Finite Model Reasoning in Expressive Fragments of First-Order Logic

Lidia Tendera

Institute of Mathematics and Informatics
Opole University, Poland

M4M Kanpur
8.–9. January 2017

Outline

Introduction/Motivation
- Standard translation of ML

Base Fragments
- Definitions
- Properties and Complexity

Extensions of Base Fragments
- More operators
- Special classes of structures

Deciding (Fin)Sat
- More or less natural reductions
- Finitary unravellings
- Linear/Integer Programming

Conclusion

Classical Decision Problem

$\mathcal{L}$ – any logic $\mathcal{FO}$– first-order logic

- ***Sat($\mathcal{L}$):***
  given a formula $\varphi \in \mathcal{L}$, does φ admit a model ?
- ***FinSat($\mathcal{L}$):***
  given a formula $\varphi \in \mathcal{L}$, does φ admit a **finite** model ?

Theorem (Church, Turing, Trahtenbrot)

Sat($\mathcal{FO}$) and FinSat($\mathcal{FO}$) are undecidable and recursively inseparable.

Possible response:

$\rightsquigarrow$ devise incomplete algorithms

$\rightsquigarrow$ identify decidable **fragments**

Why FinSat?

Databases, systems etc. are often considered to be **finite**.

- $\mathcal{L}$ has the **finite model property (FMP)** iff every satisfiable $\varphi \in \mathcal{L}$ has a finite model.

Observation

- If $\mathcal{L}$ has FMP then $Sat(\mathcal{L})$ and $FinSat(\mathcal{L})$ coincide.
- If $\mathcal{L}$ is a fragment of $\mathcal{FO}$ and $\mathcal{L}$ has FMP then $Sat(\mathcal{L})$ is decidable.

Decidable Fragments of FO

Note: one cannot study ***all*** possible fragments!

- defined by restrictions on signatures
  e.g. [Löwenheim-Skolem 1915] monadic theories (FMP)
- prenex classes defined by quantifier prefix
  $\exists^*\forall^*$, $\exists^*\forall\exists^*$, $\exists^*\forall\forall\exists^*$ (equality free)
- defined by other syntactic restrictions and suitably motivated

Also: we want to identify reasons for a logic to be (un)decidable, (in)tractable etc.

- Can we decide whether a formula is satisfiable without actually seeing a model?
- Some formulas have only infinite models.
  Can we decide whether they are satisfiable?

Motivation: Modal Logic

[Vardi 1996]: Why is modal logic so robustly decidable?

- Propositional modal logic:
  Boolean logic + operators $\Diamond$ (possibly) and $\Box$ (necessary)
- Good model-theoretical and algorithmic properties,
  robustly decidable
- Variants and extensions of modal logics have applications in various areas of computer science:
  - verification of hardware and software
  - artificial intelligence
  - distributed systems
  - knowledge representation

Standard Translation of Modal Logic (1)

- Modal logic can be translated into $\mathcal{FO}$:

$$P \wedge \Diamond(Q \vee \Box \neg P) \quad \leftrightsquigarrow \quad Px \wedge \exists y(Rxy \wedge (Qy \vee \forall z(Ryz \rightarrow \neg Pz)))$$

- $\mathcal{FO}^3$ undecidable [Kahr, Moore, Wang, 1959]

[Gabbay, 1981] TWO variables suffice!

Observation

- ML can be embedded in the **two-variable fragment** $\mathcal{FO}^2$

Standard Translation of Modal Logic (1)

- Modal logic can be translated into $\mathcal{FO}$:

$$P \wedge \Diamond(Q \vee \Box \neg P) \quad \rightsquigarrow \quad Px \wedge \exists y(Rxy \wedge (Qy \vee \forall x(Ryx \rightarrow \neg Px)))$$

- $\mathcal{FO}^3$ undecidable [Kahr, Moore, Wang, 1959]

[Gabbay, 1981] TWO variables suffice!

