Collections, Cardinalities, and Relations

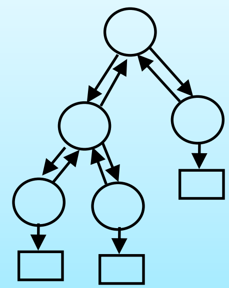
Kuat Yessenov

Ruzica Piskac

Viktor Kuncak

Verification Conditions

```
   . . .
void remove(x : Node) {
   Node prev = null;
   Node current = root;
   while (current != null) {
     if (current = x) ...
     else if (current.data < ...
     else if (current.data > ...
       ,,,,
```



Invariant is preserved

Annotated Java code

Property

Verification system
(Jahob [Zee et al'08])

Verification conditions

$$x \neq \text{null} \wedge lt[x] \in r1 \wedge rt[x] \in r2$$
$$r1 \cap r2 = \emptyset \wedge \text{null} \not\subseteq \text{data}[r2]$$
$$lt[r1] \subseteq r1 \wedge rt[r1] \subseteq r1$$
$$lt[r2] \subseteq r2 \wedge rt[r2] \subseteq r2$$

Example

Abstract model of a Map:

$\text{data} : \mathcal{U} \rightarrow \mathcal{U}$

$\text{keys} \subseteq \mathcal{U}, \text{values} \subseteq \mathcal{U}$

Operation — adding a new pair to the map:

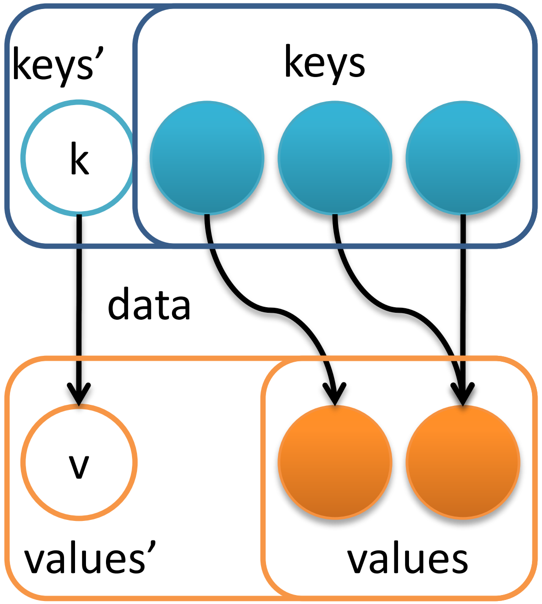
$\text{values} = \text{data}[\text{keys}] \wedge \text{values}' = \text{data}[\text{keys}']$

$\text{keys}' = \text{keys} \cup k \wedge k \not\subseteq \text{keys}$

$|k| = 1 \wedge |\text{data}[k]| = 1$

Property — at most one more new value:

$|\text{values}'| \leq |\text{values}| + 1$



Example (2)

Binary tree with items:

$\text{data}, lt, rt : \mathcal{U} \to \mathcal{U}, |x| = 1, |\text{null}| = 1$

Operation — update left child:

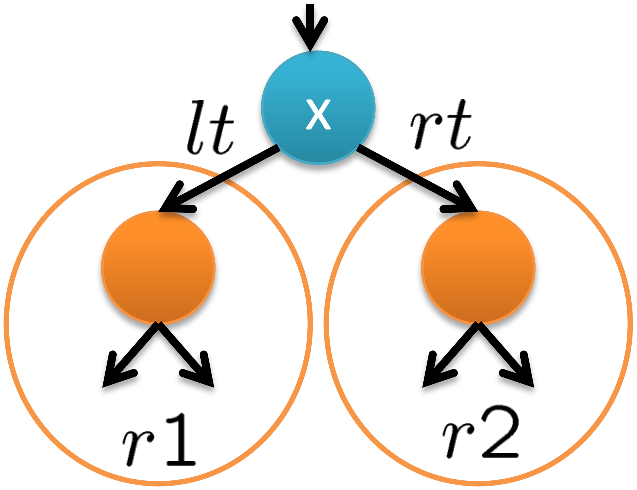
$\text{data}[lt[x]] = \text{null}$

Property — right subtree unaffected:

$\text{null} \not\subseteq \text{data}[r2]$

Regional logic [Banerjee et al'08]

$x \neq \text{null} \wedge lt[x] \in r1 \wedge rt[x] \in r2$
$r1 \cap r2 = \emptyset \wedge \text{null} \not\subseteq \text{data}[r2]$
$lt[r1] \subseteq r1 \wedge rt[r1] \subseteq r1$
$lt[r2] \subseteq r2 \wedge rt[r2] \subseteq r2$



