Guarded Negation in query languages

Vince Bárány (TU Darmstadt)

joint work with

Balder ten Cate (UC Santa Cruz) &

Martin Otto (TU Darmstadt)

Oxford June 2012

Why is modal logic so robustly decidable?

"Why is modal logic so robustly decidable?" [Vardi '96]

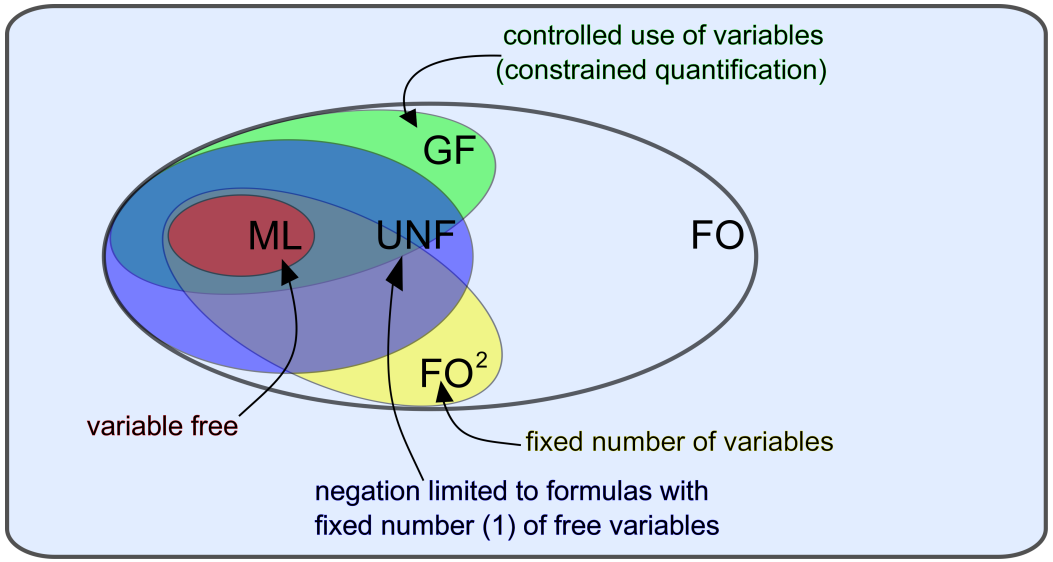
- tree model property
- translation into MSO / tree automata
- finite model property

Fragments of FO embedding ML

✗ 2-variable fragment

✓ *guarded fragment* [Andréka - van Benthem - Németi '95-98]

✓ *unary negation fragment* [ten Cate - Segoufin STACS'11]



logics of interest in computer science

finite variable fragments

- FO^2 has FMP (exp. size models) and SAT is NEXPTIME-complete
- FO^3 undecidable (already the prefix class $\forall\exists\forall$ is)

ML: modal logic K

FMP, invariant under bisimulation, tree-model prop., interpolation
SAT is PSPACE-c. [Ladner], ML $\equiv$ FO$/\sim$ [van Benthem/Rosen/(Otto)]

CS applications (verification, TLs, DBs, KBs, DLs, XML)
call for more expressive extensions: recursion, fixpoints, counting, etc.

ML embeds into FO^2 but is much better behaved under extensions

- μ-calculus = ML + fixpoints is equally well behaved
- FO^2 + TC $\rightsquigarrow$ highly undecidable [Grädel-Otto-Rosen]
- nevertheless, FO^2 + counting is still decidable

Logics of trees

Recall GFO, GFP characteristics

- invariance under guarded bisimulation
- guarded unravelling
- Tree-like Model Property

Similarly for UNFO, UNFP (and ML and temporal logics up to L_μ).

UNFO debut in [ten Cate - Marx '07] as an alternative for XPath

- GNFP $\supset$ UNFP $\supset$ μ-calculus $\supset$ most branching time logics.
- On ranked trees and XML trees UNFP, UNFP^2, μ-calculus define the regular languages.
- On XML trees UNFO/ UNFP capture CoreXPath / RegularXPath