Observation

- ML can be embedded in the **two-variable fragment** $\mathcal{FO}^2$

Standard Translation of Modal Logic (2)

$$P \wedge \Diamond(Q \vee \Box\neg P) \quad \rightsquigarrow \quad Px \wedge \exists y(Rxy \wedge (Qy \vee \forall z(Ryz \rightarrow \neg Pz)))$$

The translation suggests other restrictions of $\mathcal{FO}$:

- **fluted fragment** $\mathcal{FL}$:
  variables appear in some fixed order and no quantifier-rescoping occurs; order of quantification of variables matches order of appearance in predicates.
- **guarded fragment** $\mathcal{GF}$:
  quantifiers are relativized by atomic formulas
- **unary negation fragment** $\mathcal{UNF}$:
  negation is applied only to subformulas with a single free variable.

Problems Reducing to (Fin)Sat

Example: Query Answering

A knowledge base $\langle \mathbf{D}, \mathcal{O} \rangle$:
database $\mathbf{D}$ (a set of facts, i.e. ground atoms),
ontology $\mathcal{O}$ (i.e. a logical formula).

- **Query Answering:**
  given a knowledge base $\langle \mathbf{D}, \mathcal{O} \rangle$ and a query Q:
  does $\langle \mathbf{D}, \mathcal{O} \rangle$ entail Q, i.e.

$$\mathbf{D} \wedge \mathcal{O} \models Q?$$

Observation

$\mathbf{D} \wedge \mathcal{O} \models Q$ iff $\mathbf{D} \wedge \mathcal{O} \wedge \neg Q$ is unsatisfiable

Base Fragments

Fragments embedding modal logic

- two-variable fragment $\mathcal{FO}^2$
- fluted fragment $\mathcal{FL}$
- guarded fragment $\mathcal{GF}$
- unary negation fragment $\mathcal{UNF}$

Theorem

All four base fragments enjoy the finite model property.

- FMP often gives a bound on the size of minimal models.

Hence:

- FMP often gives an upper bound for the computational complexity of $Sat(\mathcal{L})$=$FinSat(\mathcal{L})$.

FMP and Complexity of $\mathcal{FO}^2$

Theorem (Mortimer, 75)

$\mathcal{FO}^2$ has doubly exponential *model property: every satisfiable $\varphi \in \mathcal{FO}^2$ has a model of size at most doubly exponential in $|\varphi|$.*

Theorem (Grädel, Kolaitis, Vardi, 97)

$\mathcal{FO}^2$ has exponential *model property: every satisfiable $\varphi \in \mathcal{FO}^2$ has a model of size at most exponential in $|\varphi|$.*

Corollary

Sat($\mathcal{FO}^2$) is NExpTime*-complete.*

Fluted Fragment $\mathcal{FL}$

- First identified by W.V.Quine in 1968:
  - homogeneous m-adic formulas (generalization of monadic fragment)
  - later generalized to fluted fragment
- Examples of fluted formulas:

  No student admires every professor

  $\forall x_1(\text{student}(x_1) \rightarrow \neg\forall x_2(\text{prof}(x_2) \rightarrow \text{admires}(x_1, x_2)))$

  No lecturer introduces any professor to every student

  $$\forall x_1(\text{lecturer}(x_1) \rightarrow \neg\exists x_2(\text{prof}(x_2)\wedge \forall x_3(\text{student}(x_3) \rightarrow \text{intro}(x_1, x_2, x_3)))).$$

- Order of quantification of variables matches order of appearance in predicates.

Fluted Fragment $\mathcal{FL}$

- First identified by W.V.Quine in 1968:
  - homogeneous m-adic formulas (generalization of monadic fragment)
  - later generalized to fluted fragment
- Examples of fluted formulas:

No student admires every professor

$$\forall x_1(\text{student}(x_1) \to \neg\forall x_2(\text{prof}(x_2) \to \text{admires}(x_1, x_2)))$$

No lecturer introduces any professor to every student

$$\forall x_1(\text{lecturer}(x_1) \to \neg\exists x_2(\text{prof}(x_2) \wedge \forall x_3(\text{student}(\quad) \to \text{intro}(\quad,\quad,\quad)))).$$

- Order of quantification of variables matches order of appearance in predicates.

