Universal Algebra for Constraint Satisfaction

Andrei Krokhin
Durham University

Disclaimer: these slides contain inaccuracies

Three Well-Known Problems

SAT: is a given propositional formula in CNF satisfiable?

$$F = (\neg x \vee y \vee \neg z) \wedge (x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z)$$

Linear Equations: does a given system of linear equations have a solution in the fixed field K?

$$\begin{cases} 2x + 2y + 3z = 1 \\ 3x - 2y - 2z = 0 \\ 5x - y + 10z = 2 \end{cases}$$

Graph 3-colouring: given a graph, can its vertices be coloured with 3 colours so that adjacent vertices are different colour?

Constraint Satisfaction Problem: 3 Forms

- Satisfiability (Logic, Databases)

  Given a finite structure $\mathcal{B}$ and a $\exists\wedge$-FO sentence φ, does $\mathcal{B}$ satisfy φ?

- Variable-value (AI, Algebra)

  Given finite sets A (variables), B (values), and a set of constraints $\{(\overline{s}_1, R_1), \ldots, (\overline{s}_q, R_q)\}$ over A, is there a function $\varphi : A \to B$ such that $\varphi(\overline{s}_i) \in R_i$ for all i?

- Homomorphism (Model Theory, Graph Theory)

  Given two finite similar relational structures, $\mathcal{A} = (A;\ R_1^{\mathcal{A}}, \ldots, R_k^{\mathcal{A}})$ and $\mathcal{B} = (B;\ R_1^{\mathcal{B}}, \ldots, R_k^{\mathcal{B}})$, is there a homomorphism $h : \mathcal{A} \to \mathcal{B}$?

Constraint Languages

Fix a finite set D.

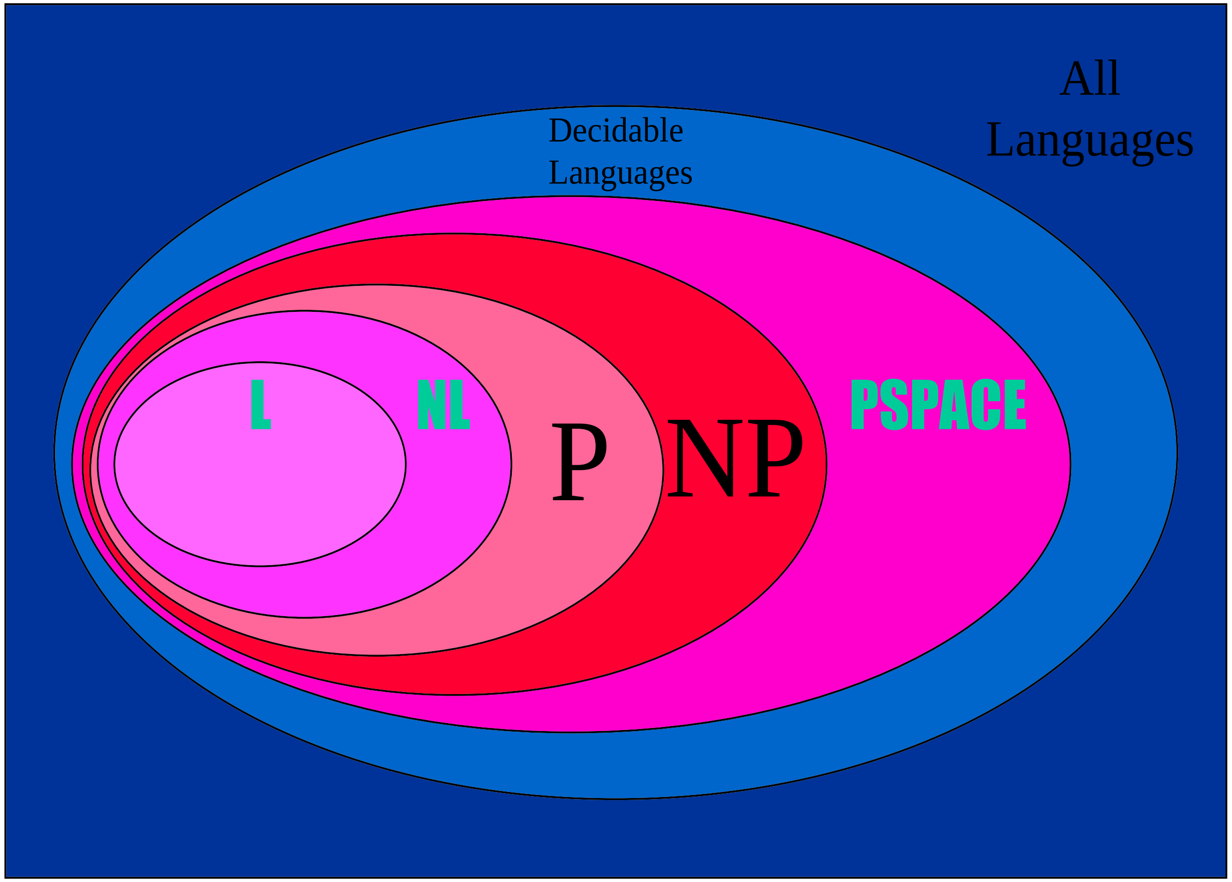
Definition 1 *A constraint language is any finite set* Γ *of relations on* D*. The problem* CSP(Γ) *is the restriction of* CSP *where all constraint relations* R_i *must belong to* Γ*.*

Equivalently, fix target structure $\mathcal{B}$ (aka template) and ask whether a given structure $\mathcal{A}$ homomorphically maps to $\mathcal{B}$.
Notation: $\mathrm{CSP}(\mathcal{B}) = \{\mathcal{A} \mid \mathcal{A} \to \mathcal{B}\}$.

The structure $\mathcal{B}$ is obtained from Γ by indexing relations.

NB. For a digraph $\mathcal{H}$, CSP($\mathcal{H}$) is known as $\mathcal{H}$-COLOURING.
For a structure $\mathcal{B}$ on $\{0, 1\}$, CSP($\mathcal{B}$) is a variant of SAT.

Main Complexity Classes



Examples

- Let $D = \{0, 1\}$ and $R = \{0, 1\}^3 \setminus \{(0, 0, 0), (1, 1, 1)\}$.
  If $\Gamma = \{R\}$ then CSP(Γ) is NOT-ALL-EQUAL SAT.
  This problem is **NP**-complete.
- Let $D = \{0, 1\}$ and $R = \{(x, y, z) \mid x \wedge y \to z\}$.
  If $\Gamma = \{R, \{0\}, \{1\}\}$ then CSP(Γ) is HORN 3-SAT.
  This problem is **P**-complete.
- Let $D = \{0, 1\}$ and $\Gamma = \{\leq, \{0\}, \{1\}\}$. Then CSP($\Gamma$) is the complement of PATH (i.e., UNREACHABILITY).
  Think: An instance is satisfiable iff it contains no path of the form $1 = x_1 \leq x_2 \leq \ldots \leq x_n = 0$.
  This problem is **NL**-complete.

More Examples

- If $\Gamma = \{\neq_D\}$ where $\neq_D$ is the disequality relation on D and $|D| = k$ then CSP(Γ) is GRAPH k-COLOURING. Think: elements of D are colours, variables are the nodes, and constraints $x \neq_D y$ are the edges of graph. Belongs to **L** if $k \leq 2$, **NP**-complete for $k \geq 3$.