Logics of databases

Finite model reasoning

- GFO, UNFO, FO^2 have the Finite Model Property
- FO^2+counting and GFO proposed as basic description logics
- finite controllability of query answering: $\varphi \models q$
  with $\varphi \in$ GFO and $q \in UCQ$
- Beth property of GFO and Craig interpolation for UNFO
  relevant for e.g. query answering

plenty of shortcomings

many interesting integrity constraints / role constructs are inexpressible

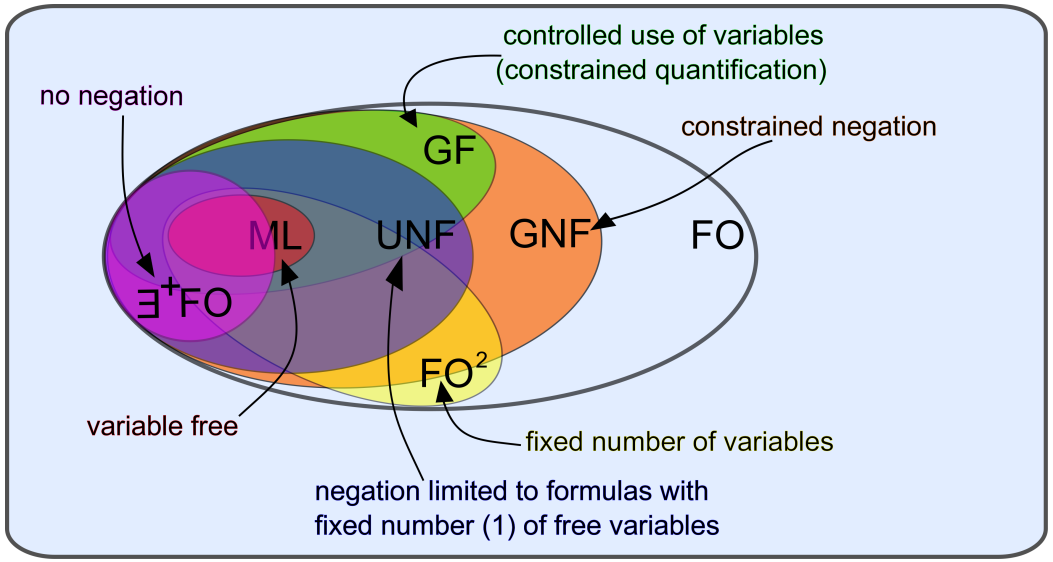
Guarded negation fragments of FO and of LFP

Idea: constrain the use of negation instead of quantification.

Common extension of

- the guarded fragments
- the positive existential fragment $\exists^+$FO
- the unary negation fragments [ten Cate-Segoufin STACS'11]

  which extend $\exists^+$FO, CoreXPath/RegularXPath, Data tree patterns,
  $\mathcal{ALCI}$ query containment, modal μ-calculus (with backward modalities), Monadic DataLog



Guarded Negation Fragment vs Guarded Fragment

Guarded fragment (GFO) [$\alpha(\bar{x}\bar{y})$ is atomic or equality)]

$$\varphi ::= \quad R(\bar{x}) \mid x = y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists\bar{y}.\alpha(\bar{x}\bar{y}) \wedge \varphi(\bar{x}\bar{y}) \mid \forall\bar{y}.\alpha(\bar{x}\bar{y}) \rightarrow \varphi(\bar{x}\bar{y})$$

can't express existence of (unguarded) cycles

Guarded Negation Fragment vs Guarded Fragment

Guarded fragment (GFO) [$\alpha(\bar{x}\bar{y})$ is atomic or equality)]

$$\varphi ::= \quad R(\bar{x}) \mid x = y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists\bar{y}.\alpha(\bar{x}\bar{y}) \wedge \varphi(\bar{x}\bar{y}) \mid \forall\bar{y}.\alpha(\bar{x}\bar{y}) \rightarrow \varphi(\bar{x}\bar{y})$$

can't express existence of (unguarded) cycles

Guarded negation fragment (GNFO) [detto]

$$\varphi ::= \quad R(\bar{x}) \mid x = y \mid \exists x\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha(\bar{x}\bar{y}) \wedge \neg\varphi(\bar{y})$$

arbitrary positive existential formulas, but $\neg$ (and $\forall$) only under a guard

Guarded Negation Fragment vs Guarded Fragment

Guarded fragment (GFO) [$\alpha(\bar{x}\bar{y})$ is atomic or equality)]

$$\varphi ::= \quad R(\bar{x}) \mid x = y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists\bar{y}.\alpha(\bar{x}\bar{y}) \wedge \varphi(\bar{x}\bar{y}) \mid \forall\bar{y}.\alpha(\bar{x}\bar{y}) \rightarrow \varphi(\bar{x}\bar{y})$$

can't express existence of (unguarded) cycles

Guarded negation fragment (GNFO) [detto]

$$\varphi ::= \quad R(\bar{x}) \mid x = y \mid \exists x\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha(\bar{x}\bar{y}) \wedge \neg\varphi(\bar{y})$$

arbitrary positive existential formulas, but $\neg$ (and $\forall$) only under a guard

Prop. Every GFO sentence is equivalent to a GNFO sentence.