Fluted Fragment $\mathcal{FL}$

- First identified by W.V.Quine in 1968:
  - homogeneous m-adic formulas (generalization of monadic fragment)
  - later generalized to fluted fragment
- Examples of fluted formulas:

  No student admires every professor

  $\forall x_1(\text{student}(x_1) \rightarrow \neg\forall x_2(\text{prof}(x_2) \rightarrow \text{admires}(x_1, x_2)))$

  No lecturer introduces any professor to every student

  $\forall x_1(\text{lecturer}(\quad) \rightarrow$
  $\neg\exists x_2(\text{prof}(\quad)\wedge$
  $\forall x_3(\text{student}(\quad) \rightarrow \text{intro}(\quad,\quad,\quad)))).$

- Order of quantification of variables matches order of appearance in predicates.

FLUTED FRAGMENT $\mathcal{FL}$

- First identified by W.V.Quine in 1968:
  - homogeneous *m*-adic formulas (generalization of monadic fragment)
  - later generalized to fluted fragment
- Examples of fluted formulas:

No student admires every professor

$$\forall x_1(\mathrm{student}(x_1) \rightarrow \neg\forall x_2(\mathrm{prof}(x_2) \rightarrow \mathrm{admires}(x_1, x_2)))$$

No lecturer introduces any professor to every student

$$\forall \quad (\mathrm{lecturer}(\quad) \rightarrow$$
$$\neg\exists \quad (\mathrm{prof}(\quad)\wedge$$
$$\forall \quad (\mathrm{student}(\quad) \rightarrow \mathrm{intro}(\quad,\quad,\quad)))).$$

- Order of quantification of variables matches order of appearance in predicates.

$\mathcal{FL}$ FORMAL DEFINITION

- Let $x_1, x_2, \ldots$ be a fixed sequence of variables.
- The fluted fragment with k free variables, $\mathcal{FL}^{[k]}$, is defined by simultaneous induction for all k:
  - any atom $p(x_\ell, \ldots, x_k)$ is in $\mathcal{FL}^{[k]}$;
  - $\mathcal{FL}^{[k]}$ is closed under Boolean operations;
  - $\mathcal{FL}^{[k]}$ contains $\exists x_{k+1}\varphi$ and $\forall x_{k+1}\varphi$ for any $\varphi \in \mathcal{FL}^{[k+1]}$.
- The fluted fragment, $\mathcal{FL}^{[k]}$ is the union:

$$\mathcal{FL} = \bigcup_{k \geq 0} \mathcal{FL}^{[k]}.$$

- For all $m > 0$, we define $\mathcal{FL}^m$, to be the set of fluted formulas containing at most the variables $x_1, \ldots, x_m$, free or bound.

FMP and Complexity of $\mathcal{FL}$

Purdy 1996, Pratt-Hartmann, Szwast, T. 2016

- Any satisfiable formula of $\mathcal{FL}^m$ has a model of m-tuply exponential size, that is, of size bounded by a function

$$2^{\cdot^{\cdot^{\cdot^{2^{p(\|\varphi\|)}}}}}\Big\}{m \text{ 2's}} = \mathfrak{t}(m, p(\|\varphi\|))$$

  where p is a polynomial.
  Hence, $Sat(\mathcal{FL}^m)$ is in m-NExpTime.
- On the other hand, satisfiable formulas of $\mathcal{FL}^{2m}$ force models of m-tuply exponential size.
- Essentially the same proof shows that $Sat(\mathcal{FL}^{2m})$ is m-NExpTime-hard.
- Therefore, for $m \geq 1$, the complexity of $Sat(\mathcal{FL}^m)$ lies between

  $\lfloor m/2 \rfloor$-NExpTime-hard and m-NExpTime.