- For a semigroup S on D, let $R_S = \{(x, y, z) \mid xy = z\}$. If $\Gamma = \{R_S\} \cup \{\{d\} \mid d \in D\}$ then CSP(Γ) is the problem of solving SYSTEMS OF EQUATIONS over S.

  Think: transform each equation $w_1 = w_2$ into pair $w_1 = u$ and $w_1 = u$, and then iteratively transform each $xyz \ldots = a$ into pair $xy = x'$ and $x'z \ldots = a$.

Classification Problems & The Holy Grail

The main classification problems about problems CSP(Γ):

1. Classify CSP(Γ) w.r.t. computational complexity, (i.e., w.r.t. membership in a given complexity class)
2. Classify CSP($\mathcal{B}$) w.r.t. descriptive complexity, (i.e., w.r.t. definability in a given logic)
3. Classify CSP($\mathcal{B}$) w.r.t. solvability by a given algorithm

Conjecture 1 (Feder,Vardi '98)
Dichotomy Conjecture: for each Γ, *the problem* CSP(Γ) *is either tractable (i.e., in* **P***) or* **NP***-complete.*

Datalog

For logical definability, we will use Datalog (and FO) .
A Datalog program has EDBs - relations from structure,
and IDBs - auxiliary predicates. One IDB is the goal.

$$
\begin{array}{lcl}
odd(X,Y) & :- & edge(X,Y) \\
odd(X,Y) & :- & odd(X,Z), edge(Z,T), edge(T,Y) \\
goal & :- & odd(X,X)
\end{array}
$$

A Datalog program recursively computes the IDBs (from EDBs).
Intuition: locally derive new constraints, trying to get a contradiction (to certify that there's no solution).

Definability in Datalog

"co-CSP($\mathcal{B}$) is definable by a Datalog program" means that the program accepts precisely structures $\mathcal{A}$ with $\mathcal{A} \not\to \mathcal{B}$.

Example 1 (Horn 3-Sat) co-CSP($\mathcal{B}_{H3Sat}$) *is definable by the following Datalog program*

$$
\begin{aligned}
acc(X) &\ :-\ O(X) \\
acc(Z) &\ :-\ acc(X), acc(Y), R(X,Y,Z) \\
goal &\ :-\ Z(X), acc(X)
\end{aligned}
$$

If co-CSP($\mathcal{B}$) is definable in Datalog then CSP($\mathcal{B}$) is in **P**.
Intuition: IDBs have bounded arity, so the program can do only polynomially many steps before stabilising.

Invariance and Polymorphisms

Definition 2 *An m-ary relation R is invariant under an n-ary operation f (or f is a polymorphism of R) if, for any tuples $\bar{a}_1 = (a_{11}, \ldots, a_{1m}), \ldots, \bar{a}_n = (a_{n1}, \ldots, a_{nm}) \in R$, the tuple obtained by applying f componentwise belongs to R.*

$$
\begin{array}{cccccccc}
 & f & & & & f & & \\
(& a_{11} & , & \ldots & , & a_{1m} &) & \in R \\
 & \vdots & & \vdots & & \vdots & & \vdots \\
(& a_{n1} & , & \ldots & , & a_{nm} &) & \in R \\
\hline
(& f(a_{11}, \ldots, a_{n1}) & , & \ldots & , & f(a_{1m}, \ldots, a_{nm}) &) & \in R
\end{array}
$$

Example

Consider the relation $R = \{0,1\}^3 \setminus \{(1,1,0)\}$ (from $\mathcal{B}_{H3Sat}$)

- the binary operation min is a polymorphism of R.

$$
\begin{array}{ccccccccl}
 & \min & & \min & & \min & & & \\
(& ? & , & ? & , & ? &) & \in R \\
(& ? & , & ? & , & ? &) & \in R \\
\hline
(& 1 & , & 1 & , & 0 &) & &
\end{array}
$$

- the binary operation max is not.

Polymorphisms of a Structure

- If f is a polymorphism of each relation in $\mathcal{B}$ then f is called a polymorphism of $\mathcal{B}$.
- Example: min is a polymorphism of $\mathcal{B}_{H3Sat}$.
- Equivalently, f is a homomorphism from $\mathcal{B}^n$ to $\mathcal{B}$.
- For a digraph: an edge-preserving mapping, i.e.

$$\begin{array}{ccccccc} a_1 & a_2 & \ldots & a_n & & f(a_1, a_2, \ldots, a_n) \\ \downarrow & \downarrow & \ldots & \downarrow & \Rightarrow & \downarrow \\ b_1 & b_2 & \ldots & b_n & & f(b_1, b_2, \ldots, b_n) \end{array}$$

From Structures to Algebras

Any structure $\mathcal{B}$ is associated an algebra $\mathbf{A}_{\mathcal{B}} = (B, \mathrm{Pol}(\mathcal{B}))$ where $\mathrm{Pol}(\mathcal{B})$ is the set of all polymorphisms of $\mathcal{B}$.

Fact 1 (Bulatov, Jeavons, K '05 + Larose, Tesson '09)
The (computational and descriptive) complexity of $\mathrm{CSP}(\mathcal{B})$ *is completely determined by the properties of* $\mathbf{A}_{\mathcal{B}}$.

Intuition: for any R, R is $(\exists \wedge =)$-definable in $\mathcal{B}$ iff $\mathrm{Pol}(\mathcal{B}) \subseteq \mathrm{Pol}(R)$, i.e. $\mathrm{Pol}()$ controls expressive power.

- Do we gain anything by using algebras?
- Why swap relations for operations?
- Algebras have much more structure than structures!

