphi \rightarrow \varphi' & | \varphi \leftrightarrow \varphi' & | \langle R \rangle \varphi & | [R] \varphi \\ \\ \mathfrak{A}:= & \begin{array}{c} p,q & R \rightarrow 4 \\ \hline 1 & R,S \\ \hline 3 & T \rightarrow 5 \end{array} \end{array}$ 

• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

• 
$$(\mathfrak{A}, 1) \models p \land q \land \neg r$$



• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

- $(\mathfrak{A}, 1) \models p \land q \land \neg r$
- $(\mathfrak{A}, 1) \models \langle R \rangle p \land \langle R \rangle q$



• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

- $(\mathfrak{A}, 1) \models p \land q \land \neg r$
- $(\mathfrak{A}, \mathbf{1}) \models \langle R \rangle p \land \langle R \rangle q$
- $(\mathfrak{A}, 2) \models [R]q$



• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

- $(\mathfrak{A}, 1) \models p \land q \land \neg r$
- $(\mathfrak{A}, 1) \models \langle R \rangle p \land \langle R \rangle q$
- $(\mathfrak{A}, 2) \models [R]q$
- $(\mathfrak{A}, 1) \models \langle R \rangle (p \land [R] q)$



• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

 $\varphi, \varphi' ::= \mathbf{p} \mid \neg \varphi \mid \varphi \land \varphi' \mid \varphi \lor \varphi' \mid \varphi \to \varphi' \mid \varphi \leftrightarrow \varphi' \mid \langle \mathbf{R} \rangle \varphi \mid [\mathbf{R}] \varphi$ 

- $(\mathfrak{A}, 1) \models p \land q \land \neg r$
- $(\mathfrak{A}, 1) \models \langle R \rangle p \land \langle R \rangle q$
- (𝔄, 2) |= [R] q
- $(\mathfrak{A}, 1) \models \langle R \rangle (p \land [R] q)$



Multi-modal logic (with converse) translates into first-order logic via the standard translation.

• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

 $\varphi, \varphi' ::= \mathbf{p} \mid \neg \varphi \mid \varphi \land \varphi' \mid \varphi \lor \varphi' \mid \varphi \to \varphi' \mid \varphi \leftrightarrow \varphi' \mid \langle \mathbf{R} \rangle \varphi \mid [\mathbf{R}] \varphi$ 

- $(\mathfrak{A}, 1) \models p \land q \land \neg r$
- $(\mathfrak{A}, 1) \models \langle R \rangle p \land \langle R \rangle q$
- (𝔄, 2) |= [R] q
- $(\mathfrak{A}, 1) \models \langle R \rangle (p \land [R] q)$



Multi-modal logic (with converse) translates into first-order logic via the standard translation.  $\langle R \rangle ((p \land \neg q) \lor [\bar{S}] r)$ 

• Modal logic is everywhere: philosophy, temporal logic (CTL), description logic (ALC), and more.







SELECT CandID FROM Candidate WHERE Major = "Computer Science"

Syntax & Semantics via First-Order Structures

 $\varphi, \varphi' ::= \mathbf{p} \mid \neg \varphi \mid \varphi \land \varphi' \mid \varphi \lor \varphi' \mid \varphi \to \varphi' \mid \varphi \leftrightarrow \varphi' \mid \langle \mathbf{R} \rangle \varphi \mid [\mathbf{R}] \varphi$ 

- $(\mathfrak{A}, 1) \models p \land q \land \neg r$
- $(\mathfrak{A}, 1) \models \langle R \rangle p \land \langle R \rangle q$
- $(\mathfrak{A}, 2) \models [R]q$
- $(\mathfrak{A}, 1) \models \langle R \rangle (p \land [R] q)$



Multi-modal logic (with converse) translates into first-order logic via the standard translation.  $\langle R \rangle ((p \land \neg q) \lor [\bar{S}]r) \rightsquigarrow \varphi(x) := \exists y (R(x, y) \land [(p(y) \land \neg q(y)) \lor \forall z (S(z, y) \rightarrow r(z))])$ 

Bartosz "Bart" Bednarczyk

Guarded Fragments Meet Dynamic Logic

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM



MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM



#### MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

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

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM



#### MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

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

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM



#### MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

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

 $\varphi, \varphi' ::= \mathbf{R}(\bar{x}) \mid \neg \varphi \mid \varphi \land \varphi' \mid \exists x \varphi(x) \mid \forall x \varphi(x) \mid \forall \bar{x} \alpha(\bar{x}) \rightarrow \varphi(\bar{x}) \mid \exists \bar{x} \alpha(\bar{x}) \land \varphi(\bar{x})$ 

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM



#### MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

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

 $\varphi, \varphi' ::= \mathbf{R}(\bar{x}) \mid \neg \varphi \mid \varphi \land \varphi' \mid \exists x \varphi(x) \mid \forall x \varphi(x) \mid \forall \bar{x} \alpha(\bar{x}) \rightarrow \varphi(\bar{x}) \mid \exists \bar{x} \alpha(\bar{x}) \land \varphi(\bar{x})$ 