Logic of Sets, Cardinalities, and Relations

- **Boolean algebra of sets:**

$$B ::= x \mid \emptyset \mid \mathcal{U} \mid B_1 \cup B_2 \mid B^c$$

- **… function and relation images and inverse images:**

$$B ::= \ldots \mid f[B] \mid f^{-1}[B] \mid r[B] \mid r^{-1}[B]$$

- **… and higher arity relation images:**

$$B ::= \ldots \mid r[B_1, \ldots, B_{i-1}, *, B_{i+1}, \ldots, B_k]$$

$$r[B] = r[B, *]$$
$$r^{-1}[B] = r[*, B]$$

- **Linear arithmetic and set cardinality:**

$$T ::= k \mid \mathbb{Z} \mid T_1 + T_2 \mid |B|$$

- **Propositional logic**

$$F ::= B_1 \subseteq B_2 \mid T_1 < T_2 \mid F_1 \vee F_2 \mid \neg F$$

Logics of Sets, Cardinalities, and Relations

$$B ::= x \mid \emptyset \mid \mathcal{U} \mid B_1 \cup B_2 \mid B^c \mid f[B] \mid f^{-1}[B] \mid r[B] \mid r^{-1}[B]$$
$$\mid r[B_1, \ldots, B_{i-1}, *, B_{i+1}, \ldots, B_k]$$

$$T ::= k \mid \mathbb{Z} \mid T_1 + T_2 \mid |B|$$

$$F ::= B_1 \subseteq B_2 \mid T_1 < T_2 \mid F_1 \vee F_2 \mid \neg F$$

Questions:

- Is this logic (or its fragments) *decidable*?
- What is the *complexity*?

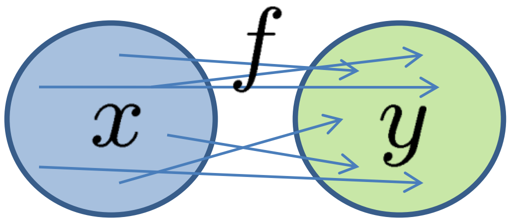
Decision Procedure

Main approach:

Reduction to *quantifier-free Boolean Algebra with Presburger Arithmetic* (QFBAPA) [Kuncak, Rinard'07]

Eliminating total function symbol:

$$f[x] = y$$

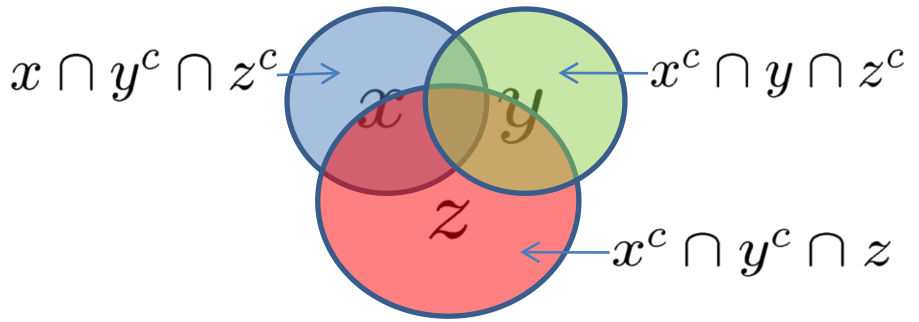


Functional consistency axiom:

$$|y| \leq |x| \wedge (|x| = 0 \Leftrightarrow |y| = 0)$$

Decision Procedure (2)

Partition domains into disjoint Venn regions:



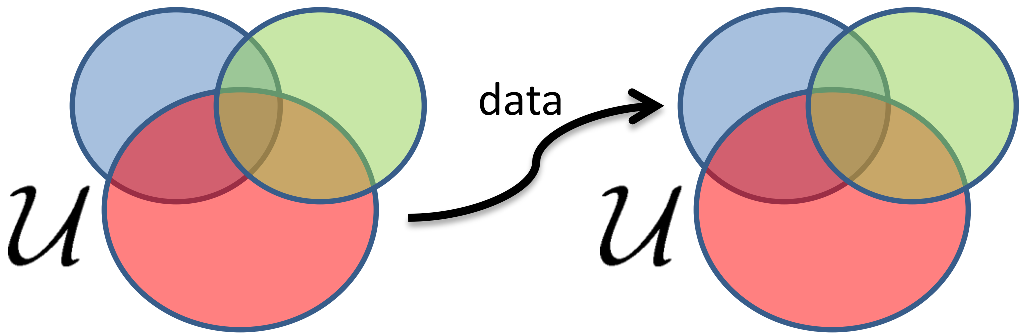
Introduce set variables for function images of Venn regions:

$$f[x \cap y] = a_{00}, f[x \cap y^c] = a_{01}$$
$$f[x^c \cap y] = a_{10}, f[x^c \cap y^c] = a_{11}$$