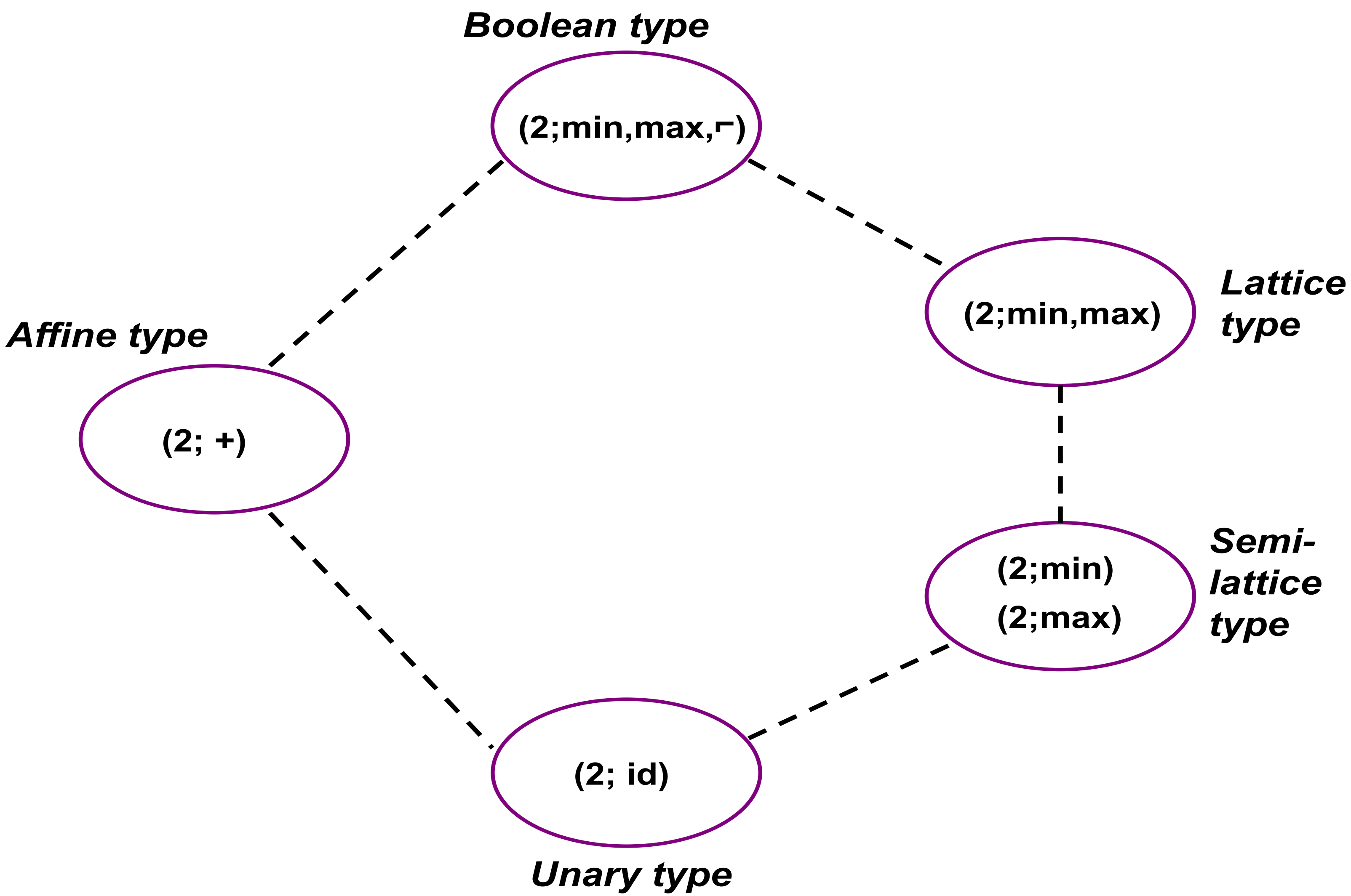
The Five Types (in Conservative Algebras)

Let $\mathcal{B}$ contain all unary relations and fix $X = \{0, 1\} \subseteq B$. Each $g \in \mathrm{Pol}(\mathcal{B})$ preserves X (i.e. $\mathbf{A}_\mathcal{B}$ is conservative). The set X can be assigned (in $\mathbf{A}_\mathcal{B}$) one of the five types:

By Post'41, there exist only five possibilities for the set $\{f(x_1, \ldots, x_n, 0, 1) \mid f = g_{|\{0,1\}}$ with $g \in \mathrm{Pol}(\mathcal{B})\}$:

1. essentially unary op's $s(x_1, \ldots, x_n) = t(x_i)$ unary
2. all linear Boolean op's $\sum a_i x_i + a_0\ (mod\ 2)$ affine
3. all possible Boolean operations Boolean
4. all monotone Boolean operations lattice
5. all op's of the form $\min(x_1, \ldots, x_n)$ and 0,1 semilattice