Example 1. Some artist admires only beekeepers

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

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM



#### MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

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

 $\varphi, \varphi' ::= \mathbf{R}(\bar{x}) \mid \neg \varphi \mid \varphi \land \varphi' \mid \exists x \varphi(x) \mid \forall x \varphi(x) \mid \forall \bar{x} \alpha(\bar{x}) \rightarrow \varphi(\bar{x}) \mid \exists \bar{x} \alpha(\bar{x}) \land \varphi(\bar{x})$ 

Example 1. Some artist admires only beekeepers

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

Example 2. Every artist envies every beekeeper he admires

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

HAJNAL ANDRÉKA, ISTVÁN NÉMETI and JOHAN VAN BENTHEM



#### MODAL LANGUAGES AND BOUNDED FRAGMENTS OF PREDICATE LOGIC

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

 $\varphi, \varphi' ::= \mathbf{R}(\bar{x}) \mid \neg \varphi \mid \varphi \land \varphi' \mid \exists x \varphi(x) \mid \forall x \varphi(x) \mid \forall \bar{x} \alpha(\bar{x}) \rightarrow \varphi(\bar{x}) \mid \exists \bar{x} \alpha(\bar{x}) \land \varphi(\bar{x})$ 

Example 1. Some artist admires only beekeepers

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

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

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

Bartosz "Bart" Bednarczyk Guarded Fragments Meet Dynamic Logic

• Recall our translation from modal logic to the guarded fragment.

• Recall our translation from modal logic to the guarded fragment.

 $\langle R \rangle \left( \left( p \land \neg q \right) \lor \left[ \bar{S} \right] r \right)$ 

• Recall our translation from modal logic to the guarded fragment.

$$\langle \mathrm{R} \rangle \left( (\mathrm{p} \land \neg \mathrm{q}) \lor \left[ \, \bar{\mathrm{S}} \, \right] \mathrm{r} \right) \rightsquigarrow \varphi(\mathbf{x}) := \exists \mathbf{y} \Big( \mathrm{R}(\mathbf{x}, \mathbf{y}) \land \Big[ (\mathrm{p}(\mathbf{y}) \land \neg \mathrm{q}(\mathbf{y})) \lor \forall \mathbf{z} \Big( \mathrm{S}(\mathbf{z}, \mathbf{y}) \to \mathrm{r}(\mathbf{z}) \Big) \Big] \Big)$$

• Recall our translation from modal logic to the guarded fragment.

$$\mathbb{R}\left(\left(\mathbf{p}\wedge\neg\mathbf{q}\right)\vee\left[\bar{\mathbf{S}}\right]\mathbf{r}\right)\rightsquigarrow \varphi(\mathbf{x}):=\exists y \left(\mathbb{R}(\mathbf{x},\mathbf{y})\wedge\left[\left(\mathbf{p}(\mathbf{y})\wedge\neg\mathbf{q}(\mathbf{y})\right)\vee\forall z \left(\mathbf{S}(\mathbf{z},\mathbf{y})\rightarrow\mathbf{r}(\mathbf{z})\right)\right]\right)$$

• What about other classical modal logics from the modal cube?



• Recall our translation from modal logic to the guarded fragment.

$$|\mathrm{R}\rangle\left((\mathrm{p}\wedge\neg\mathrm{q})\vee\left[\,\mathrm{ar{S}}\,\right]\mathrm{r}
ight)\rightsquigarrow arphi(x):=\exists y \Big(\mathrm{R}(x,y)\wedge\left[(\mathrm{p}(y)\wedge\neg\mathrm{q}(y))\vee\forall z\Big(\mathrm{S}(z,y)
ightarrow\mathrm{r}(z)\Big)\Big]\Big)$$

• What about other classical modal logics from the modal cube?

Not in GF!



• Recall our translation from modal logic to the guarded fragment.

 $\langle \mathrm{R} \rangle \left( (\mathrm{p} \land \neg \mathrm{q}) \lor \left[ \, \bar{\mathrm{S}} \, \right] \mathrm{r} \right) \rightsquigarrow \varphi(x) := \exists y \Big( \mathrm{R}(x, y) \land \left[ (\mathrm{p}(y) \land \neg \mathrm{q}(y)) \lor \forall z \Big( \mathrm{S}(z, y) \to \mathrm{r}(z) \Big) \right] \Big)$ 

• What about other classical modal logics from the modal cube?

Not in GF! It cannot express transitivity/equivalence/Euclidean/etc.



• Recall our translation from modal logic to the guarded fragment.

 $\langle \mathrm{R} \rangle \left( (\mathrm{p} \land \neg \mathrm{q}) \lor \left[ \, \bar{\mathrm{S}} \, \right] \mathrm{r} \right) \rightsquigarrow \varphi(x) := \exists y \Big( \mathrm{R}(x, y) \land \left[ (\mathrm{p}(y) \land \neg \mathrm{q}(y)) \lor \forall z \Big( \mathrm{S}(z, y) \to \mathrm{r}(z) \Big) \right] \Big)$ 

• What about other classical modal logics from the modal cube?

Not in GF! It cannot express transitivity/equivalence/Euclidean/etc.

The Two-Variable Guarded Fragment with Transitive Relations

H. Ganzinger, C. Meyer, and M. Veanes Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany



• Recall our translation from modal logic to the guarded fragment.