Not true for formulas in general: $\neg R(x, y)$ is in GFO but not in GNFO.

Disjunctive Normal Form for GNFO and Width

a good way to understand GNFO formulas...

DNF $\bigvee_i \phi_i$ a disjunction of ϕ_i generated by

$$\begin{array}{rcl} \phi & ::= & \exists x_1, \ldots, x_n(\zeta_1 \wedge \cdots \wedge \zeta_m) \\ \zeta & ::= & R(\bar{x}) \mid (x = y) \mid R(\bar{x}) \wedge \neg\phi(\bar{x}) \end{array}$$

- GNFO $\rightsquigarrow$ DNF (possibly exponential)
- UCQ are DNF without negation

width number of variables occurring (free or bound) in the DNF

Guarded Negation Fixpoint Logic (GNFP)

Syntax

$$\phi \quad ::= \quad R(\bar{x}) \mid \mathsf{x{=}y} \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \exists x\, \phi \mid \alpha(\bar{x}\bar{y}) \wedge \neg\phi(\bar{x}) \mid$$

$$Z(\bar{x}) \mid [\lambda Z, \bar{z}.\ \underbrace{\mathtt{guarded}_\sigma(\bar{z}) \wedge \phi(\bar{Y}, Z, \bar{z})}_{\text{explicit guarding of free vars } \bar{z}}](\bar{x}) \qquad (\lambda \in \{\mu, \nu\})$$

like GFP

- fixpoint vars Z occur only positively in the scope of a binding μ or ν
- no first-order params
- fixpoint vars cannot stand as guard
- duality, negation nf. *(despite earlier claims to the contrary)*

unlike GFP

- explicitly guarded fixpoint formulas (wlog. assumed in GFP)

GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $Z : M \approx_{GN}^{(k)} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$

GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $\quad Z : M \approx_{GN}^{(k)} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$

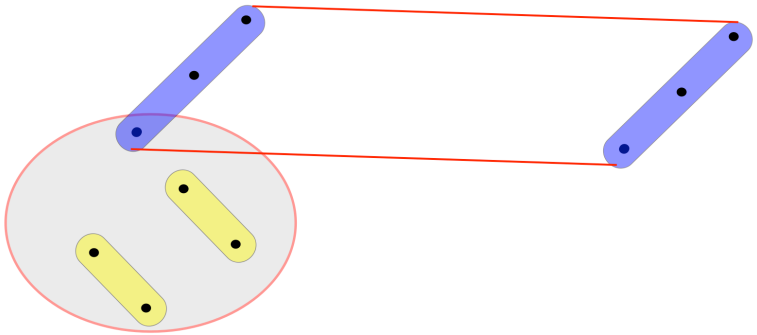
GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $\quad Z : M \approx_{GN}^{(k)} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$



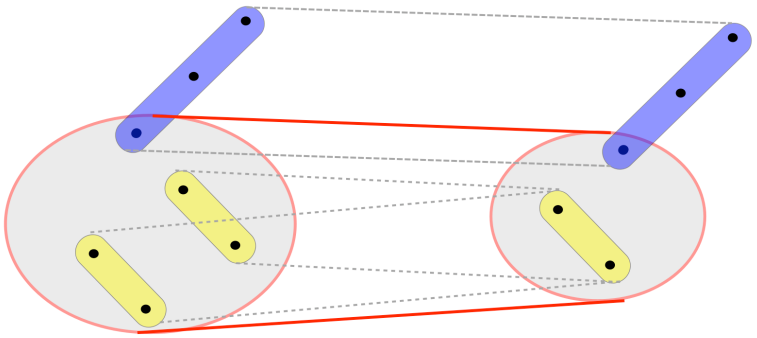
GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $\quad Z : M \approx_{GN}^{(k)} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$



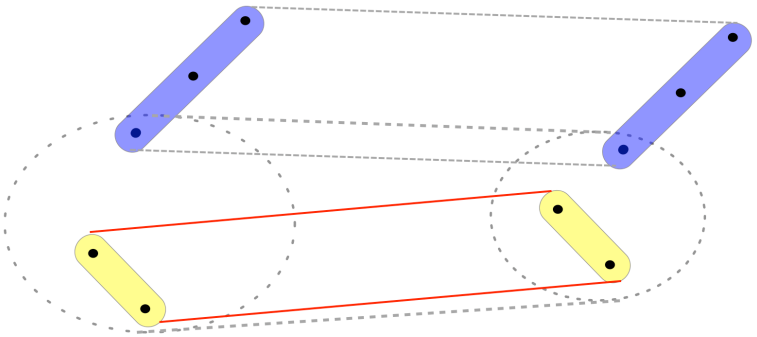
GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $\quad Z : M \approx_{GN}^{(k)} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$