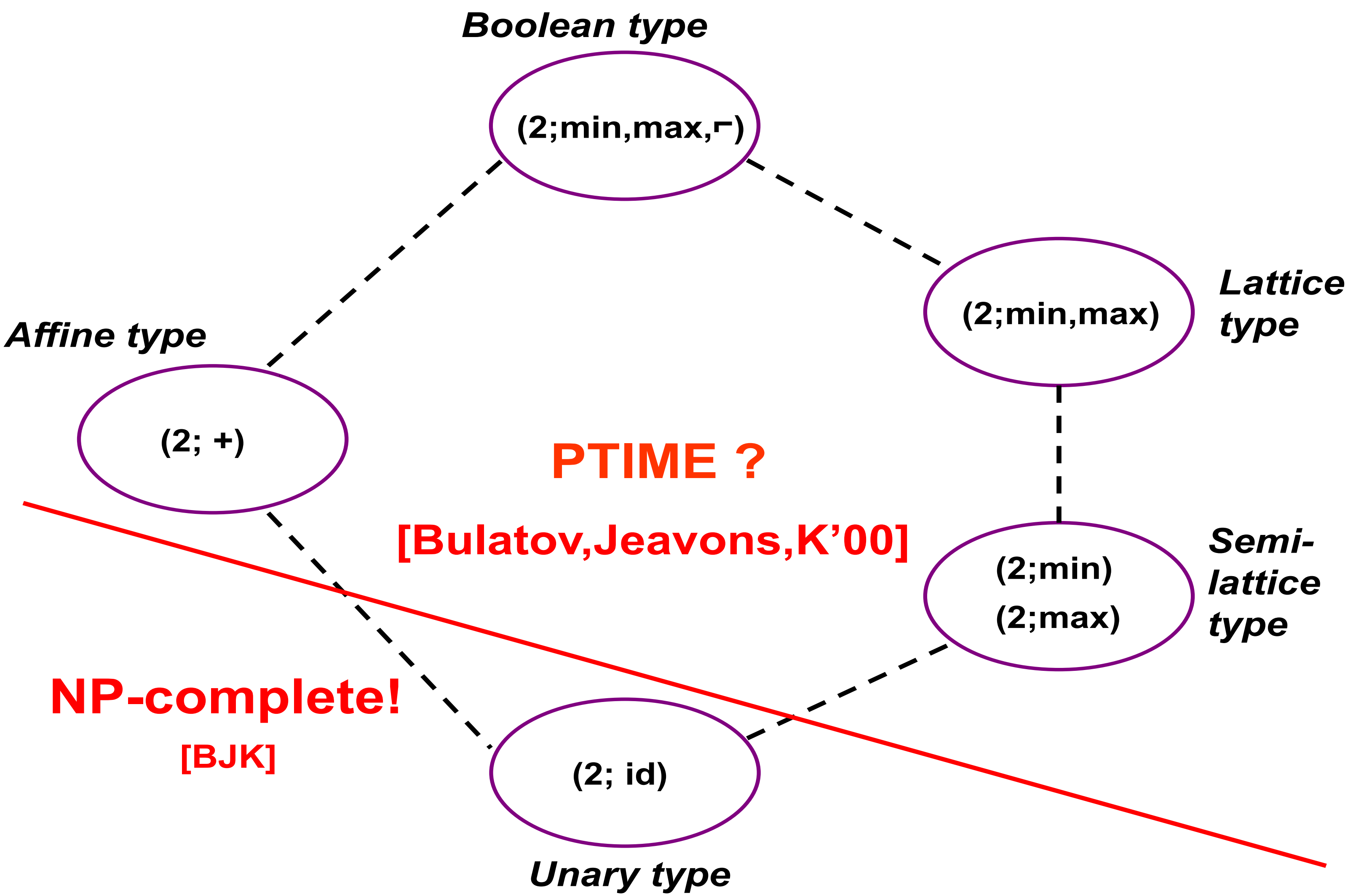
Ordering of Types



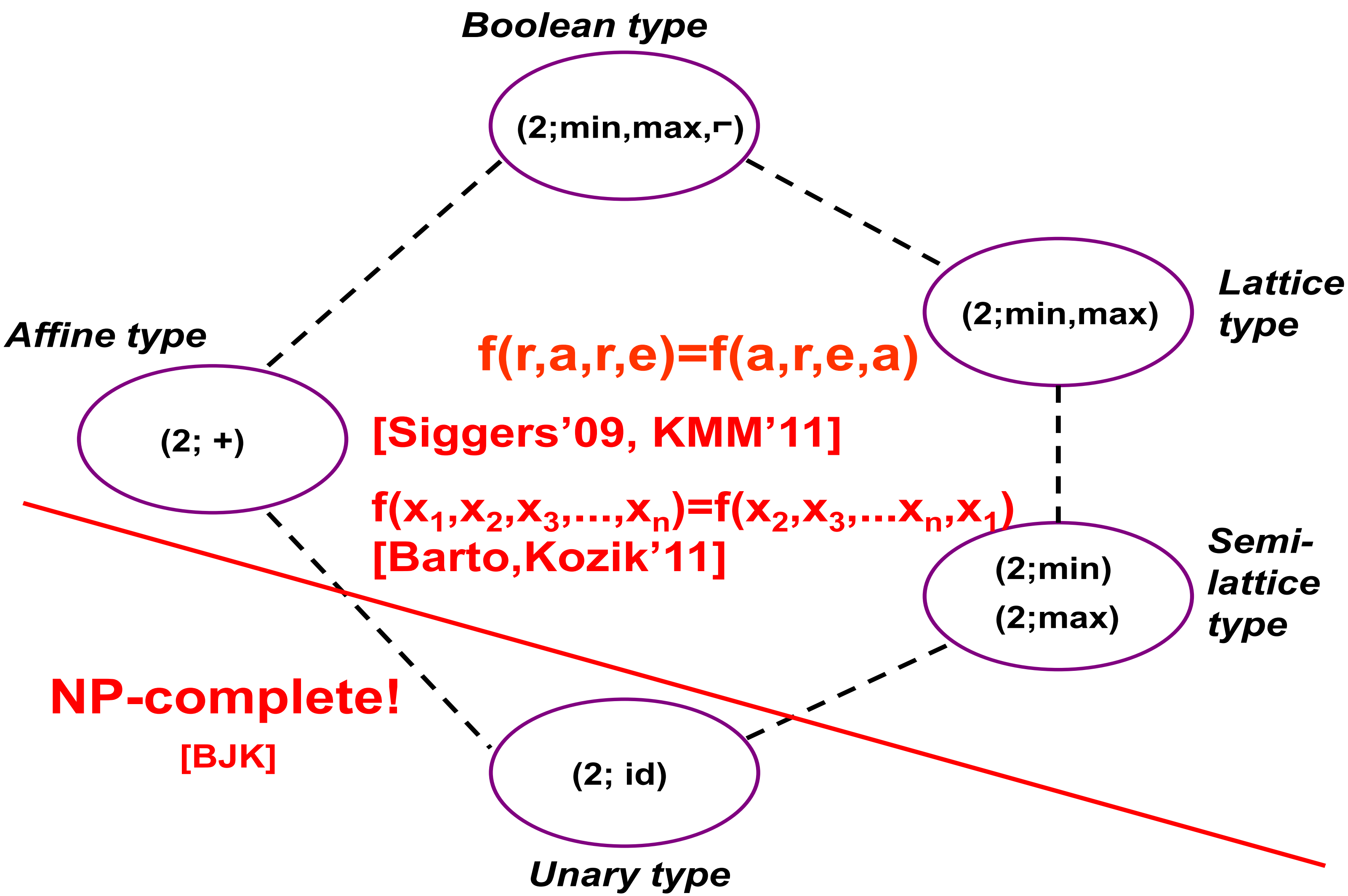
The Five Types in General Algebras

- Tame Congruence Theory (Hobby, McKenzie, 80's)
- The same five basic types of "local" behaviour
- "local" has a much more involved meaning
- Very advanced theory (focused on congruences)
- Presence of some types in var($\mathbf{A}_{\mathcal{B}}$) - hardness for CSP
- Absence of those types - positive results for CSP
  - Requires new theory focused on relations
  - Massive attack by universal algebraists
    Barto, Kozik, Bulatov, McKenzie, Valeriote,
    Willard, Maroti, Markovic, many others