 $\langle \mathrm{R} \rangle \left( (\mathrm{p} \land \neg \mathrm{q}) \lor \left[ \, \bar{\mathrm{S}} \, \right] \mathrm{r} \right) \rightsquigarrow \varphi(x) := \exists y \Big( \mathrm{R}(x, y) \land \left[ (\mathrm{p}(y) \land \neg \mathrm{q}(y)) \lor \forall z \Big( \mathrm{S}(z, y) \to \mathrm{r}(z) \Big) \right] \Big)$ 

• What about other classical modal logics from the modal cube?

Not in GF! It cannot express transitivity/equivalence/Euclidean/etc.

The Two-Variable Guarded Fragment with Transitive Relations

H. Ganzinger, C. Meyer, and M. Veanes Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany

Even worse: the satisfiability problem for GF with transitive relations is undecidable!



• Recall our translation from modal logic to the guarded fragment.

 $\langle \mathrm{R} \rangle \left( (\mathrm{p} \land \neg \mathrm{q}) \lor \left[ \, \bar{\mathrm{S}} \, \right] \mathrm{r} \right) \rightsquigarrow \varphi(x) := \exists y \Big( \mathrm{R}(x, y) \land \left[ (\mathrm{p}(y) \land \neg \mathrm{q}(y)) \lor \forall z \Big( \mathrm{S}(z, y) \to \mathrm{r}(z) \Big) \Big] \Big)$ 

• What about other classical modal logics from the modal cube?

Not in GF! It cannot express transitivity/equivalence/Euclidean/etc.

The Two-Variable Guarded Fragment with Transitive Relations

H. Ganzinger, C. Meyer, and M. Veanes Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany



Even worse: the satisfiability problem for GF with transitive relations is undecidable!

**Guarded Fragment with Semantically Constrained Guards**
#### Standard Translation From Modal Logics to First-Order Logic

• Recall our translation from modal logic to the guarded fragment.

 $\langle \mathrm{R} \rangle \left( (\mathrm{p} \land \neg \mathrm{q}) \lor \left[ \, \bar{\mathrm{S}} \, \right] \mathrm{r} \right) \rightsquigarrow \varphi(x) := \exists y \Big( \mathrm{R}(x, y) \land \left[ (\mathrm{p}(y) \land \neg \mathrm{q}(y)) \lor \forall z \Big( \mathrm{S}(z, y) \to \mathrm{r}(z) \Big) \Big] \Big)$ 

• What about other classical modal logics from the modal cube?

Not in GF! It cannot express transitivity/equivalence/Euclidean/etc.

The Two-Variable Guarded Fragment with Transitive Relations

H. Ganzinger, C. Meyer, and M. Veanes Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany



Even worse: the satisfiability problem for GF with transitive relations is undecidable!

# **Guarded Fragment with Semantically Constrained Guards**

• Ganzinger et al. initiated the study of  $\mathcal{GF}$  where distinguished relations may appear only as guards.

## Standard Translation From Modal Logics to First-Order Logic

• Recall our translation from modal logic to the guarded fragment.

 $\langle \mathbf{R} \rangle \left( (\mathbf{p} \land \neg \mathbf{q}) \lor \left[ \mathbf{\bar{S}} \right] \mathbf{r} \right) \rightsquigarrow \varphi(\mathbf{x}) := \exists \mathbf{y} \left( \mathbf{R}(\mathbf{x}, \mathbf{y}) \land \left[ (\mathbf{p}(\mathbf{y}) \land \neg \mathbf{q}(\mathbf{y})) \lor \forall \mathbf{z} \left( \mathbf{S}(\mathbf{z}, \mathbf{y}) \rightarrow \mathbf{r}(\mathbf{z}) \right) \right] \right)$ 

• What about other classical modal logics from the modal cube?

Not in GF! It cannot express transitivity/equivalence/Euclidean/etc.

The Two-Variable Guarded Fragment with Transitive Relations

H. Ganzinger, C. Meyer, and M. Veanes Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany



Even worse: the satisfiability problem for GF with transitive relations is undecidable!

# **Guarded Fragment with Semantically Constrained Guards**

• Ganzinger et al. initiated the study of  $\mathcal{GF}$  where distinguished relations may appear only as guards. Dec. cases: transitive, equivalence (closure) guards







Bartosz "Bart" Bednarczyk









## Standard Translation From Modal Logics to First-Order Logic

• Recall our translation from modal logic to the guarded fragment.

 $\langle \mathbf{R} \rangle \left( (\mathbf{p} \land \neg \mathbf{q}) \lor \left[ \mathbf{\bar{S}} \right] \mathbf{r} \right) \rightsquigarrow \varphi(\mathbf{x}) := \exists \mathbf{y} \Big( \mathbf{R}(\mathbf{x}, \mathbf{y}) \land \left[ (\mathbf{p}(\mathbf{y}) \land \neg \mathbf{q}(\mathbf{y})) \lor \forall \mathbf{z} \Big( \mathbf{S}(\mathbf{z}, \mathbf{y}) \rightarrow \mathbf{r}(\mathbf{z}) \Big) \right] \Big)$ 

• What about other classical modal logics from the modal cube?

Not in GF! It cannot express transitivity/equivalence/Euclidean/etc.

The Two-Variable Guarded Fragment with Transitive Relations

H. Ganzinger, C. Meyer, and M. Veanes Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany



Even worse: the satisfiability problem for GF with transitive relations is undecidable!