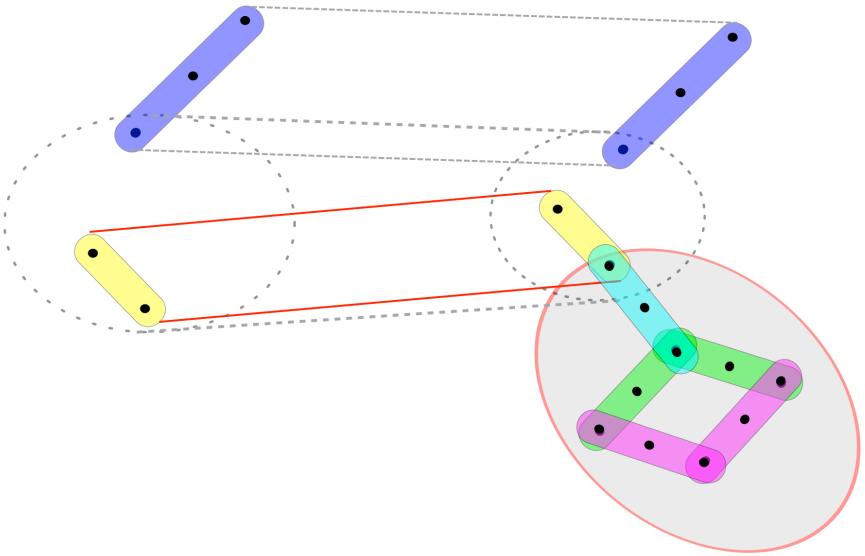
GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $Z : M \approx_{GN}^{(k)} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$



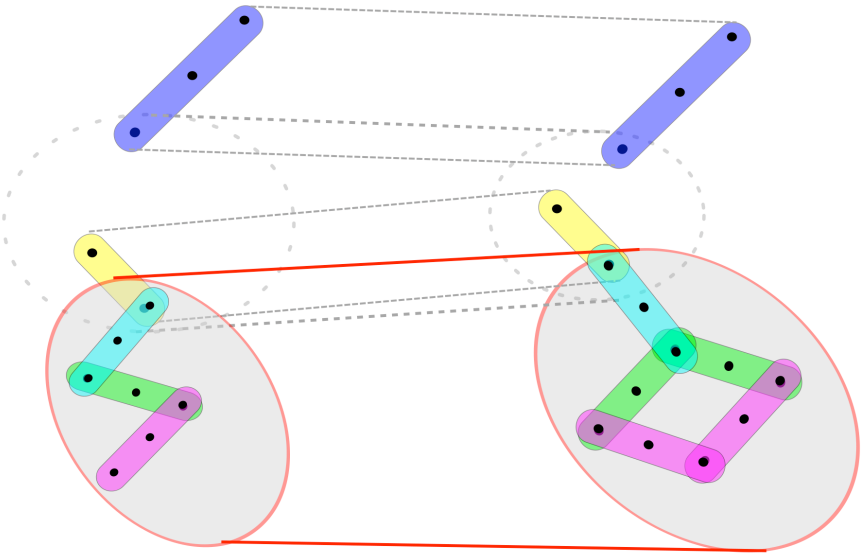
GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $\quad Z : M \approx_{GN}^{(k)} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$