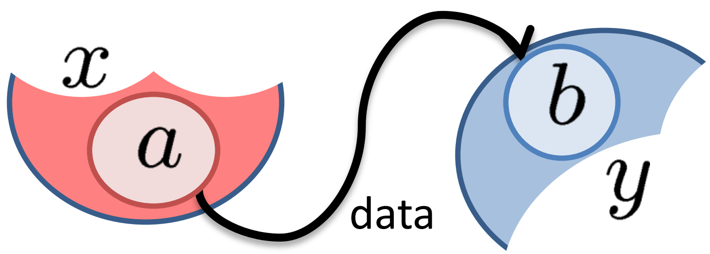
Replace function image terms with union of fresh variables:

$$f[x] = a_{00} \cup a_{01}, f[y] = a_{00} \cup a_{10}$$

Eliminating Binary Relations



Restrict relation to pairs of Venn regions:



$$b = \text{data}[x]$$
$$a = \text{data}^{-1}[y]$$
$$\text{data} \subseteq a \times b$$

Relational consistency axiom:

$$a \subseteq x \wedge b \subseteq y \wedge (|a| = 0 \Leftrightarrow |b| = 0)$$

Complexity Bounds

NEXPTIME algorithm by EXPTIME reduction to QFBAPA

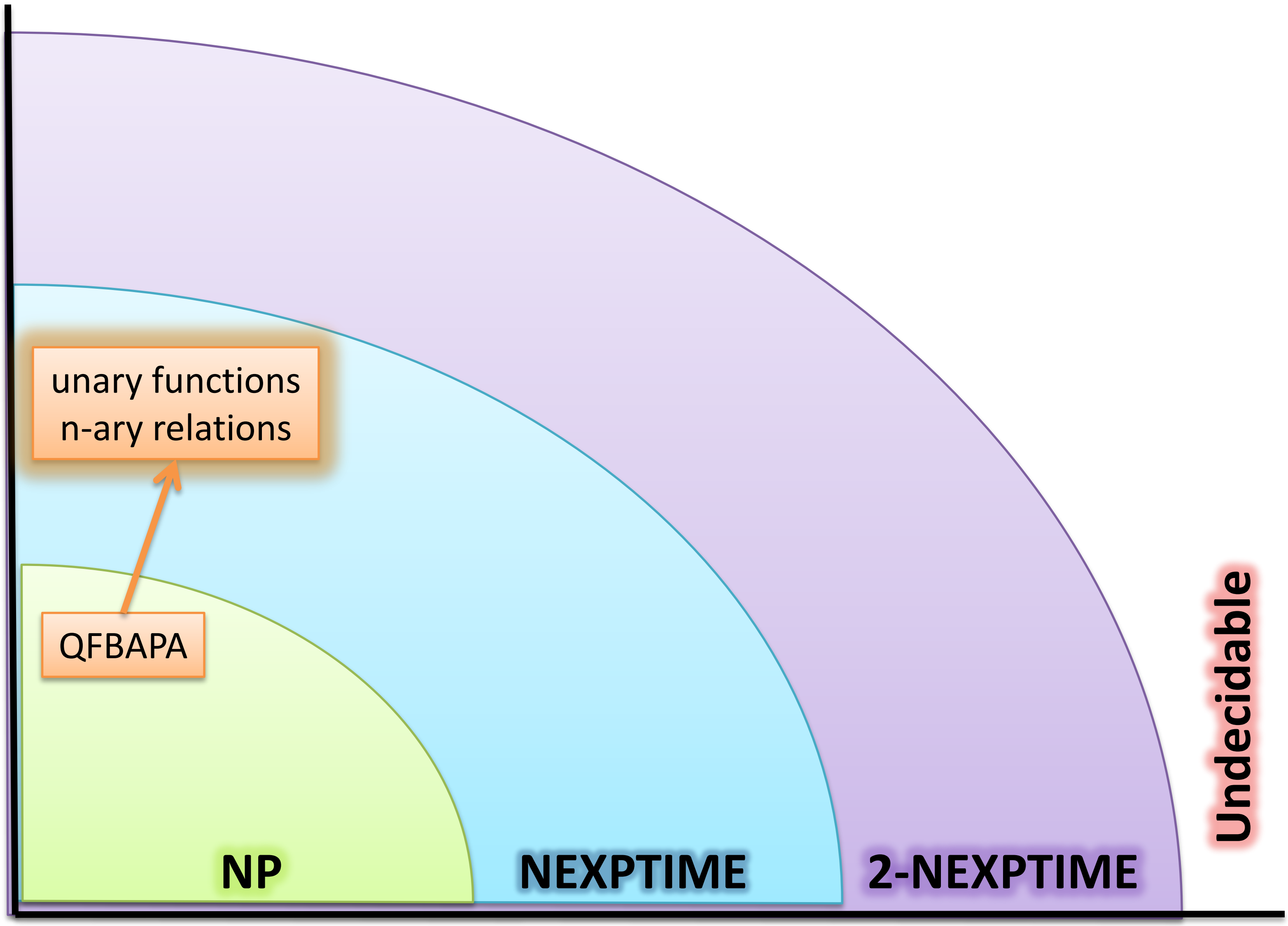
NEXPTIME lower bound via Lewis clauses [Lewis'80]:

$$\exists z.F_1 \wedge \forall y \exists x.F_2 \wedge \forall y_1 \forall y_2.F_3$$

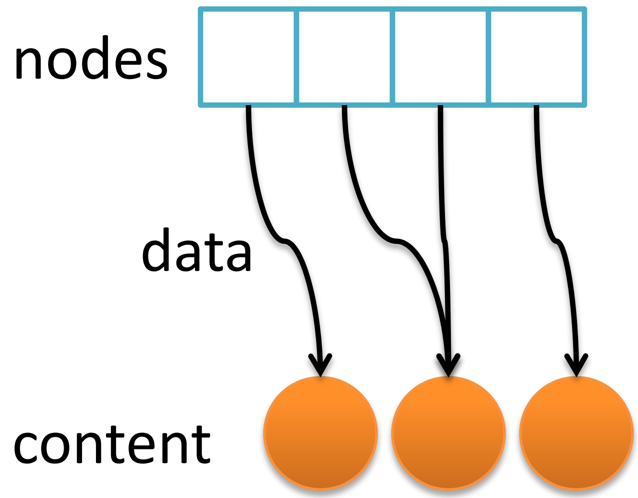
$\exists z.P(z) \vee Q(z)$	$P \cup Q \neq \emptyset$
$\forall x \exists y.P(x) \vee Q(y)$	$f[P^c] \subseteq Q$
$\forall y_1 \forall y_2.P(y_1) \vee Q(y_2)$	$P = \mathcal{U} \vee Q = \mathcal{U}$

Exact complexity bound

Decidable Extensions



Extension: Multisets

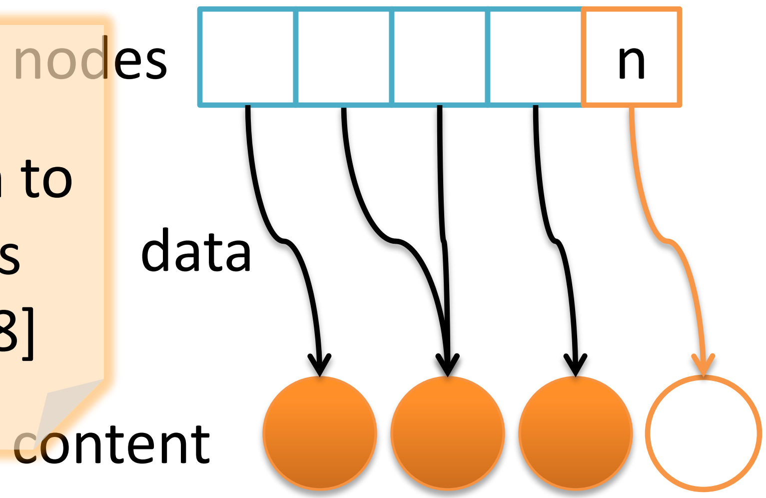


$$\mathsf{content}, \mathsf{nodes} : \mathcal{U} \to \mathbb{N}, \mathsf{data} : \mathcal{U} \to \mathcal{U}$$
$$\mathsf{content} = \mathsf{data}[\mathsf{nodes}]$$
$$\mathsf{content}(e) = |\{x \mid x \in \mathsf{nodes} \wedge \mathsf{data}(x) = e\}|$$