# **Guarded Fragment with Semantically Constrained Guards**

• Ganzinger et al. initiated the study of  $\mathcal{GF}$  where distinguished relations may appear only as guards. Dec. cases: transitive, equivalence (closure) guards Undec. cases:  $\approx$  + expon., composition, conj. of TG







Bartosz "Bart" Bednarczyk









• Ganzinger et al. initiated the study of  $\mathcal{GF}$  where distinguished relations may appear only as guards.

• Ganzinger et al. initiated the study of  $\mathcal{GF}$  where distinguished relations may appear only as guards. Dec. cases: transitive, equivalence (closure) guards



























Bartosz "Bart" Bednarczyk Guarded Fragments Meet Dynamic Logic













Bartosz "Bart" Bednarczyk Guarded Fragments Meet Dynamic Logic



**Our Motivation** 



#### **Our Motivation**

• Goal: Provide an extension of GF that captures dynamic logic (ICPDL).



#### **Our Motivation**

• Goal: Provide an extension of GF that captures dynamic logic (ICPDL).

 $\begin{aligned} \mathsf{ICPDL} &= \mathsf{ML} + \langle \pi \cap \ldots \cap \pi' \rangle. \varphi \\ \pi \text{ - two-way regular expressions} \end{aligned}$ 

E.g.  $(\mathfrak{A}, 1) \models \langle (T \circ \overline{T}) \cap \mathbb{R}^* \rangle q$ 

Bartosz "Bart" Bednarczyk

 $\mathfrak{A} := \underbrace{\begin{array}{c} p, q \\ 1 \end{array}}_{q} R \xrightarrow{p} R \xrightarrow{q} 4 \xrightarrow{p} T$ 

Guarded Fragments Meet Dynamic Logic



#### **Our Motivation**

• Goal: Provide an extension of GF that captures dynamic logic (ICPDL).

Motivation I: Allowing for expressive navigation features á la RPQs to model graph database scenarios.

$$\begin{aligned} \mathsf{ICPDL} &= \mathsf{ML} + \langle \pi \cap \ldots \cap \pi' \rangle. \varphi \\ \pi \text{ - two-way regular expressions} \end{aligned}$$

E.g.  $(\mathfrak{A}, 1) \models \langle (T \circ \overline{T}) \cap \mathbb{R}^* \rangle q$ 





#### **Our Motivation**

- Goal: Provide an extension of GF that captures dynamic logic (ICPDL).
- Motivation I: Allowing for expressive navigation features  $\dot{a}$  la RPQs to model graph database scenarios. Motivation II: Improve state of the art. Limitation: undecidability of the logics with  $\approx$ .

- $\begin{aligned} \mathsf{ICPDL} &= \mathsf{ML} + \langle \pi \cap \ldots \cap \pi' \rangle.\varphi \\ \pi \text{ two-way regular expressions} \end{aligned}$
- E.g.  $(\mathfrak{A}, 1) \models \langle (T \circ \overline{T}) \cap \mathbb{R}^* \rangle q$





#### **Our Motivation**

• Goal: Provide an extension of GF that captures dynamic logic (ICPDL).

Motivation I: Allowing for expressive navigation features á la RPQs to model graph database scenarios. Motivation II: Improve state of the art. Limitation: undecidability of the logics with  $\approx$ . Motivation III: Some positive results are only for the two-variable fragments.

 $\mathsf{ICPDL} = \mathsf{ML} + \langle \pi \cap \ldots \cap \pi' \rangle.\varphi$ \$\pi\$ a - two-way regular expressions

E.g.  $(\mathfrak{A}, 1) \models \langle (T \circ \overline{T}) \cap \mathbb{R}^* \rangle q$ 

 $\mathfrak{A} := \underbrace{\begin{array}{c} p, q \\ 1 \end{array}}_{q} R \xrightarrow{p} q \xrightarrow{q} q \xrightarrow{q} R \xrightarrow{q} q \xrightarrow{q} x \xrightarrow{q} q \xrightarrow{$ 



#### **Our Motivation**

• Goal: Provide an extension of GF that captures dynamic logic (ICPDL).

Motivation I: Allowing for expressive navigation features  $\dot{a}$  la RPQs to model graph database scenarios. Motivation II: Improve state of the art. Limitation: undecidability of the logics with  $\approx$ .

Motivation III: Some positive results are only for the two-variable fragments.

Motivation IV: Provide a more high-level proof capturing many variants of GF in a uniform way.

ICPDL = ML +  $\langle \pi \cap ... \cap \pi' \rangle$ . $\varphi$  $\pi$  - two-way regular expressions

E.g.  $(\mathfrak{A}, 1) \models \langle (T \circ \overline{T}) \cap R^* \rangle q$ 

 $\mathfrak{A} := \underbrace{\begin{array}{c} p, q \\ 1 \end{array}}_{q} R \xrightarrow{p} R \xrightarrow{q} 4 \mathcal{P} T$ 

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{\rm FO}$  by

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

• Example 1:

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

• Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$ 

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

• Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

• Example 1:  $\langle (T \circ \overline{T}) \cap \mathbb{R}^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap \mathbb{R}^*) (xy) \land q(y)$ 

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (\mathrm{T} \circ \overline{\mathrm{T}}) \cap \mathrm{R}^* \rangle \mathrm{q}$  translates to  $\exists y \ ((\mathrm{T} \circ \overline{\mathrm{T}}) \cap \mathrm{R}^*) \ (xy) \land \mathrm{q}(y)$
- Example 2: All R-reachable elements are B-connected:

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy \, \mathbb{R}^*(xy) \to \mathbb{B}(xy)$

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]
- $\bullet\,$  Captures transitivity of via  $\mathrm{R}^+$  ,

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy \, \mathbb{R}^*(xy) \to \mathbb{B}(xy)$  [not in  $\mu \mathsf{GF}$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \bar{R})^*$ ,

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \overline{R})^*$ , composition and more.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (\mathrm{T} \circ \overline{\mathrm{T}}) \cap \mathrm{R}^* \rangle q$  translates to  $\exists y ((\mathrm{T} \circ \overline{\mathrm{T}}) \cap \mathrm{R}^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \bar{R})^*$ , composition and more.
- Built-in support for regular path queries.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy \, \mathbb{R}^*(xy) \to \mathbb{B}(xy)$  [not in  $\mu \mathsf{GF}$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \bar{R})^*$ , composition and more.
- Built-in support for regular path queries.
- Due to the presence of  $\cap$  we generalize GF with conjunction of transitive guards.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \overline{R})^*$ , composition and more.
- Built-in support for regular path queries.
- Due to the presence of  $\cap$  we generalize GF with conjunction of transitive guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \overline{R})^*$ , composition and more.
- Built-in support for regular path queries.
- Due to the presence of  $\cap$  we generalize GF with conjunction of transitive guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea:

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \overline{R})^*$ , composition and more.
- Built-in support for regular path queries.
- Due to the presence of  $\cap$  we generalize GF with conjunction of transitive guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is 2ExPTIME-complete.

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy R^*(xy) \rightarrow B(xy)$  [not in  $\mu GF$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \overline{R})^*$ , composition and more.
- Built-in support for regular path queries.
- Due to the presence of  $\cap$  we generalize GF with conjunction of transitive guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is 2ExpTIME-complete.

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

- Example 1:  $\langle (T \circ \overline{T}) \cap R^* \rangle q$  translates to  $\exists y ((T \circ \overline{T}) \cap R^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy \, \mathbb{R}^*(xy) \to \mathbb{B}(xy)$  [not in  $\mu \mathsf{GF}$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \bar{R})^*$ , composition and more.
- Built-in support for regular path queries.
- Due to the presence of  $\cap$  we generalize GF with conjunction of transitive guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

**Theorem** (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

- Example 1:  $\langle (\mathrm{T} \circ \overline{\mathrm{T}}) \cap \mathrm{R}^* \rangle q$  translates to  $\exists y ((\mathrm{T} \circ \overline{\mathrm{T}}) \cap \mathrm{R}^*) (xy) \land q(y)$
- Example 2: All R-reachable elements are B-connected:  $\forall xy \, \mathbb{R}^*(xy) \to \mathbb{B}(xy)$  [not in  $\mu \mathsf{GF}$ ]
- Captures transitivity of via  $R^+$ , equivalence relations via  $(R + \bar{R})^*$ , composition and more.
- Built-in support for regular path queries.
- Due to the presence of  $\cap$  we generalize GF with conjunction of transitive guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

**Theorem** (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Significantly improves prior results by Gottlob&Pieris&Tendera from ICALP 2013.


Bartosz "Bart" Bednarczyk Guarded Fragments Meet Dynamic Logic

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of: •  $\forall x_1 \lambda(x_1)$ 

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of: •  $\forall x_1 \lambda(x_1)$  [the general universal conjunct]

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$  [the general universal conjunct]
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$  [the general universal conjunct]
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$

[∃<sup>FO</sup>-conjuncts]

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$

[the general universal conjunct]

[∃<sup>FO</sup>-conjuncts]

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$

[the general universal conjunct] [∃<sup>FO</sup>-conjuncts] [∀<sup>FO</sup>-conjuncts]

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1x_2) \land \phi_i(x_1x_2)$

[the general universal conjunct] [∃<sup>FO</sup>-conjuncts]

 $[\forall^{FO}\text{-conjuncts}]$ 

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1x_2) \land \phi_i(x_1x_2)$

[the general universal conjunct] [∃<sup>FO</sup>-conjuncts] [∀<sup>FO</sup>-conjuncts]

 $[\exists^{reg}-conjuncts]$ 

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1x_2) \land \phi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct] [∃<sup>FO</sup>-conjuncts] [∀<sup>FO</sup>-conjuncts]

 $[\exists^{\mathsf{reg}}\mathsf{-}\mathsf{conjuncts}]$ 

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1x_2) \land \phi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1x_2) \land \phi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ ,

[the general universal conjunct]  $[\exists^{FO}$ -conjuncts]  $[\forall^{FO}$ -conjuncts]  $[\exists^{reg}$ -conjuncts]  $[\forall^{reg}$ -conjuncts]

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# Atomic One-Types and Two-Types

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

### Atomic One-Types and Two-Types

• 1-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ .  $(\alpha_{\sigma})$ 

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

### Atomic One-Types and Two-Types

- 1-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ .  $(\alpha_{\sigma})$
- 2-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ ,  $x_2$ . ( $\beta_{\sigma}$ )

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

## Atomic One-Types and Two-Types

- 1-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ .  $(\alpha_{\sigma})$
- 2-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ ,  $x_2$ . ( $\beta_{\sigma}$ )

Intuition: 1-types are colours of elements, and 2-types are colours of edges.