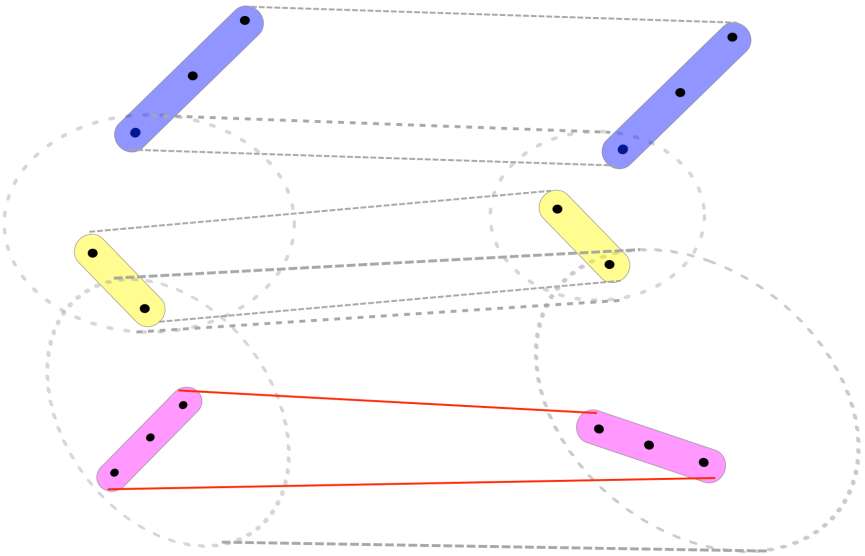
GN-bisimulation

GN-bisimulation *(of width $k \geq 1$)* $\quad Z : M \approx^{(k)}_{GN} N$

non-empty family Z of local isomorphisms $f : M \to N$ s.t. f.a. $f \in Z$

(forth) for all finite $X \subseteq dom(M)$ *(with $|X| \leq k$)*
ex. hom. $h : M|_X \to N$ compatible with f
and $h|_{\bar{c}} \in Z$ for every $\bar{c}$ guarded in M

(back) likewise in the other direction, where $X \subseteq dom(N)$



[*X restricted to guarded sets $\rightsquigarrow$ guarded bisimulation*]

GN-bisimulation invariance

Prop (cf. GN-normal form) every $\varphi \in \mathrm{GNFP}^{(k)}$ is invariant under $\approx_{GN}^{(k)}$

$\Longrightarrow$ Tree-like Model Property (via GN-unravelling...)

Thrm $\mathrm{GNFO}^{(k)}$ is the $\approx_{GN}^{(k)}$-invariant fragment of FO (unrestr. models)

proof uses Compactness Theorem & ω-saturated models ...

Henessy-Milner prop on ω-saturated structures M, N

$$\{\ (\bar{a}, \bar{b}) \in \mathit{guarded}(M) \times \mathit{guarded}(N) \mid M, \bar{a} \equiv_{\mathrm{GNFO}}^{(k)} N, \bar{b}\ \}$$

is a GN-bisimulation (of width k).

(subtle issue with definition of GNFO^k *)*

Recall basic prop's of GNFO & GNFP

smooth model theory

- invariance under GN-bisimulation
- Tree-like Model Property of GNFP
- Finite Model Property of GNFO
- Beth definability *and Craig interpolation*

no added computational cost

- satisfiability is 2ExpTime-complete for both GNFO and GNFP (also for validity and entailment)
- also GNFP-finSAT is 2ExpTime-complete (via [B.,Bojańczyk '11])

model checking (combined complexity)

- is $\mathrm{P}^{\mathrm{NP}[O(log^2 n)]}$-complete for GNFO
- is hard for P^{NP} and in $\mathrm{NP}^{\mathrm{NP}} \cap \mathrm{coNP}^{\mathrm{coNP}}$ for GNFP

Where we left off... what's news!

Lots of questions

- prove GNFO $\equiv$ FO/$\approx_{GN}$ in the finite
  Done! [M. Otto '12]
- characterize GNFP in terms of $\approx_{GN}$-invariance
  GNFP = GSO/$\approx_{GN}$ [Erich's BSc student! '12]
- boundedness problem for GNFP
  we are far from there, but decidable for GN-Datalog !
- Craig interpolation for GNFO (for UNFO$^{(k)}$ cf. [Balder-Luc '11])
  Done! (as of Saturday), fails for GNFO^k
- Beth property for GNFO
  follows from Craig, also holds for GNFO^k

What's new still ?

GN-RA

unrestricted: selection, projection, crossproduct, intersection, union
difference restricted to $\pi_{i_1,\ldots,i_k}(R) \setminus Expr$

GN-SQL

`not`(*condition*) — only when *condition* has ≤ 1 free tuple variable
q1 `except` *q2* — only when *q1* is a simple projection (`select` ... `from` R)

GNFP with simultaneous fixpoints

syntactic sugar only

GN-Datalog

stratified Datalog with only guarded negation

Codd completeness

GN-RA $\equiv$ GN-SQL $\equiv$ non-rec. GN-Datalog $\equiv$ domain-indep. GNFO

GN-RA

GN-RA Codd's RA with the sole restriction

- $E1 \setminus E2$ allowed only when $E1$ is of the form R or $\pi_{\ldots}(R)$

Cf. *semijoin algebra* SA ($\ltimes_\vartheta$ in place of $\times$) Codd complete for GFO [Leinders et al.'05]

non-examples

- $(\pi_1(R) \times S) - \pi_{1,1}(R)$ (distinct pairs from $\pi_1(R) \times S$)
- $\pi_{1,4}(\sigma_{2=3}(R \times R)) - R$ (reachability in two steps, not one)
- $\pi_1(R) - \pi_1((\pi_1(R) \times S) - R)$ (the quotient $R \div S$)