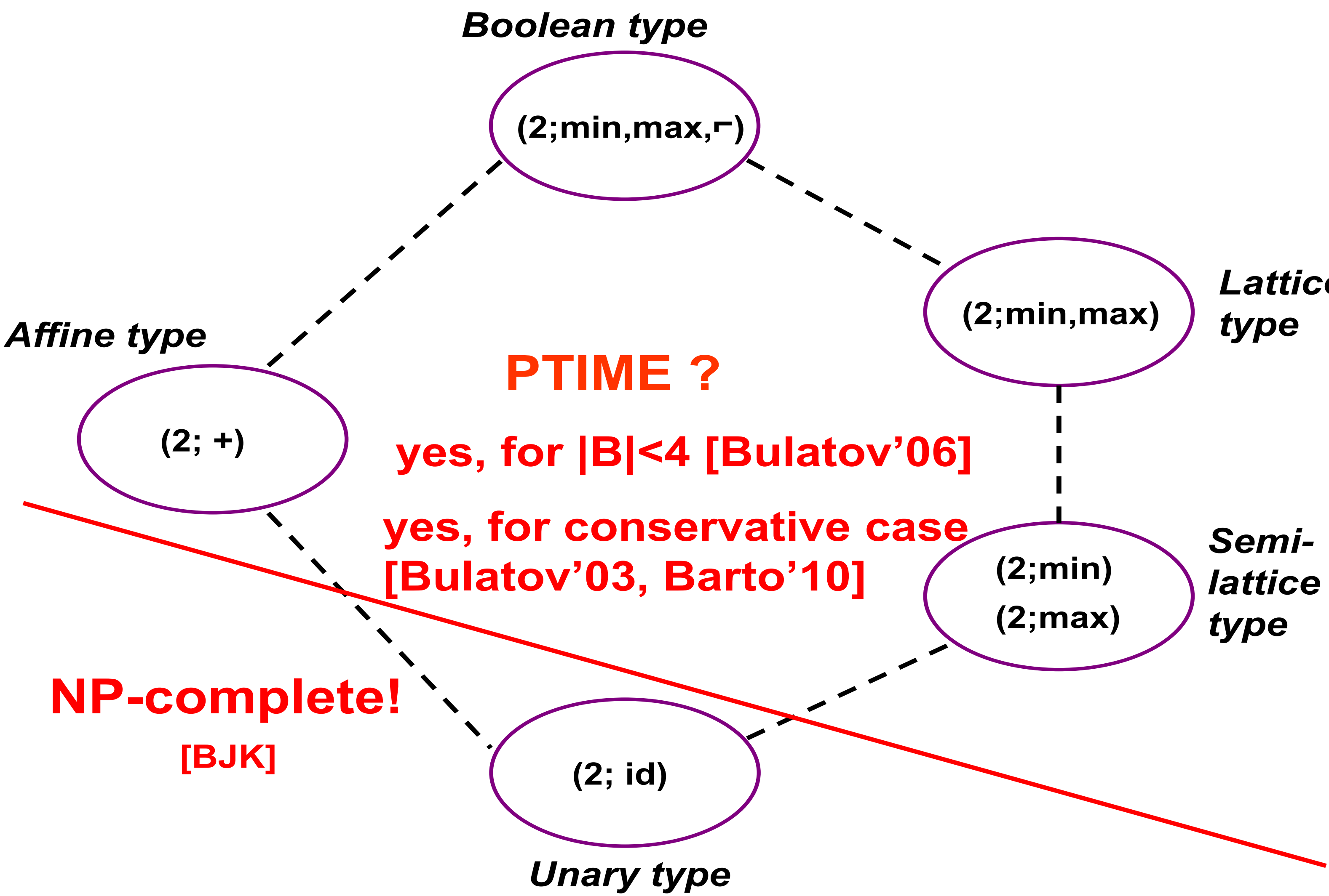
The Algebraic Dichotomy Conjecture



Some Algebraic Dichotomy Results



Some Algebraic Dichotomy Results

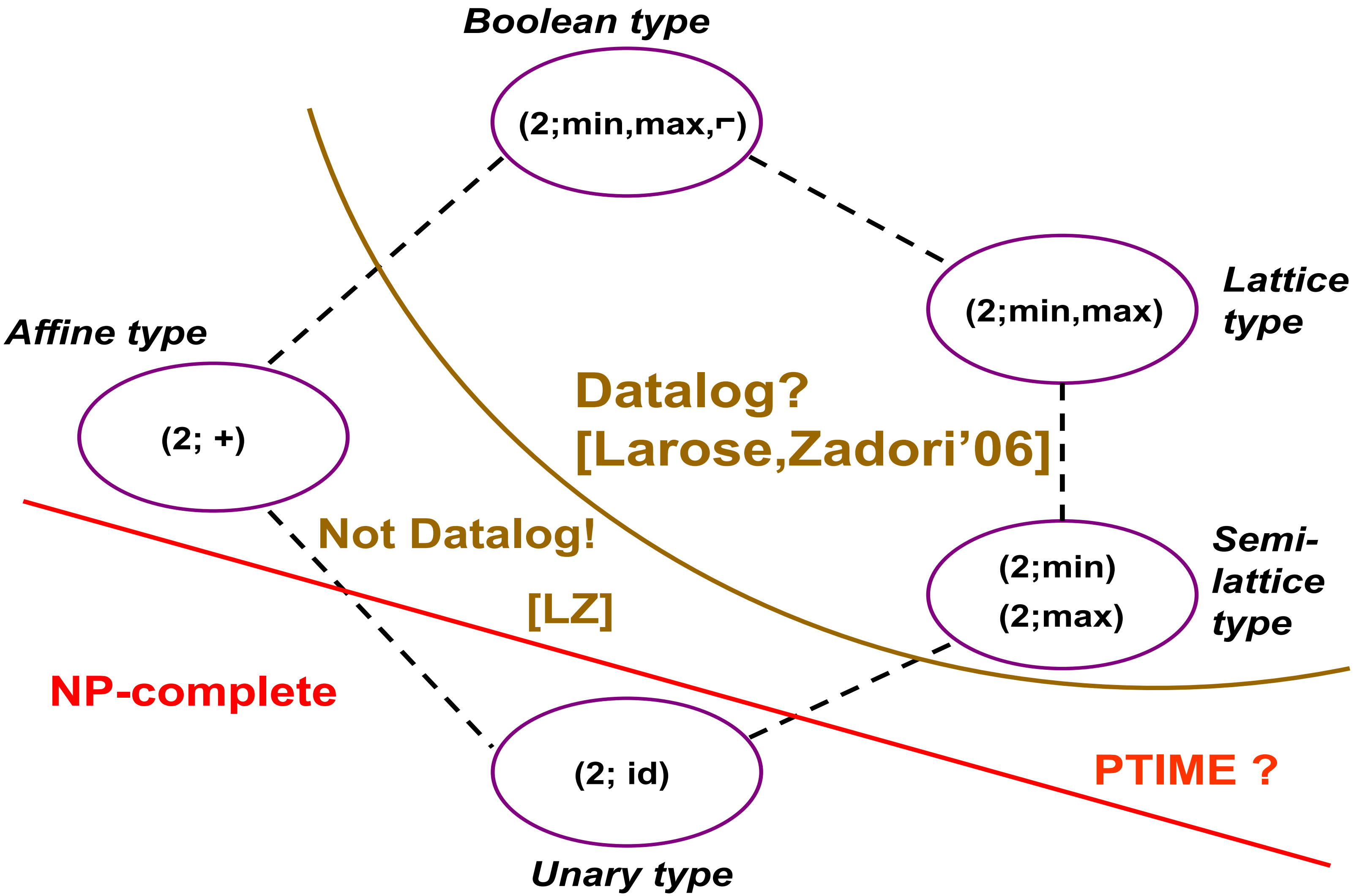


A Bait for Semigroup Theorists ...

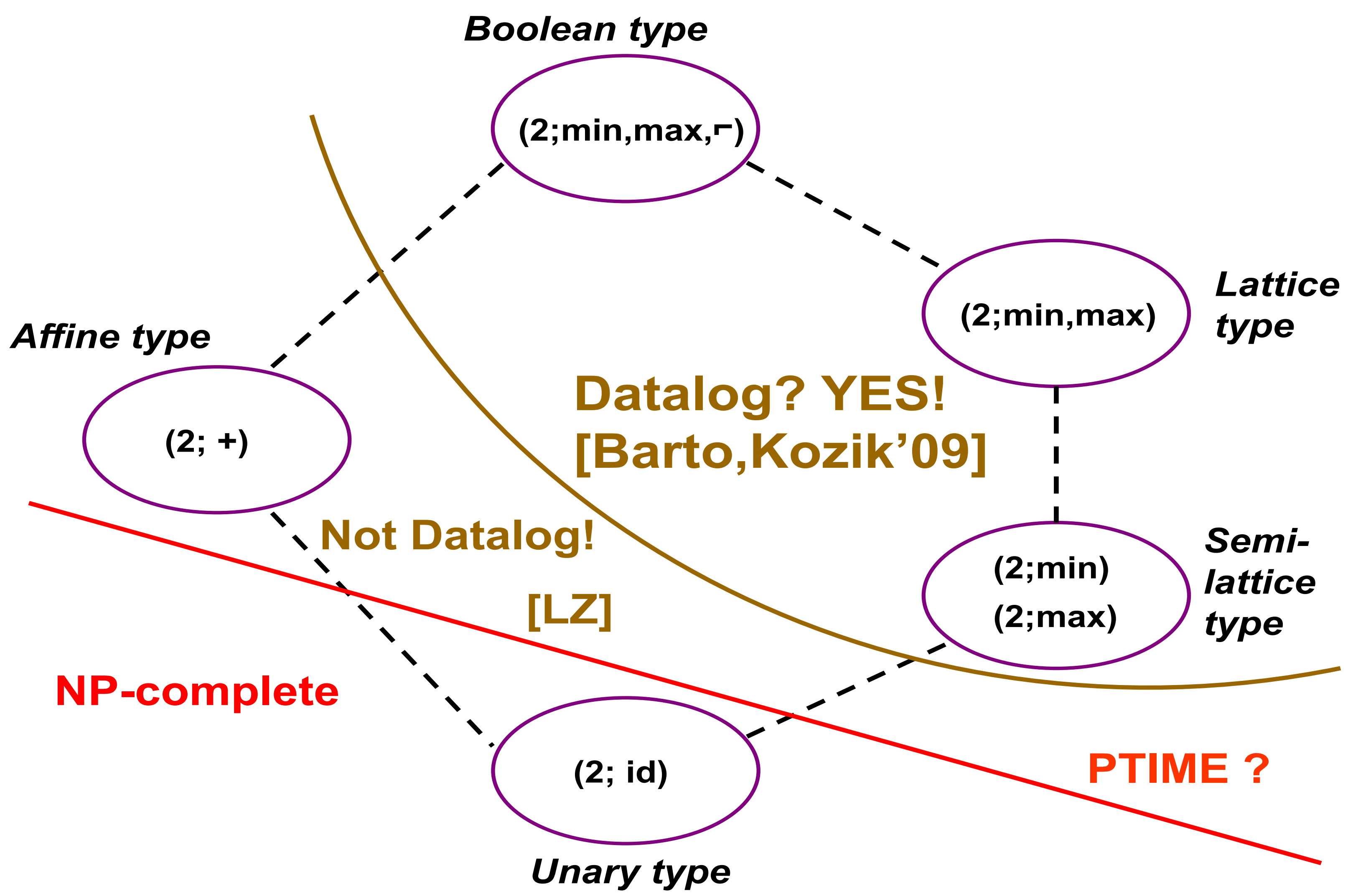
Theorem 1 (Klíma, Tesson, Thérien '07)
For every structure $\mathcal{B}$*, there is a finite semigroup* S *satisfying* $x^2 = x$ *and* $xyz = yxz$ *and such that* $\mathrm{CSP}(\mathcal{B})$ *is poly-time equivalent to* SYSTEMS OF EQUATIONS *over* S*.*