Guarded Fragment $\mathcal{GF}$

Andréka, van Benthem and Németi, 1996

- Restricting the use of quantifiers:

$$\forall \mathbf{x}(G(\mathbf{x}, \mathbf{y}) \rightarrow \varphi(\mathbf{x}, \mathbf{y}))$$

$$\exists \mathbf{x}(G(\mathbf{x}, \mathbf{y}) \wedge \varphi(\mathbf{x}, \mathbf{y}))$$

$G(\mathbf{x}, \mathbf{y})$ – atomic formula, ***guard*** of the quantifier.

FMP and Complexity of $\mathcal{GF}$

Theorem (Grädel, 99)

GF has doubly exponential model property[1]*. Moreover,*

- *Sat*($\mathcal{GF}$) *is* 2-EXPTIME*-complete*
- *Sat*($\mathcal{GF}^k$) *is* EXPTIME*-complete, for any fixed k.*

The above complexity bounds do not follow from the bound on size of finite models.

- $\mathcal{GF}$ has the **tree model property**.

[1] Nice application of combinatorial results by Hrushowski, Herwig and Lascar about extensions of certain graphs.

Unary Negation Fragment $\mathcal{UNF}$

ten Cate, Segoufin 2011

Idea: start from existential $\mathcal{FO}$ (unions of conjunctive queries) allow only $\neg\varphi(x)$, where x is a **single** variable.
Formally:

- any atom of the form $R(\bar{x})$ or $x = y$ is in $\mathcal{UNF}$;
- $\mathcal{UNF}$ is closed under $\vee$, $\wedge$ and $\exists$;
- if $\varphi(x)$ is a formula of $\mathcal{UNF}$ with no free variables besides (possibly) x, then $\neg\varphi(x)$ belongs to $\mathcal{UNF}$.

Examples.

- $\forall x \forall y \forall z (Pxyz \rightarrow Rxyz) \in \mathcal{GF}$ (but not in $\mathcal{UNF}$)
- $\forall x \exists y \exists z (Rxy \wedge Ryz \wedge Rzx) \in \mathcal{UNF}$ (but not in $\mathcal{GF}$ or $\mathcal{FO}^2$)
- $\exists x \exists y \neg Rxy \in \mathcal{FL}^2$ (but not in $\mathcal{UNF}$ or $\mathcal{GF}$)

Extensions

Trace the limits of decidability: study properties of extensions of base fragments obtained by adding e.g.

- counting operators / functions
- built-in relations (e.g. orderings, equivalences)

and going beyond FO:

- transitive (or equivalence) closure
- fixed-points
- …

More operators and Loss of FMP

Infinity Axioms

$$\exists x \forall y \neg Ryx \land \forall x \exists y Rxy \land \forall x \exists^{\leq 1} y Ryx \tag{1}$$

$$\circ \longrightarrow \circ \longrightarrow \circ \longrightarrow \circ \longrightarrow \circ \longrightarrow \circ \quad \cdots$$

$$\forall x \neg Rxx \land \forall x \exists y Rxy \land \forall x \forall y (\mathrm{TC}(R)xy \leftrightarrow Rxy) \tag{2}$$

Overview: Extensions of Base Fragments

motivated by extensions of modal/temporal/description logics

Logic	Transitive closure	mon-Fixed Points	Counting
$\mathcal{GF}$	undecidable G 99	2-ExpTime *Sat:* GW 99 *FinSat:* BB 12	undecidable G 99
$\mathcal{GF}^2$	decidable* M 09	ExpTime *Sat:* GW 99 *FinSat:* BB 12	ExpTime P-H 05
$\mathcal{FO}^2$	undecidable GOR 97	undecidable GOR 97	NExpTime *Sat:* PST 97, P-H 05 *FinSat:* P-H 05
$\mathcal{UNF}$	?	2-ExpTime StC 13	?
$\mathcal{FL}$	?	?	?