Codd completeness

GN-RA $\rightsquigarrow$ GNFO linear, GNFO $\rightsquigarrow$ GN-RA exponential

GN-SQL

FO-SQL without aggregation, arithmetic, etc.

query := `select` (t_1 as $\mathtt{ATTR}_1, \ldots, t_n$ as $\mathtt{ATTR}_n$)
`from` ($\mathtt{REL}_1$ $R_1, \ldots,$ $\mathtt{REL}_m$ R_m) `where` *condition*
| *query* `union` *query* | *query* `intersect` *query*
| *query* `except` *query*

condition := `true` | $t_1 = t_2$ | t `in` *query* | `exists`(*query*)
| `not`(*condition*) | *condition* `and` *condition*
| *condition* `or` *condition*

GN-SQL negation-guarded FO-SQL, meaning:

- *q1* `except` *q2* – only for FV(*q2*) = $\emptyset$ and *q1* a simple projection: '`select` ... `from` R `where true`'
- `not`(*cond*) – only for $|\mathrm{FV}(cond)| \leq 1$

Codd completeness GN-SQL $\equiv$ domain-independent GNFO

GN-Datalog

Stratified Datalog
sequence $\tilde{\Pi} = (\Pi_1, \ldots, \Pi_n)$ of Datalog$^\neg$ programs (strata),
where $\mathrm{EDB}^{\Pi_i} = \mathrm{EDB}^{\Pi_{i-1}} \cup \mathrm{IDB}^{\Pi_{i-1}}$ $(i = 2 \ldots n)$

GN-Datalog program stratified $\tilde{\Pi} = (\Pi_1, \ldots, \Pi_n)$, where each rule

$$\phi_0 \leftarrow (\neg)\phi_1, \ldots, (\neg)\phi_n \quad \in \mathrm{Rules}^{\Pi_k} \;\; (1 \leq k \leq n)$$

is *negation guarded*, meaning: the head atom ϕ_0 and every negated atom $\neg\phi_i$ has a positive EDB^{Π_k}-atom ϕ_j guarding it

GN-Datalog query GN-Datalog program + UCQ over $\mathrm{EDB}^{\Pi_N} \cup \mathrm{IDB}^{\Pi_N}$

- non-recursive GN-Datalog is Codd complete for GNFO
- GN-Datalog $\rightsquigarrow$ GNFP with simultaneous fixpoints
- GNFP with simultaneous fp. $\rightsquigarrow$ alt-free GNFP

(each of these translations incurs an exponential blow-up)

Query containment

Thrm 2ExpTime-complete for both GN-SQL and GN-Datalog queries

GN-SQL $\overset{poly}{\rightsquigarrow}$ **GNFO** $q_1 \subseteq q_2$ iff $\neg\exists\bar{x}(q_1(\bar{x}) \wedge dummy(\bar{x}) \wedge \neg q_2(\bar{x}))$
(can assume domain-indep. in hardness proofs for GNFO)

GN-Datalog $\overset{exp}{\rightsquigarrow}$ GNFP-simult.fp. $\overset{exp}{\rightsquigarrow}$ GNFP *2x exp. blowup workaround*

- expand signature with IDBs
- push simultaneous fixpoints through the reductions
  GNFP $\rightsquigarrow$ GFP $\rightsquigarrow$ 2-way tree automata, in particular..

GNFP with simultaneous fixpoints (FIN-)SAT is 2ExpTime

Corollaries decidability of containment of

- monadic Datalog queries and UCQs [Cosmadakis et al.'88]
- Datalog queries in UCQs [Chaudhuri-Vardi'97] *(just add guards!)*

+ *finite controllability* of satisfiability and query containment for GN-SQL and non-rec. GN-Datalog

Closed-world query answering (combined complexity)

GN-SQL $\mathrm{P}^{\mathrm{NP}[\log^2]}$-complete

via GN-SQL $\overset{poly}{\rightsquigarrow}$ GNFO and GNFO $\overset{poly}{\rightsquigarrow}$ GN-SQL in the presence of ADOM

GN-Datalog P^{NP}-complete

(already for non-recursive GN-Datalog on a fixed instance with unary IDBs and only zero-ary negation (!))