There's a full classification result for monoids, though ...

The Datalog Conjecture



The Datalog Theorem



Linear and Symmetric Datalog

A Datalog program is said to be linear if each rule contains at most one occurrence of an IDB in the body.

In other words, each rule looks like this

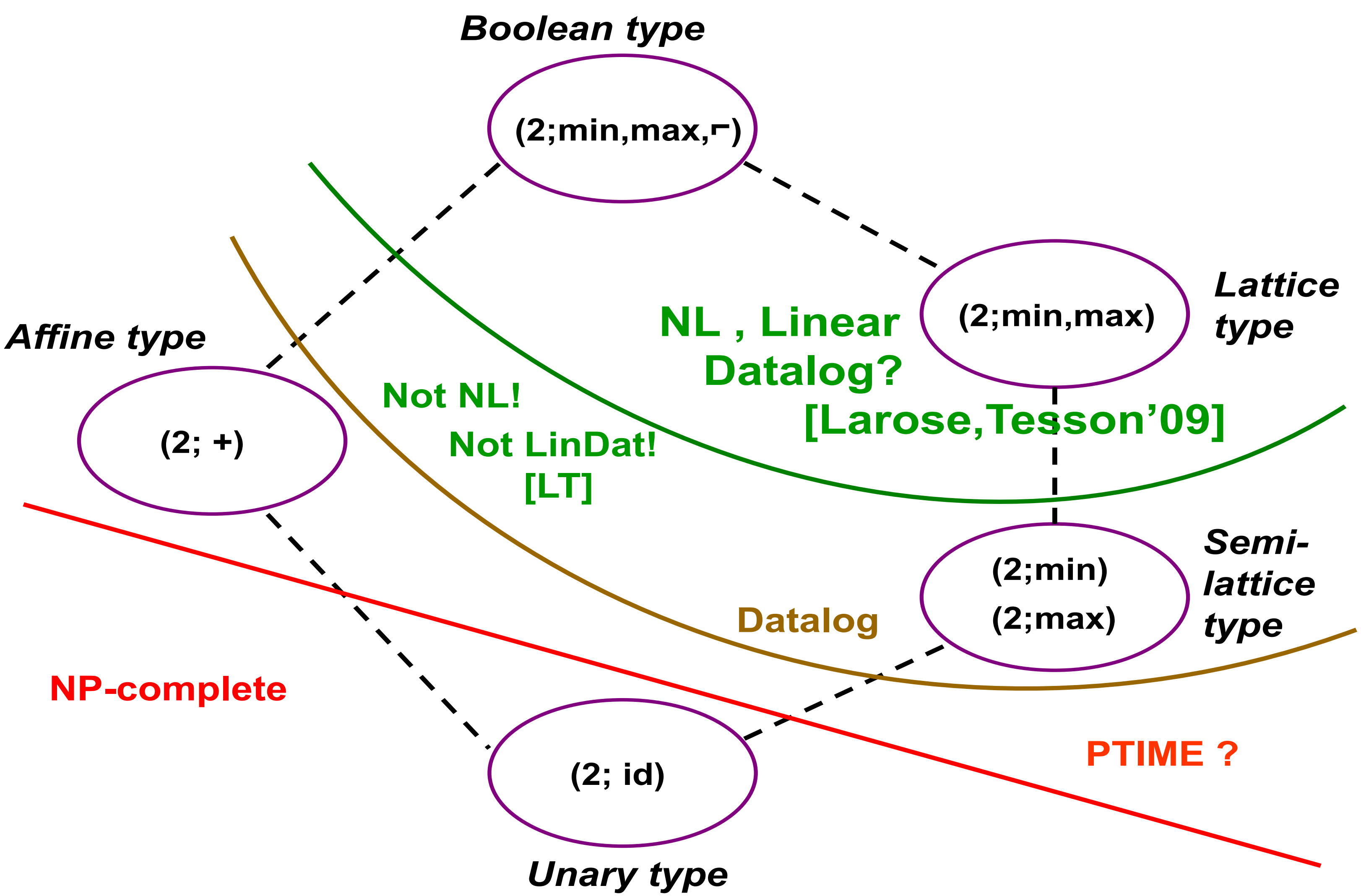
$$\theta_1(x, y) \ :- \ [\theta_2(w, u, x),]R_1(x, y, z), R_2(x, w)$$

where θ_i's are the only IDBs in it.