Bárány, Bojańczyk, Grädel, Michaliszyn, Otto, Pratt-Hartmann, Rosen, Pacholski, Segoufin, Szwast, ten Cate, T., Walukiewicz

Restricted Classes of Structures

- None of the base fragments can express transitivity e.g. transitivity allows one to write **infinity axioms**:

$$\forall x \neg(x < x) \wedge \forall x \exists y \, (x < y) \quad \text{with transitive} <$$

- Transitivity is a useful property
- Solution: consider satisfiability in classes of structures with predefined interpretation of some binary symbols (as transitive relations, orders, equivalences, etc.)
- Corresponds to (multi-)modal logics K4, S4, S5

Theorem

$\mathcal{FO}^2$ and $\mathcal{GF}^2$ undecidable with several transitive, order or equivalence relations (Grädel, Otto, Rosen, Ganzinger, Meyer, Veanes, Kazakov, Kieroński, T....)

$\mathcal{FO}^2$ AND $\mathcal{GF}^2$ OVER SPECIAL CLASSES OF STRUCTURES

Logic	Special symbols	Number of special symbols in the signature: 1	2	3 or more
$\mathcal{GF}^2$ FMP EXPTIME	Transitivity	2-EXPTIME K05	undecidable K05, Kaz06	undecidable GMV99
	Equivalence	FMP, NEXPTIME KO05	*Sat:* KO05 2-EXPTIME *FinSat:* KP-HT15	undecidable KO05
$\mathcal{FO}^2$ GKV97 FMP NEXPTIME	Transitivity	*Sat:* ST13 in 2-NEXPTIME$^{*)}$ *FinSat:* ?	undecidable K05, Kaz06	undecidable GOR99
	Linear order	NEXPTIME Ott01	*Sat:* ? EXPSPACE$^{*)}$ *FinSat:* SchZ10	undecidable Ott01, K11
	Equivalence	FMP, NEXPTIME KO05	2-NEXPTIME KMP-HT12	undecidable KO05
	Equivalence Closure	FMP, NEXPTIME KMP-HT12	2-NEXPTIME KMP-HT12	undecidable KO05

. . . , Ganzinger, Meyer, Veanes, Kazakov, Schwentick, Zeume

Deciding Finite Satisfiability

1. Finite Model Property and Tree Model Property.
2. More or less Natural Reductions.
3. Locally Acyclic Covers.
4. via Linear/Integer Programming.

FMP and TMP

- FMP often gives natural upper complexity bounds.
- FMP does not help when $\mathcal{L}$ contains infinity axioms.

$\mathcal{L}$ has **tree (tree-like) model property** iff every satisfiable $\varphi \in \mathcal{L}$ has a tree (tree-like) model.

- Advantages:
  - Useful for logics without FMP.
  - Often gives better complexity bounds.
- Positive examples:
  $\mathcal{FO}^2, \mathcal{GF}, \mathcal{UNF}$.
- Disadvantage:
  Tree-like models are usually infinite, so TMP is not suitable to decide *FinSat*.

Reductions to other fragments

Theorem

[Segoufin, ten Cate 2015] There is an exponential reduction from UNF with fixed-points to μ-calculus preserving finiteness of models. Hence:

- *$\mathcal{UNF}$ with fixed-points has TMP and is 2-EXPTIME-complete.*
- *$\mathcal{UNF}$ has FMP.*

REDUCING $FinSat(\mathcal{L})$ TO $Sat(\mathcal{L})$

- $\mathcal{DL}$-Lite [Rosati 2008]
- Horn-$\mathcal{SHIQ}$ [Garcia, Lutz, Schneider 2013]

Idea. Complete a given TBox $\mathcal{T}$ to $\mathcal{T}_{fin}$ by adding new axioms (reversing cycles in $\mathcal{T}$) and show that

$$\mathcal{T} \text{ is finitely satisfiable iff } \mathcal{T}_{fin} \text{ is satisfiable.}$$

- Advantage:
  allows to run existing reasoners for *Sat*.