Extension: Multisets

NEXPTIME

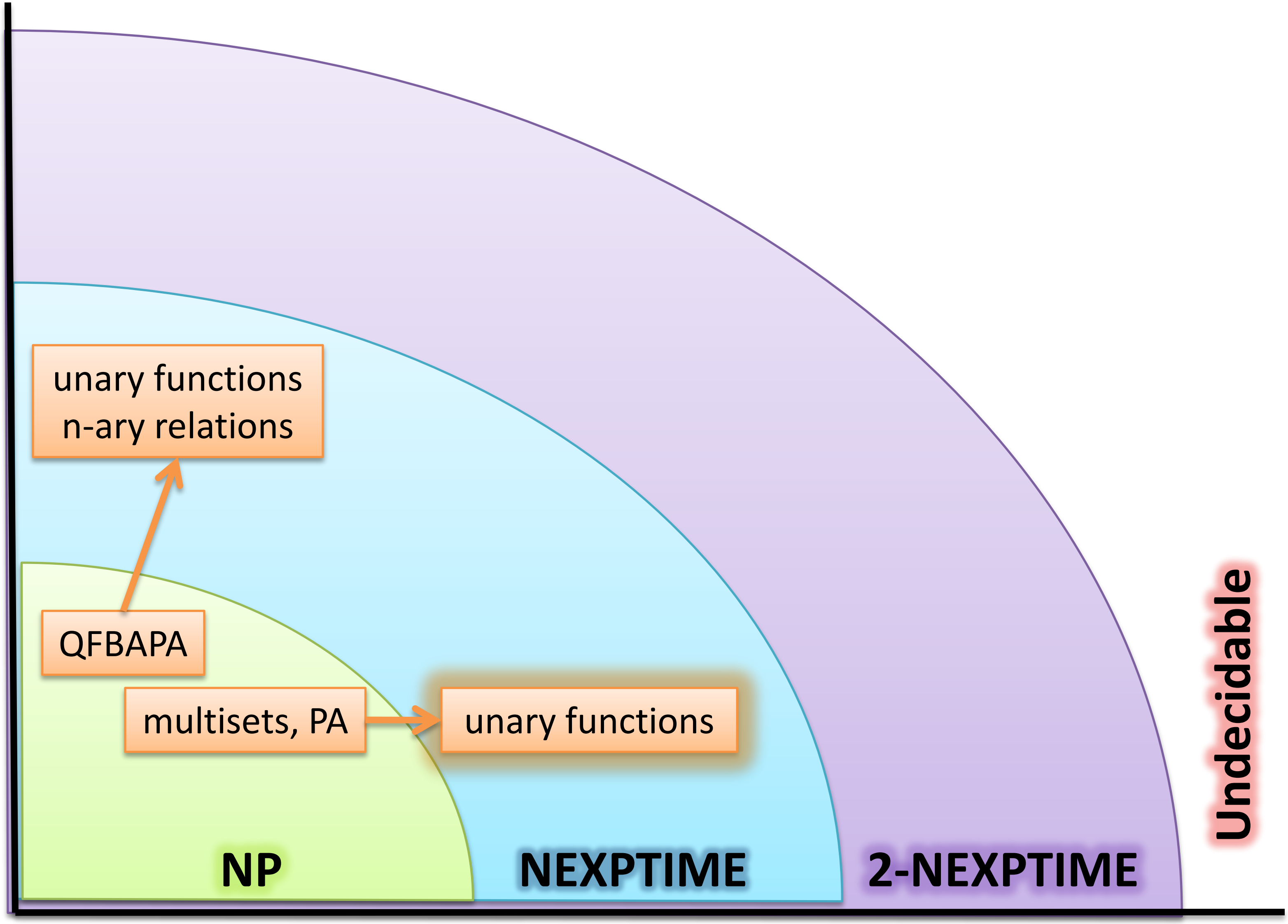
EXPTIME reduction to logic of multi-sets [Piskac, Kuncak'08]



Property: after inserting an element into a list, size of a list increases *exactly* by one

$$\mathsf{nodes}' = \mathsf{nodes} \cup \mathsf{n} \wedge |\mathsf{n}| = 1 \wedge \mathsf{n} \cap \mathsf{nodes} = \emptyset \wedge$$
$$\mathsf{content}' = \mathsf{data}[\mathsf{nodes}'] \wedge \mathsf{content} = \mathsf{data}[\mathsf{nodes}]$$
$$\Rightarrow |\mathsf{content}'| = |\mathsf{content}| + 1$$

Decidable Extensions



Extension: *n*-ary Functions

Functional consistency axiom for binary function:

necessary condition $|f[p,q]| \leq |p| \cdot |q|$

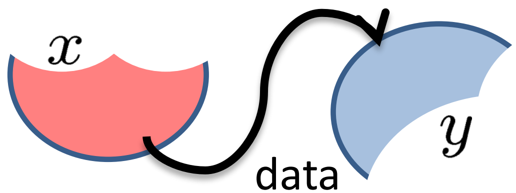
Reduction to non-linear integer constraints: $x \leq y_1 \ldots y_n$

pre-quadratic constraints [Givan et al'02]