- reduction from LEX(SAT) = $\left\{ \begin{array}{l} \text{most significant bit of} \\ \text{lex. min. sat. assignement} \end{array} \right.$
- cf. complexity gap for GNFP: P^{NP}-hard and in $\mathrm{NP}^{\mathrm{NP}} \cap \mathrm{coNP}^{\mathrm{NP}}$

Model Checking (combined complexity)

P-complete for ML

[Berwanger-Grädel '01]

- P-complete for GFO
- in NP $\cap$ coNP for GFP

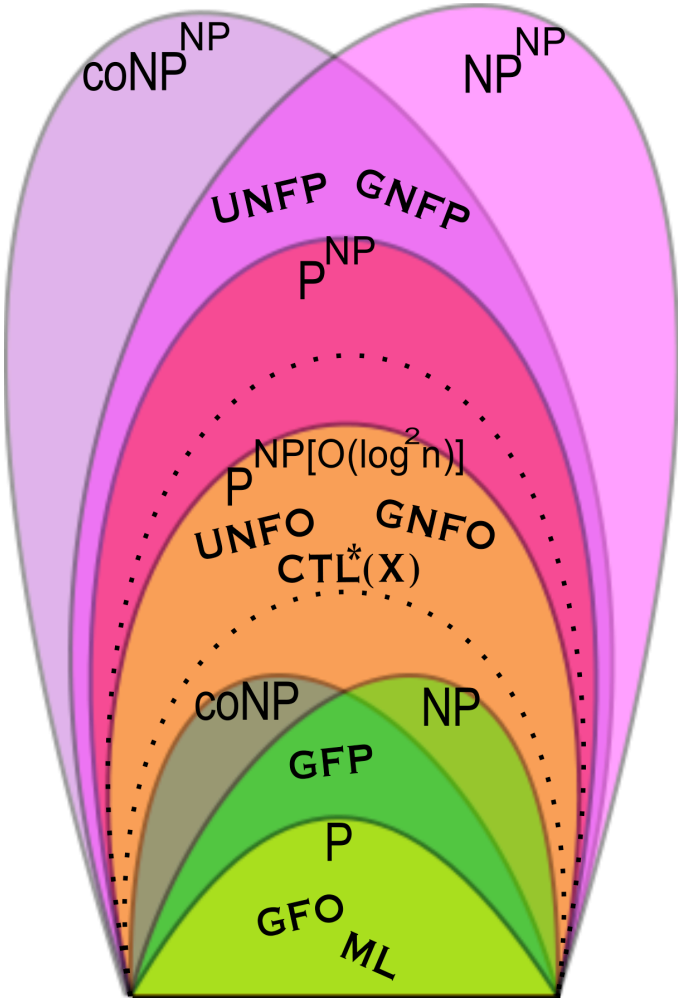
[Schnoebelen '03]

$\mathsf{P}^{\mathsf{NP}[O(\log^2 n)]}$-complete for $\mathsf{CTL}^*(X)$

[ten Cate-Segoufin '11]

- $\mathsf{P}^{\mathsf{NP}[O(\log^2 n)]}$-complete for UNFO
- in $\mathsf{NP}^{\mathsf{NP}} \cap \mathsf{coNP}^{\mathsf{NP}}$ for UNFP

same for GNFO and GNFP, resp.
via reduction to the above



Open-world query answering (data complexity)

OWA semantics (incomp. DBs, data exchange, ontological reasoning...)

$$I \models_{OWA} q(\bar{a}) \qquad \textbf{iff} \qquad I \cup \{\neg q(\bar{a})\} \text{ unsat.}$$

$OWA_{q,\Sigma}$ asks whether $I, \Sigma \models_{OWA} q(\bar{a})$
(constraints Σ, query q, input instance I and $\bar{a}$ in $\mathrm{adom}(I)$)

TGD $\forall \bar{x}\bar{y}\ \Phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\ \Psi(\bar{y}, \bar{z})$ with Φ, Ψ conj. of atoms
frontier guarded if Φ contains an atom guarding $\bar{y}$ [Baget et al.'11]

Thrm $OWA_{q,\Sigma}$ is coNP-complete for $q \in$ GNFO and fgTGDs Σ

- fgTGDs can be compiled into the GNFO query q
- **lemma** whenever $I \subseteq J \models q(\bar{a})$ then there is some $I \subseteq J' \models q(\bar{a})$ such that $|J'| = O(|I|)$.

Open-world query answering cont'd (data complexity)

Serial GNFO queries (SGNQ) are in DNF
with no positive occurrence of a subformula $\neg\chi \wedge \ldots \wedge \neg\psi$

- frontier-guarded TGDs $\Sigma \rightsquigarrow \bigvee_{\sigma \in \Sigma} \neg\sigma$ serial GNFO query

Thrm for fgTGDs Σ and q a serial GNFO query

- $OWA_{q,\Sigma}$ is PTime-complete
- ex. *Datalog rewriting* (Π, Ans) s.t. $I \models_{OWA} q$ iff $\Pi(I) \models Ans$

reduction to [Baget et al.'11]
CQs are Datalog rewritable over fgTGDs $\rightsquigarrow$ PTime data complexity