Finitary Unravellings

Roughly: TMP in the finite

- [Otto 2004]: "Finitary unravelling" – construction that make a structure locally acyclic (avoiding short cycles).
- [Bárány, Gottlob, Otto 2009]:
  Every finite structure is $\mathcal{GF}$-bisimilar to a finite structure whose hypergraph is locally acyclic (suitably defined).

Applied to obtain:

- small model property for $\mathcal{GF}$.
- decidability of *FinSat* for $\mathcal{GF}$ with fixed points.
- correctness of the reduction from $\mathcal{UNF}$ with fixed points to μ-calculus.

Reduction to Linear/Integer Programming

Idea: Depending on the logic:
identify (finitely many types of) building blocks of a potential model and connecting conditions for them, describe them in a succinct way by a set of (in)equalities.

Advantages:

- Useful for solving *simultaneously Sat* and *FinSat*.
  We look for solutions over $\mathbb{N}$ (*FinSat*) or over $\mathbb{N} \cup \{\infty\}$ (*Sat*), e.g.

  $$x + 1 = x$$

  has a solution $x = \infty$.
- Does not depend on TMP.
- Gives better (optimal) complexity bounds.

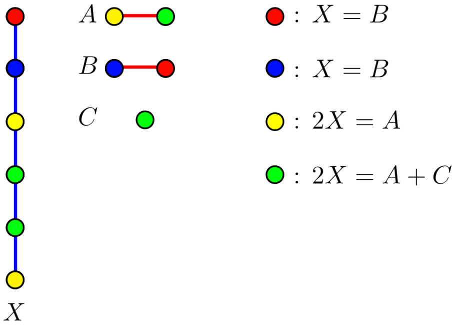
LP/IP: from Models to Bipartite Graphs

Example: $\mathcal{FO}^2$ with two equivalence relations E_1, E_2

- Lemma (small intersections property): in a model of φ we can replace every equivalence class of $E_1 \cap E_2$ by a class bounded exponentially in $|\varphi|$
- Think about E_1-classes and E_2-classes as about nodes of a bipartite graph
- Intersections are represented by edges of the graph
- Colors of edges represent isomorphism types of intersections
- A question, whether φ is satisfiable becomes a question about the existence of a graph satisfying some constraints.
- These constraints can be expressed in terms of linear inequalities.

$\mathcal{FO}^2+\{E_1,E_2\}$: MODEL CONSTRUCTION (IDEA)

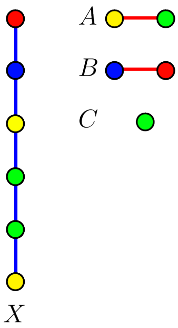
Admissible types: System of equations:



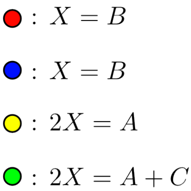
Solution: $X = 1, A = 2, B = 1, C = 0$

$\mathcal{FO}^2+\{E_1, E_2\}$: MODEL CONSTRUCTION (IDEA)

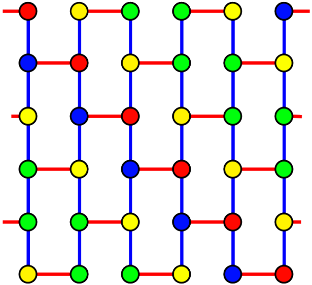
Admissible types:



System of equations:



Model construction:



Solution: $X = 1, A = 2, B = 1, C = 0$

$\mathcal{FO}^2+\{E_1, E_2\}$: Complexity

Construct a system of linear inequalities

- variables correspond to types of classes: triply exponentially many
- inequalities correspond to intersections: doubly exponentially many

As integer programming is in NP this gives 3-NExpTime-upper bound.