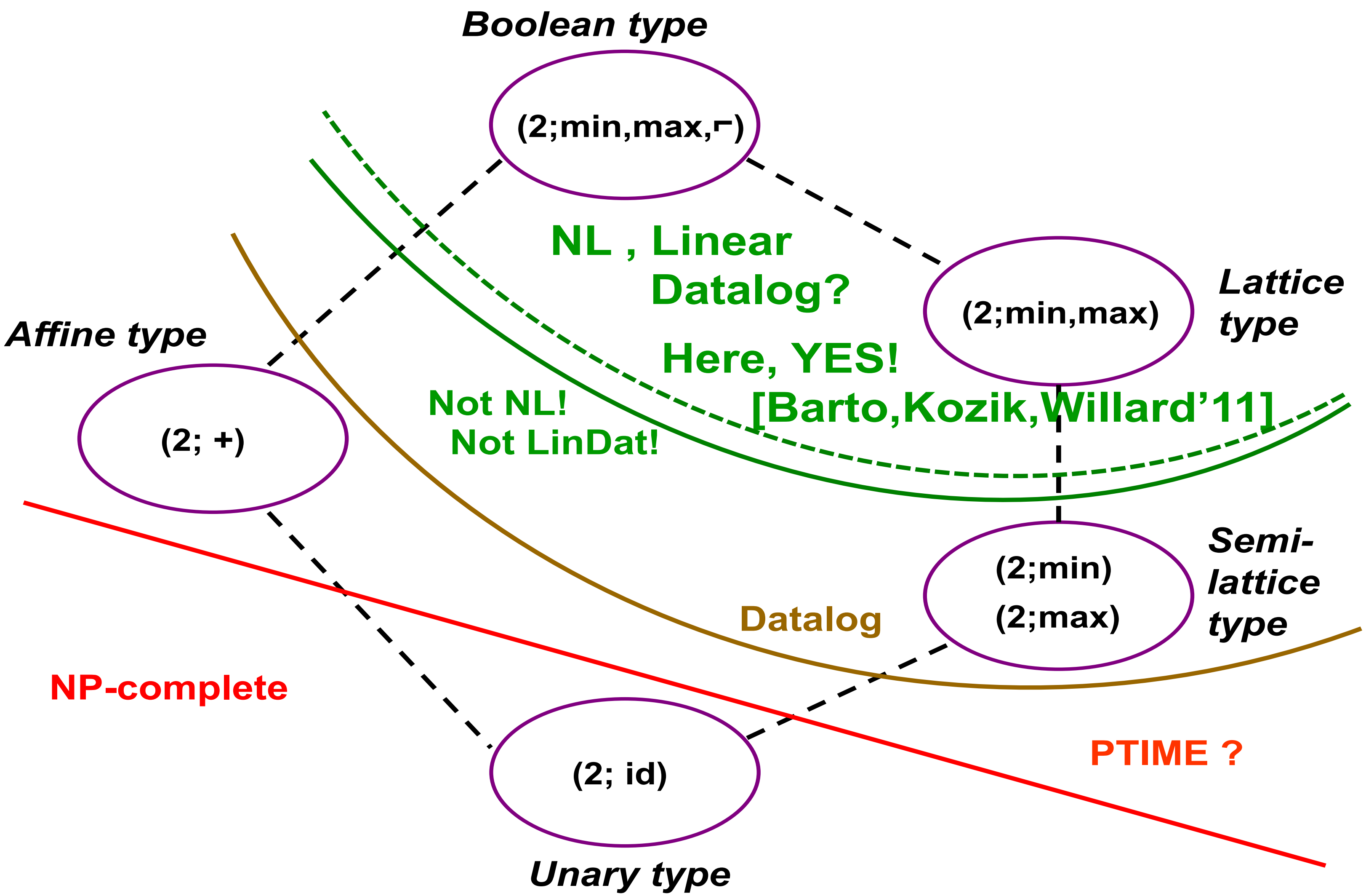
A Datalog program is said to be symmetric if (i) it is linear and (ii) it is invariant under symmetry of rules.

Definability in LinDat $\Rightarrow$ **NL**, in SymDat $\Rightarrow$ **L**.
[Dalmau'05, Egri,Larose,Tesson'07]
Idea: program looks for a derivation path that ends in *goal*.