Thrm there is a boolean SGNQ q and a single *key constraint* κ
such that $OWA_{q,\{\kappa\}}$ is *undecidable*

Boundedness of Datalog programs

Boundedness Datalog program Π is *bounded* (*in the finite*) if
l.f.p. of Π is reached in k steps on any (*finite*) instance: $\Pi^\infty = \Pi^k$

[Barwise-Moschovakis '78] (classically)
IDB-positive first-order program Π is bounded iff Π^∞ is FO-definable

undecidability is the rule (even for very rudimentary programs)

hitherto champion [Cosmadakis,Gaifman,Kanellakis,Vardi '88]
for monadic Datalog boundedness is decidable and coinsides with boundedness in the finite

employed two-way alternating automata on trees for the purpose,
$\rightsquigarrow$ *main vehicle of decision proc. for* L_μ^-, GFP, UNFP, *and* GNFP

Boundedness of GN-Datalog programs

GNFO-program IDB-positive program Π with rules

$$X_i(\bar{x}) \longleftarrow \alpha_i(\bar{x}) \wedge \phi_i(\bar{X}, \bar{x})$$

where α_i is an EDB predicate, and ϕ_i is positive in the IDB preds $\bar{X}$

Claim for Π a GNFO-program t.f.a.e.

1. Π^∞ is FO-definable over all (finite) instances
2. Π^∞ is GNFO-definable over all (finite) instances
3. Π is bounded over all (finite) instances

Godfather theorem [Blumensath,Otto,Weyer'11, Colcombet-Löding?]
boundedness is decidable for GSO* over structures of tree-width k

$\Longrightarrow$ boundedness for GNFO-programs is decidable

$\Longrightarrow$ boundedness for GN-Datalog is decidable
using the result for GNFO-programs and the Claim stratum-by-stratum

Interpolation and definability

Craig interpolation Given $\varphi \models \psi$ in respective signatures τ and σ there is some χ in signature $\tau \cap \sigma$ such that $\varphi \models \chi \models \psi$

Def a $\tau \cup \nu \cup \{Q\}$-formula φ *implicitly defines* Q in terms of ν
if for every $(A, \bar{T}, \bar{V}, Q) \models \varphi$ and $(A, \bar{S}, \bar{U}, P) \models \varphi$ such that $\bar{V} = \bar{U}$ it holds that $Q = P$.

projective Beth property If $\tau \cup \nu \cup \{Q\}$-formula φ *implicitly defines* Q then there is some ν-formula ψ such that $\varphi \models \forall \bar{x}(Q\bar{x} \leftrightarrow \psi(\bar{x}))$

Beth property as above only for $\tau = \emptyset$

Fact Craig $\Longrightarrow$ projective Beth

consider the entailment $\varphi \wedge Q(\bar{x}) \models \varphi' \rightarrow Q'(\bar{x})$
where in φ' *all pred. names* $R \in \tau \cup \{Q\}$ *are subst'd with* R'

Interpolation and definability cont'd

GFO [Hoogland,Marx,Otto '99]

- no Craig interpolation !
- Beth definability intact

UNFO [Balder & Luc '11]

- UNFO has Craig interpolation
- UNFO^k has Craig for all k

GNFO *new*

- GNFO has Craig interpolation
- GNFO^k does *not* have Craig interpolation
  for any k ex. GFO^3-formulas with no GNFO^k interpolant
- GNFO^k has Beth definability (projective Beth open)

Further research

things to do this summer

- Balder: extend GN-Datalog to capture alt.-free GNFP (cf. DatalogLITE) !
- Balder (not really): complexity of boundedness for GN-Datalog ?
- Luc: is GNFP model checking P^{NP} ?
- Martin: char. of GNFP using $\approx_{GN}$-inv. *in the finite* (long open for L_μ) !
- Martin: Lindström char. of GNFO (cf. van Benthem on ML) ?
- Michael: any bound on complexity of interpolants ?
- V.: show that GNFO is the least ... extension of GFO with interpolation !
- Lucifer: boundedness problem for GNFP?

Further research

things to do this summer

- Balder: extend GN-Datalog to capture alt.-free GNFP (cf. DatalogLITE) !
- Balder (not really): complexity of boundedness for GN-Datalog ?
- Luc: is GNFP model checking P^{NP} ?
- Martin: char. of GNFP using $\approx_{GN}$-inv. *in the finite* (long open for L_μ) !
- Martin: Lindström char. of GNFO (cf. van Benthem on ML) ?
- Michael: any bound on complexity of interpolants ?
- V.: show that GNFO is the least ... extension of GFO with interpolation !
- Lucifer: boundedness problem for GNFP?

Thank You!