NEXPTIME satisfiability algorithm, conjectured in NP

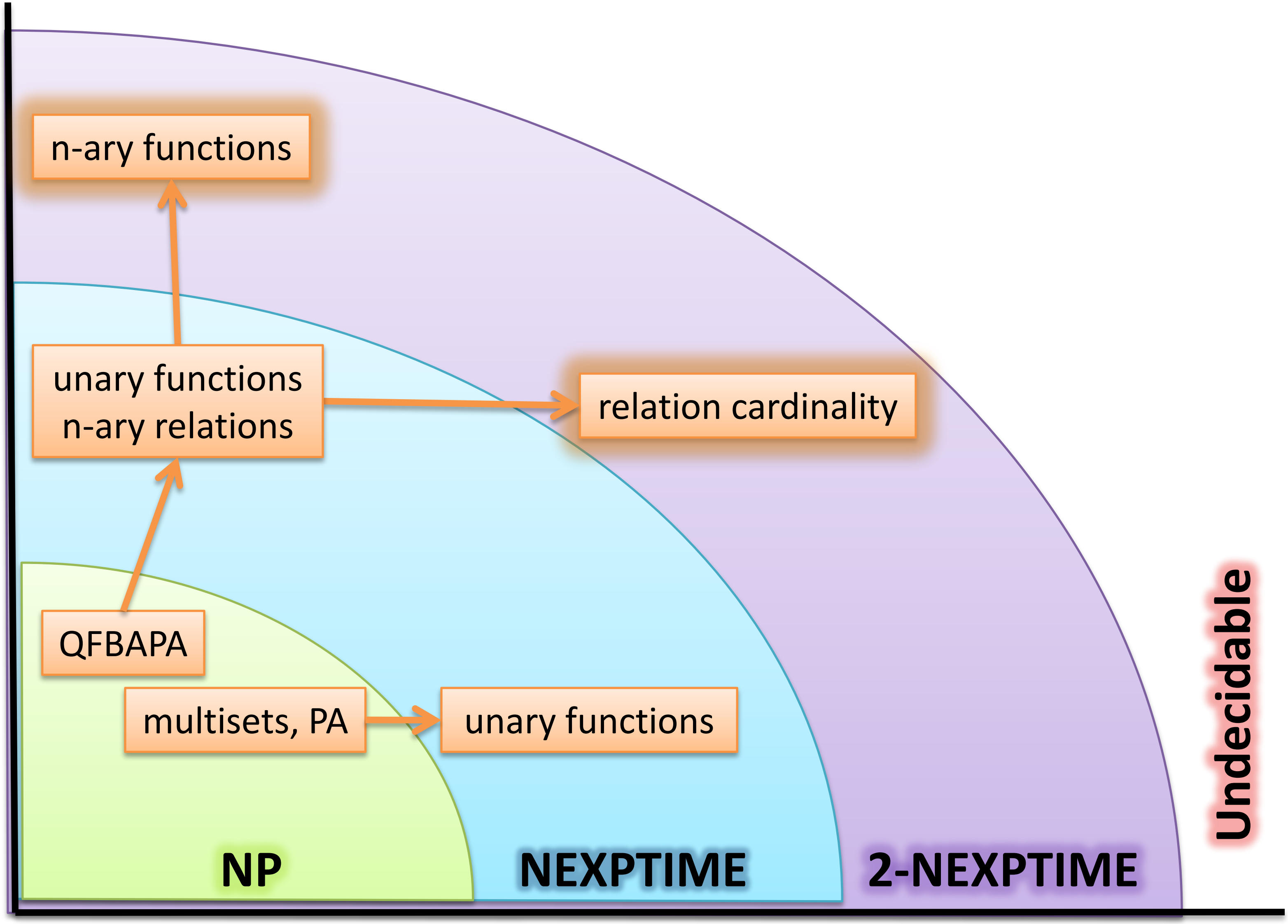
2-NEXPTIME

Another extension: relation cardinality



$|\text{data}| \leq |x| \cdot |y|$

Decidable Extensions



Undecidable Extensions

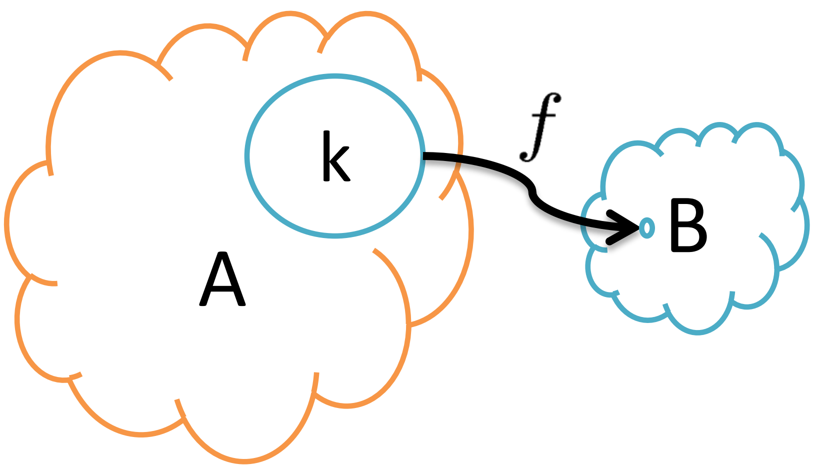
- Quantification: $\forall e, \ \exists e$
- Injective binary functions: $f\,[p, q]$
- Relation cardinality with Cartesian product:

$$|p \times q|$$

Hilbert's 10th problem

Undecidable Extensions (proof)