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

## Atomic One-Types and Two-Types

- 1-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ .  $(\alpha_{\sigma})$
- 2-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ ,  $x_2$ . ( $\beta_{\sigma}$ )

Intuition: 1-types are colours of elements, and 2-types are colours of edges.

Note: Given  $\varphi$  we have  $|\alpha_{\varphi}| \in O(2^{|\varphi|})$  but  $|\beta_{\varphi}| \in O(2^{2^{|\varphi|}})$  [reason: arbitrary arity symbols!].

We say that  $\varphi \in \mathsf{RGF}^2$  is in Scott's normal form if it is a conjunction of:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# Atomic One-Types and Two-Types

- 1-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ .  $(\alpha_{\sigma})$
- 2-type over  $\sigma$  = maximal satisfiable conjunction of  $\sigma$ -literals involving  $x_1$ ,  $x_2$ . ( $\beta_{\sigma}$ )

Intuition: 1-types are colours of elements, and 2-types are colours of edges. Note: Given  $\varphi$  we have  $|\alpha_{\varphi}| \in O(2^{|\varphi|})$  but  $|\beta_{\varphi}| \in O(2^{2^{|\varphi|}})$  [reason: arbitrary arity symbols!]. Usually we rephrase satisfiaction of  $\varphi$  in terms of realizable types.



•  $\forall x_1 \lambda(x_1)$  [the general universal conjunct]

•  $\forall x_1 \lambda(x_1)$ 

[the general universal conjunct]

•  $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$ 

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$

[the general universal conjunct] [∃<sup>FO</sup>-conjuncts]

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$

[the general universal conjunct] [∃<sup>FO</sup>-conjuncts]

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$

[the general universal conjunct]  $[\exists^{FO}$ -conjuncts]  $[\forall^{FO}$ -conjuncts]

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ ,

[the general universal conjunct]
[∃<sup>FO</sup>-conjuncts]
[∀<sup>FO</sup>-conjuncts]
[∃<sup>reg</sup>-conjuncts]
[∀<sup>reg</sup>-conjuncts]

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# From RGF<sup>2</sup> to ICPDL.
- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# **From** RGF<sup>2</sup> **to** ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types.

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# **From** RGF<sup>2</sup> **to** ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains:

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# **From** RGF<sup>2</sup> **to** ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# From RGF<sup>2</sup> to ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

**2.** Fresh binary  $B_{\beta}$  per each 2-type  $\beta$  from  $\beta_{\varphi}$  restricted to  $\sigma_{FO}$ 

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# From RGF<sup>2</sup> to ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

**2.** Fresh binary  $B_{\beta}$  per each 2-type  $\beta$  from  $\beta_{\varphi}$  restricted to  $\sigma_{FO}$  + all binary predicates in  $\sigma_{reg}$  from  $\varphi$ . Translation almost easy, e.g.  $\forall x_1 \gamma(x_1) \rightarrow \exists x_2 \pi(x_1 x_2) \land \phi(x_1 x_2) \mapsto$ 

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# From RGF<sup>2</sup> to ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

**2.** Fresh binary  $B_{\beta}$  per each 2-type  $\beta$  from  $\beta_{\varphi}$  restricted to  $\sigma_{FO}$  + all binary predicates in  $\sigma_{reg}$  from  $\varphi$ . Translation almost easy, e.g.  $\forall x_1 \gamma(x_1) \rightarrow \exists x_2 \pi(x_1 x_2) \land \phi(x_1 x_2) \mapsto [\star] (\forall_{\alpha \models \gamma} U_{\alpha} \rightarrow \langle \pi \cap \cup_{\beta \models \phi} B_{\beta} \rangle. \top)$ 

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

# From RGF<sup>2</sup> to ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

**2.** Fresh binary  $B_{\beta}$  per each 2-type  $\beta$  from  $\beta_{\varphi}$  restricted to  $\sigma_{FO}$  + all binary predicates in  $\sigma_{reg}$  from  $\varphi$ . Translation almost easy, e.g.  $\forall x_1 \gamma(x_1) \rightarrow \exists x_2 \pi(x_1 x_2) \land \phi(x_1 x_2) \mapsto [\star] (\forall_{\alpha \models \gamma} U_{\alpha} \rightarrow \langle \pi \cap \cup_{\beta \models \phi} B_{\beta} \rangle. \top)$ 



- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

Bartosz "Bart" Bednarczyk

# **From** RGF<sup>2</sup> **to** ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

**2.** Fresh binary  $B_{\beta}$  per each 2-type  $\beta$  from  $\beta_{\varphi}$  restricted to  $\sigma_{FO}$  + all binary predicates in  $\sigma_{reg}$  from  $\varphi$ . Translation almost easy, e.g.  $\forall x_1 \gamma(x_1) \rightarrow \exists x_2 \pi(x_1 x_2) \land \phi(x_1 x_2) \mapsto [\star] (\forall_{\alpha \models \gamma} U_{\alpha} \rightarrow \langle \pi \cap \cup_{\beta \models \phi} B_{\beta} \rangle. \top)$ 





Guarded Fragments Meet Dynamic Logic

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1x_2) \land \phi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