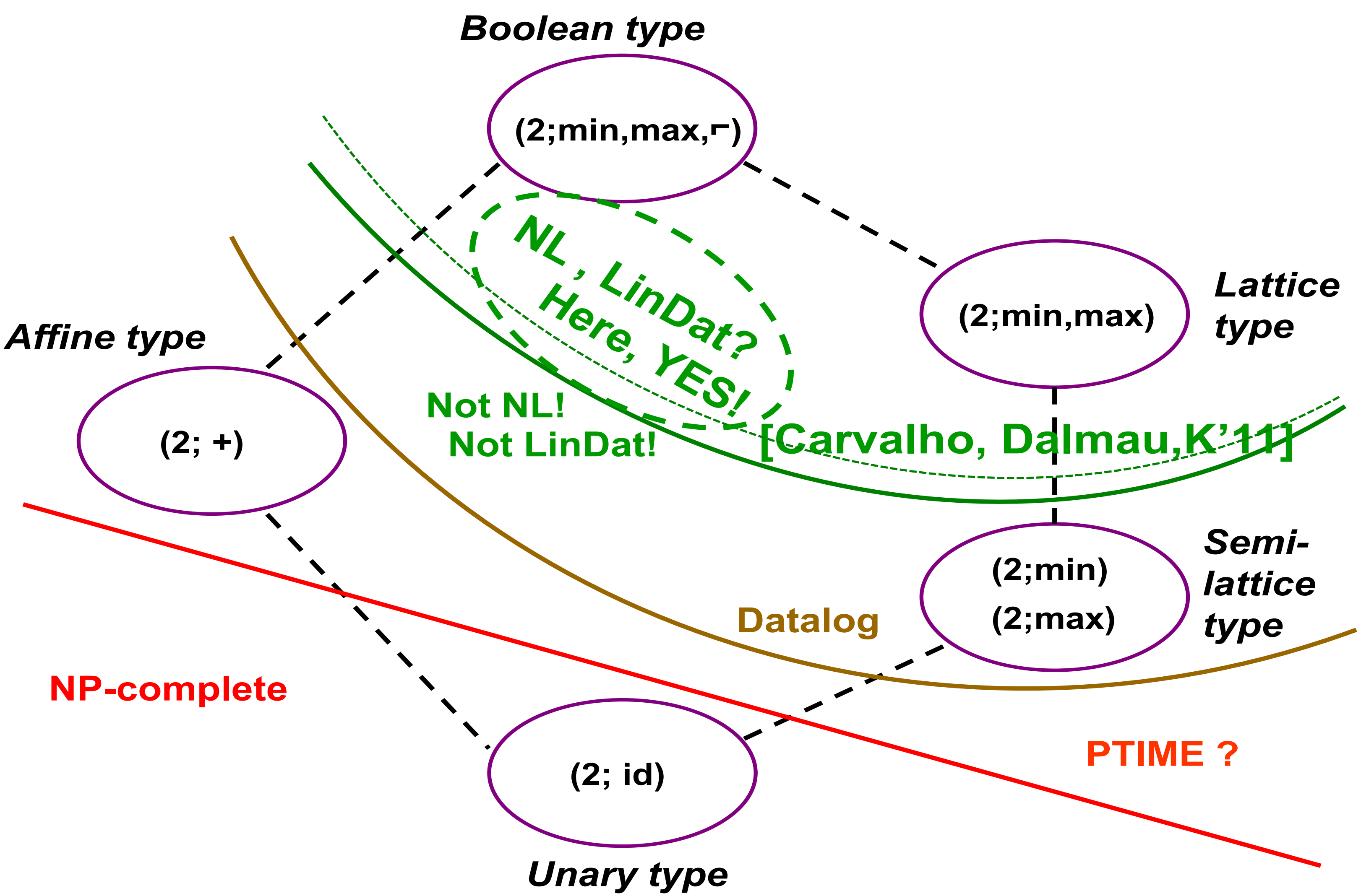
The Linear Datalog/NL Conjecture



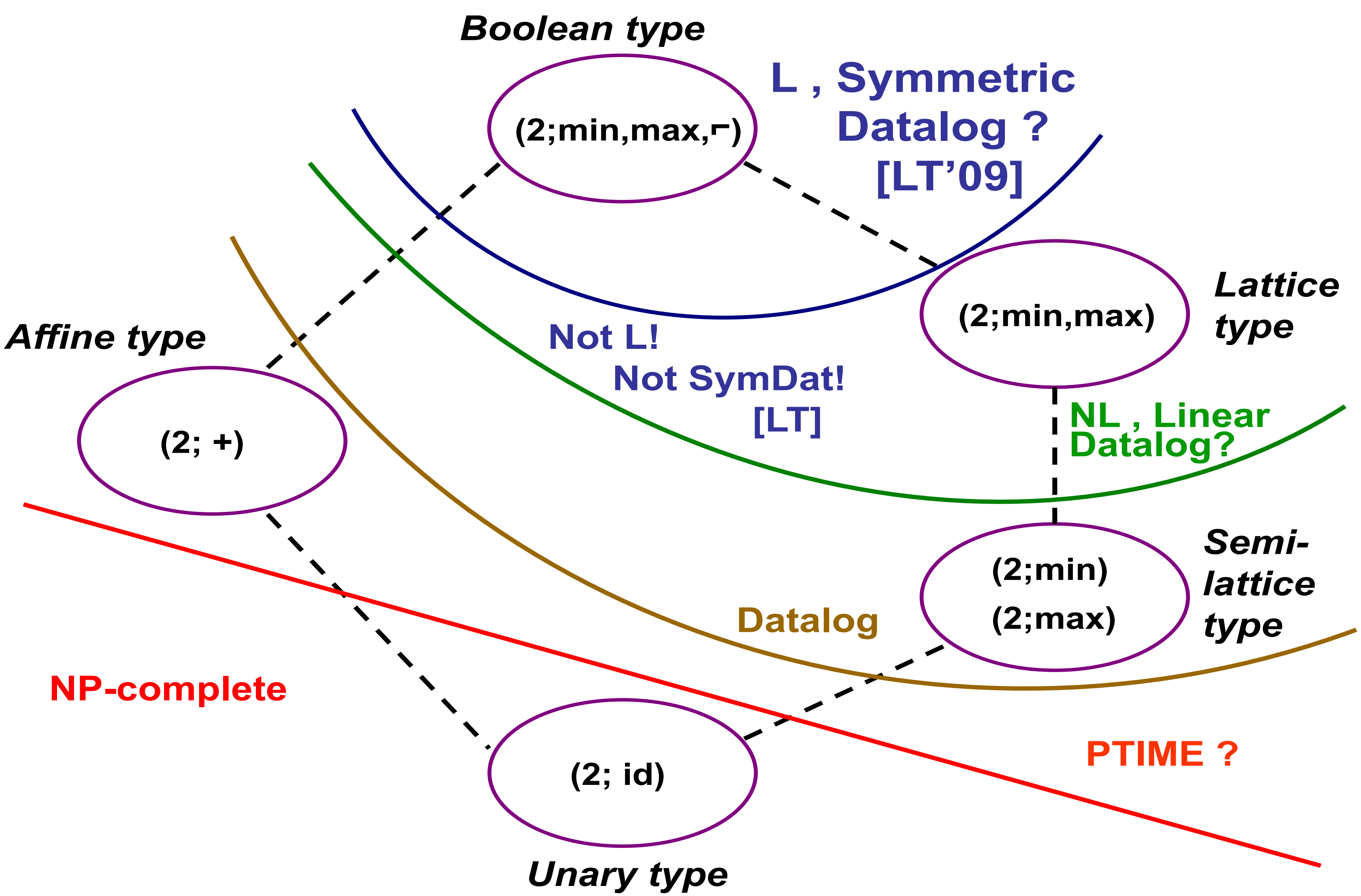
A Linear Datalog/NL Result



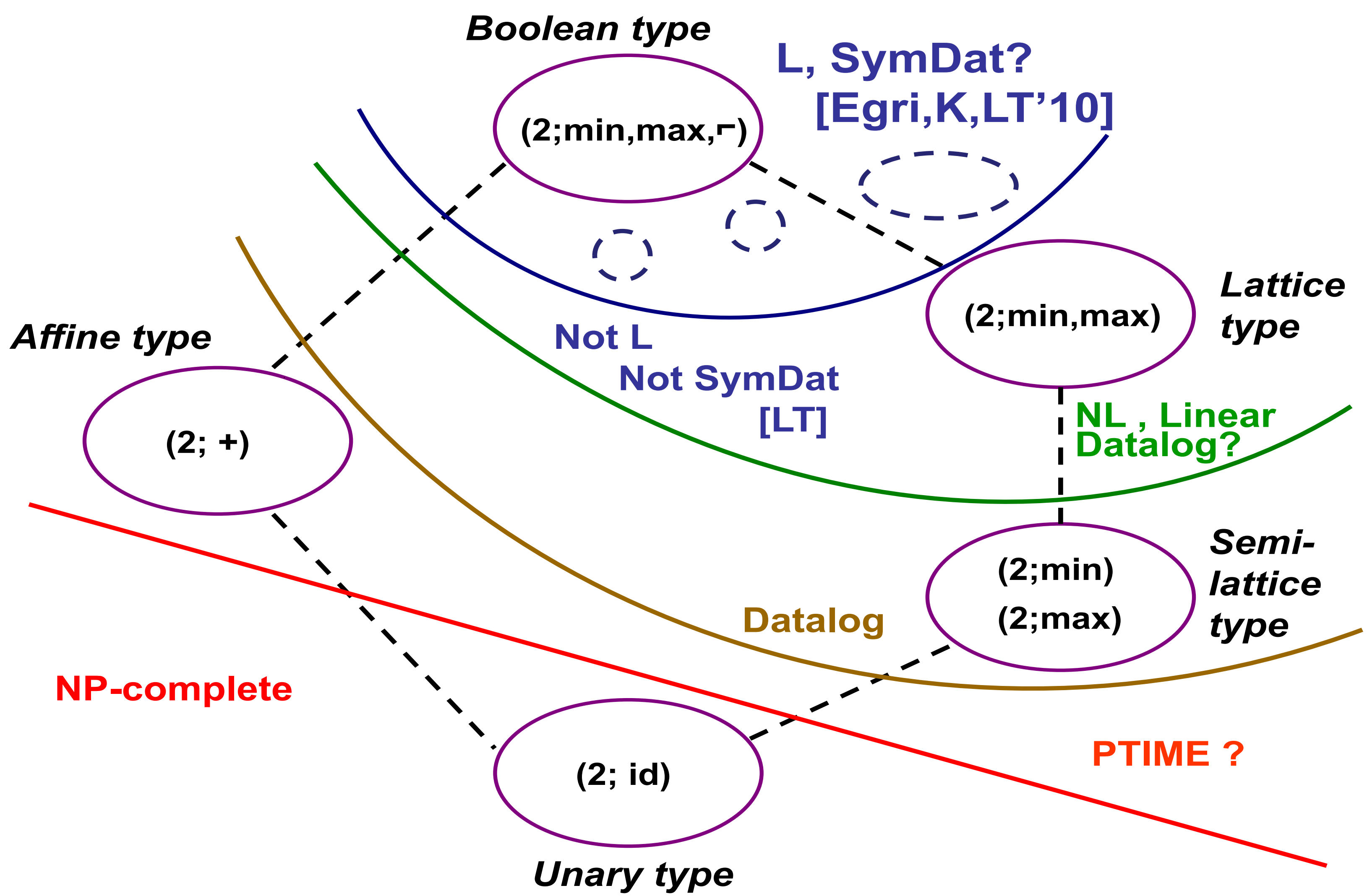
A Linear Datalog/NL Result



The Symmetric Datalog/L Conjecture



A Symmetric Datalog/L Result



A Picture to Take Home