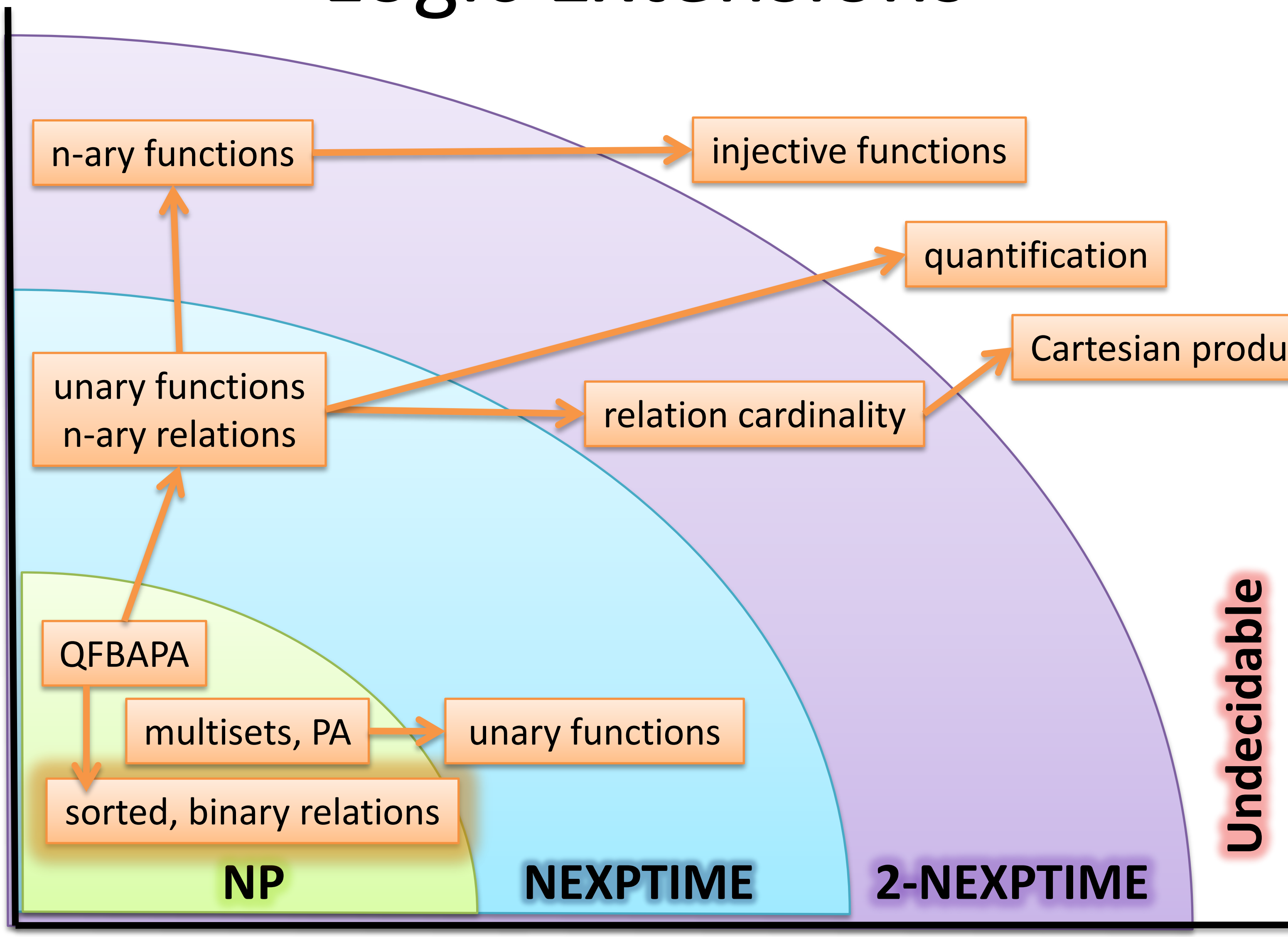
Quantification: $\forall e$



$$B = f[A] \wedge$$
$$\forall e. e \subseteq B \wedge |e| = 1 \Rightarrow$$
$$\left| f^{-1}[e] \right| = k$$