Bartosz "Bart" Bednarczyk

# **From** RGF<sup>2</sup> **to** ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

**2.** Fresh binary  $B_{\beta}$  per each 2-type  $\beta$  from  $\beta_{\varphi}$  restricted to  $\sigma_{FO}$  + all binary predicates in  $\sigma_{reg}$  from  $\varphi$ . Translation almost easy, e.g.  $\forall x_1 \gamma(x_1) \rightarrow \exists x_2 \pi(x_1 x_2) \land \phi(x_1 x_2) \mapsto [\star] (\forall_{\alpha \models \gamma} U_{\alpha} \rightarrow \langle \pi \cap \cup_{\beta \models \phi} B_{\beta} \rangle. \top)$ 





Guarded Fragments Meet Dynamic Logic

- $\forall x_1 \lambda(x_1)$
- $\forall x_1 \eta_i(x_1) \rightarrow \exists x_2 \vartheta_i(x_1x_2) \land \psi_i(x_1x_2)$
- $\forall x_1 \forall x_2 \eta_i(x_1 x_2) \rightarrow \psi_i(x_1 x_2)$
- $\forall x_1 \gamma_i(x_1) \rightarrow \exists x_2 \pi_i(x_1 x_2) \land \phi_i(x_1 x_2)$
- $\forall x_1 \forall x_2 \pi_i(x_1 x_2) \rightarrow \phi_i(x_1 x_2),$

where all  $\pi_i$  are RPQ with  $\cap$ , and all other formulae are quantifier-free and over  $\sigma_{FO}$ .

Bartosz "Bart" Bednarczyk

# **From** RGF<sup>2</sup> **to** ICPDL.

Take  $\varphi$ . If satisfiable,  $\varphi$  has a model realizing at most exponentially many 2-types. Our vocabulary contains: **1.** Fresh unary  $U_{\alpha}$  per each 1-type  $\alpha$  from  $\alpha_{\varphi}$ .

**2.** Fresh binary  $B_{\beta}$  per each 2-type  $\beta$  from  $\beta_{\varphi}$  restricted to  $\sigma_{FO}$  + all binary predicates in  $\sigma_{reg}$  from  $\varphi$ . Translation almost easy, e.g.  $\forall x_1 \gamma(x_1) \rightarrow \exists x_2 \pi(x_1 x_2) \land \phi(x_1 x_2) \mapsto [\star] (\forall_{\alpha \models \gamma} U_{\alpha} \rightarrow \langle \pi \cap \cup_{\beta \models \phi} B_{\beta} \rangle. \top)$ Main challenge: ensure the assignment of missing 2-types.





Guarded Fragments Meet Dynamic Logic

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{\rm reg}$  in RGF<sup>2</sup> and  $\varphi_{\rm GF}$  in GF as follows:

•  $\varphi_{\rm GF}$  contains all  $\exists^{\rm FO}$  and  $\forall^{\rm FO}$  conjuncts of  $\varphi$ .

- Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.
- Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:
  - $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
  - $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

**Correctness Proof: The Fusion** 

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

## **Correctness Proof: The Fusion**

• Take models  $\mathfrak{A} \models \varphi_{\mathsf{reg}}$  and  $\mathfrak{A} \models \varphi_{\mathsf{GF}}$ .

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

### **Correctness Proof: The Fusion**

- Take models  $\mathfrak{A} \models \varphi_{\mathsf{reg}}$  and  $\mathfrak{A} \models \varphi_{\mathsf{GF}}$ .
- Equilize their domains by Löwenheim Skolem (w.l.o.g. to  $\aleph_0$ ).

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

### **Correctness Proof: The Fusion**

- Take models  $\mathfrak{A} \models \varphi_{\mathsf{reg}}$  and  $\mathfrak{A} \models \varphi_{\mathsf{GF}}$ .
- Equilize their domains by Löwenheim Skolem (w.l.o.g. to  $\aleph_0$ ).
- We next ensure that for each 1-type  $\alpha$  in  $\alpha_{\varphi}$ :

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

### **Correctness Proof: The Fusion**

- Take models  $\mathfrak{A} \models \varphi_{\mathsf{reg}}$  and  $\mathfrak{A} \models \varphi_{\mathsf{GF}}$ .
- Equilize their domains by Löwenheim Skolem (w.l.o.g. to  $\aleph_0$ ).
- We next ensure that for each 1-type  $\alpha$  in  $\alpha_{\varphi}$ :

 $\#\alpha$  in  $\mathfrak{A}$  is equal to  $\#\alpha$  in  $\mathfrak{B}$ .

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

# **Correctness Proof: The Fusion**

- Take models  $\mathfrak{A} \models \varphi_{\mathsf{reg}}$  and  $\mathfrak{A} \models \varphi_{\mathsf{GF}}$ .
- Equilize their domains by Löwenheim Skolem (w.l.o.g. to  $\aleph_0$ ).
- We next ensure that for each 1-type  $\alpha$  in  $\alpha_{\varphi}$ :

 $\#\alpha$  in  $\mathfrak{A}$  is equal to  $\#\alpha$  in  $\mathfrak{B}$ .

• Form a two-dimensional grid  $\mathfrak{C}$ . Each row (column) isomorphic to  $\mathfrak{A}(\mathfrak{B})$ .