- A Caratheodory-type result of Eisenbrand and Shmonin (2006) says that any system of linear inequalities, solvable over integers, has a solution in which the number of non-zero variables is polynomial in the number of inequalities.
- This allows to show 2-NExpTime-upper bound:
  - Just guess relevant variables and construct a system of doubly exponential size.

$\mathcal{FO}^2$+$\{E_1, E_2\}$: Complexity

Construct a system of linear inequalities

- variables correspond to types of classes: triply exponentially many
- inequalities correspond to intersections: doubly exponentially many

As integer programming is in NP this gives 3-NExpTime-upper bound.

- A Caratheodory-type result of Eisenbrand and Shmonin (2006) says that any system of linear inequalities, solvable over integers, has a solution in which the number of non-zero variables is polynomial in the number of inequalities.
- This allows to show 2-NExpTime-upper bound:
  - Just guess relevant variables and construct a system of doubly exponential size.

LP/IP Approach

Successfully used to get optimal complexity bounds for:

- $\mathcal{FO}^2$ and $\mathcal{GF}^2$ with counting quantifiers [Pratt-Hartmann 2005]
- $\mathcal{FO}^2$ with two equivalence relations [Kieroński, Michaliszyn, Pratt-Hartmann, T. 2013]
- $\mathcal{FO}^2$ with counting quantifiers and one equivalence relation [Pratt-Hartmann 2013]
- …

Disadvantages:

- not ideal for implementation.
- not yet clear how to extend to logics with predicates of higher arity.

LP/IP Approach

Successfully used to get optimal complexity bounds for:

- $\mathcal{FO}^2$ and $\mathcal{GF}^2$ with counting quantifiers [Pratt-Hartmann 2005]
- $\mathcal{FO}^2$ with two equivalence relations [Kieroński, Michaliszyn, Pratt-Hartmann, T. 2013]
- $\mathcal{FO}^2$ with counting quantifiers and one equivalence relation [Pratt-Hartmann 2013]
- …

Disadvantages:

- not ideal for implementation.
- not yet clear how to extend to logics with predicates of higher arity.

Future Research (1)

- answer remaining open question
- optimise known algorithms towards practical implementation
- combine fragments
  - [Bárány, ten Cate, Segoufin 2011]
    Guarded Negation Fragment
    a common generalisation of $\mathcal{GF}$ and $\mathcal{UNF}$.
  - [Kuusisto, Hella 2014]
    Uniform One-Dimensional $\mathcal{FO}$
    generalization of $\mathcal{FO}^2$ to contexts with relations of arbitrary arity
- identify useful (and tractable) subfragments
- ...

Future Research (2)

Query Answering:

- $\mathbf{D} \wedge \mathcal{O} \models \mathit{Query}$ iff $\mathbf{D} \wedge \mathcal{O} \wedge \neg \mathit{Query}$ is unsatisfiable

Classes of queries:

- positive existential
- conjunctive queries
- unions of conjunctive queries

Depending on $\mathcal{L}_{\mathcal{O}}$ and considered class of queries, $\neg \mathit{Query}$ is not necessarily in $\mathcal{L}_{\mathcal{O}}$, so existing procedures deciding satisfiability cannot be applied directly.

QA: Complexity Measures

$$\mathbf{D} \wedge \mathcal{O} \models \mathit{Query}$$

- **Data complexity:** only the size of the database matters. The ontology $\mathcal{O}$ and *Query* are considered fixed.
- **Schema complexity:** only the size of the ontology matters. **D** and *Query* are considered fixed.
- **Combined complexity:** no parameter is considered fixed.

In practise one often assumes that the size of the data largely dominates the size of the ontology (and of the query) and considers **data complexity** as the relevant complexity measure.

$\leadsto$ Identify fragments with low data complexity.

Thank You!

Dziękuję!

Questions?