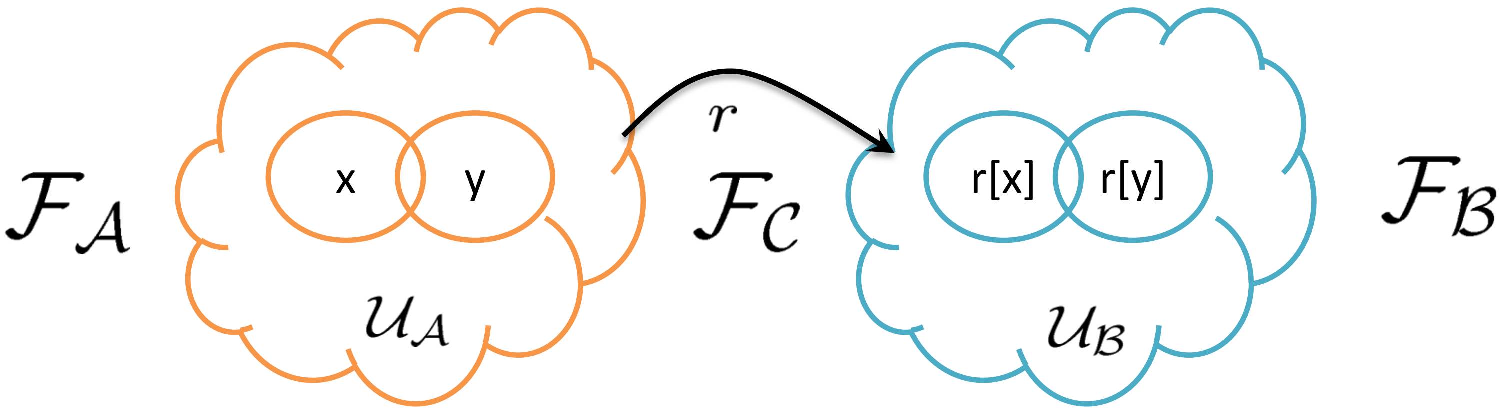
$$|A| = k \cdot |B|$$

Logic Extensions



NP-complete Two-Sorted Fragment

$$x \subseteq y \Rightarrow r[x] \subseteq r[y]$$



Theorem: If a formula has a model, then it has a *completed model* α (i.e. α(*r)* is a union of products of Venn regions)

Theorem: If a formula has a model, then it has a *sparse model* (in which only polynomially many Venn regions are non-empty)

Sparsification technique

Summary of Our Results

Class of formulas	Expressive power	Lower bound	Upper bound
$\lvert k\rvert = 1 \wedge \lvert \mathrm{data}[k]\rvert = 1$	Sets, unary total functions, n-ary relations, set cardinality	NEXPTIME	NEXPTIME
$x \subseteq y \Rightarrow f[x,\mathcal{U}] \subseteq f[y,\mathcal{U}]$	Sets, n-ary total functions, n-ary relations, set cardinality	NEXPTIME	2-NEXPTIME
$\lvert r\rvert \geq \lvert r[\mathcal{U},*]\rvert$	Sets, unary total functions, n-ary relations, relation cardinalities	NEXPTIME	2-NEXPTIME
$\lvert f[x]\rvert = \lvert x\rvert$	Multisets, unary total functions, collection cardinality	NEXPTIME	NEXPTIME
$x \subseteq y \Rightarrow r[x] \subseteq r[y]$	Sorted sets, set cardinality, binary relations acyclic over sorts	NP	NP
Sets, unary functions, set quantification		undecidable	
Sets, injective binary functions		undecidable	
Sets, relation cardinality, Cartesian product		undecidable	

Related Work

- NP-completeness of *quantifier free Boolean algebra with Presburger arithmetic (QFBAPA)* [Kuncak, Rinard’07]
- NP-completeness of multi-sets with cardinality constraints [Piskac, Kuncak’08]
- Tarskian set constraints [Givan et al’02]
  - Tarskian interpretation of function symbols, subsumptions
  - no symbolic cardinality
- certain Description logics: set ≡ concept, relation ≡ role
  - ALC, bridging functions [Ohlbach, Koehler’99]
  - no symbolic cardinality
- two-variable logic with counting [Pacholski et al’00]
  - images of n-ary relations are inexpressible

Conclusion

- Proof techniques for QFBAPA are effective in dealing with functions and relations
- Achieved optimal complexity and gave "natural" reduction algorithms
- BAPA reduction allows combination with other BAPA-reducible logics (WS2S, $\mathcal{C}_2$)
  – See Viktor Kuncak's tutorial tomorrow!