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{\rm reg}$  and  $\varphi_{\rm FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

<b>Correctness Proof: The Fusion</b>	0	0	0	0	0
<ul> <li>Take models 𝔅 ⊨ φ<sub>reg</sub> and 𝔅 ⊨ φ<sub>GF</sub>.</li> <li>Equilize their domains by Löwenheim Skolem (w.l.o.g. to ℵ₀).</li> <li>We next ensure that for each 1-type α in α<sub>φ</sub>: #α in 𝔅 is equal to #α in 𝔅.</li> <li>Form a two-dimensional grid 𝔅. Each row (column) isomorphic to 𝔅 (𝔅).</li> </ul>		0	0	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0

•  $\mathfrak{C}$  is almost a model of  $\varphi$ , but some elements may miss witnesses for  $\exists^{\mathsf{FO}}$ -conjuncts.

Step I: Transform your input  $\varphi$  in RGF to Scott's normal form.

Step II: Construct  $\varphi_{reg}$  in RGF<sup>2</sup> and  $\varphi_{GF}$  in GF as follows:

- $\varphi_{\mathsf{GF}}$  contains all  $\exists^{\mathsf{FO}}$  and  $\forall^{\mathsf{FO}}$  conjuncts of  $\varphi$ .
- $\varphi_{\text{reg}}$  contains all  $\exists^{\text{reg}}$  and  $\forall^{\text{reg}}$  conjuncts of  $\varphi$  and the two-variable versions of  $\exists^{\text{FO}}$  and  $\forall^{\text{FO}}$  conjuncts.
- Synchronization: require that models of  $\varphi_{reg}$  and  $\varphi_{FO}$  realize the same sets of 1-types and 2-types.

Step III: Test both  $\varphi_{reg}$  and  $\varphi_{GF}$  for satisfiability (this gives the right upper bound). Viola!

<b>Correctness Proof: The Fusion</b>	0	0	0	0	0
• Take models $\mathfrak{A} \models \varphi_{reg}$ and $\mathfrak{A} \models \varphi_{GF}$ .		0	0	0	0
• Equilize their domains by Löwenheim Skolem (w.l.o.g. to $\aleph_0$ ).		0	0	$\mathbf{O}$	
• We next ensure that for each 1-type $lpha$ in $oldsymbol{lpha}_{arphi}$ :		0	0	0	0
$\# \alpha$ in $\mathfrak{A}$ is equal to $\# \alpha$ in $\mathfrak{B}$ .	0	0	0	0	0
• Form a two-dimensional grid $\mathfrak{C}$ . Each row (column) isomorphic to $\mathfrak{A}$ ( $\mathfrak{B}$ ).		0	0	0	0

- $\mathfrak{C}$  is almost a model of  $\varphi$ , but some elements may miss witnesses for  $\exists^{\mathsf{FO}}$ -conjuncts.
- Provide them with a circular-witnessing-scheme á la Grädel & Kolaitis & Vardi.

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

RGF is the least extension of GF over  $\sigma_{\rm FO}$  by

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

**Theorem** (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

**Theorem** (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea:

Consider disjoint vocabularies  $\sigma_{FO}$  and  $\sigma_{reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E{\rm XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

**Theorem** (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

**Theorem** (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

**Theorem** (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Significantly improves prior results by Gottlob&Pieris&Tendera from ICALP 2013.
Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

**Theorem** (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

**Theorem** (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Significantly improves prior results by Gottlob&Pieris&Tendera from ICALP 2013.

**Theorem** (under submission with E. Kieroński)

There exists an EXPSPACE-complete sublogic of  $\mathsf{RGF}[\cdot^+, \cdot^*]$ .

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

**Theorem** (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

#### **Theorem** (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Significantly improves prior results by Gottlob&Pieris&Tendera from ICALP 2013.

**Theorem** (under submission with E. Kieroński)

There exists an EXPSPACE-complete sublogic of  $\mathsf{RGF}[\cdot^+, \cdot^*]$ .

Combines ideas from the fluted fragment and one-way guards.

Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

### Theorem (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Significantly improves prior results by Gottlob&Pieris&Tendera from ICALP 2013.

Theorem (under submission with E. Kieroński)

There exists an ExpSpace-complete sublogic of  $RGF[\cdot^+, \cdot^*]$ .

Combines ideas from the fluted fragment and one-way guards.



Consider disjoint vocabularies  $\sigma_{\rm FO}$  and  $\sigma_{\rm reg}$ .

RGF is the least extension of GF over  $\sigma_{FO}$  by allowing ICPDL programs  $\pi(xy)$  over  $\sigma_{reg}$  as binary guards.

Theorem (under submission with E. Kieroński)

The satisfiability problem for RGF is  $2E\mathrm{XPTIME}\text{-}complete.$ 

Proof idea: solve  $RGF^2$  via a reduction to ICPDL and lift it to the full RGF via a fusion.

# Theorem (under submission with E. Kieroński)

Finite&Unrestricted Conjunctive Query Entailment Problems is undecidable for a tiny fragment of RGF.

Significantly improves prior results by Gottlob&Pieris&Tendera from ICALP 2013.

Theorem (under submission with E. Kieroński)

There exists an ExpSpace-complete sublogic of  $RGF[\cdot^+, \cdot^*]$ .

Combines ideas from the fluted fragment and one-way guards.

# **Questions?**

