Destructive rule-based properties and first-order logic

David Duris

Équipe de logique mathématique - Paris 7

www.logique.jussieu.fr/∼duris

Main idea

- Let $\phi(x_1, ...x_k)$ be a **first-order** formula.

Main idea

Let $\phi(x_1, ...x_k)$ be a first-order formula.

► For any finite structure $\mathcal{A}$, we remove subsets $\{a_1, ..., a_k\}$ satisfying $\phi(a_1, ..., a_k)$ successively as long as we can.

Main idea

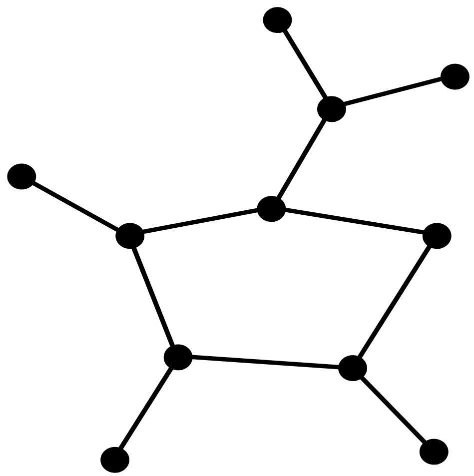
Let $\phi(x_1, ...x_k)$ be a first-order formula.

For any finite structure $\mathcal{A}$, we remove subsets $\{a_1, ..., a_k\}$ satisfying $\phi(a_1, ..., a_k)$ successively as long as we can.

► What are the structures $\mathcal{A}$ such that we **can** obtain the empty structure?

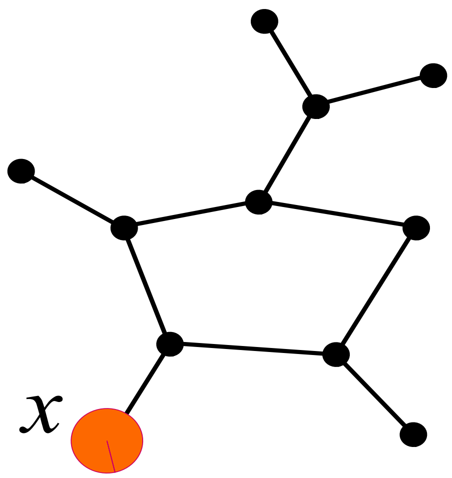
Example 1

$$\begin{aligned}\phi(x) &= \forall u \forall v((Exu \wedge Exv) \Rightarrow u = v) \\ &= \text{“}x \text{ has degree at most 1”}.\end{aligned}$$



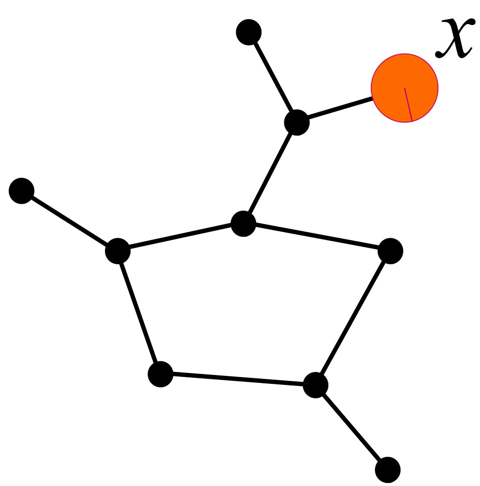
Example 1

$$\begin{aligned}\phi(x) &= \forall u \forall v((Exu \wedge Exv) \Rightarrow u = v) \\ &= \text{“}x\text{ has degree at most 1”}.\end{aligned}$$



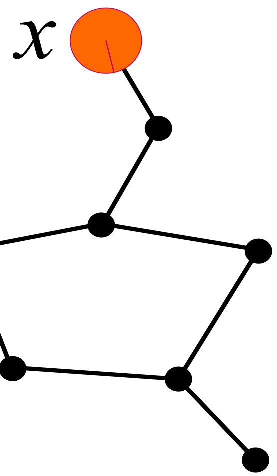
Example 1

$$\begin{aligned}\phi(x) &= \forall u \forall v((Exu \wedge Exv) \Rightarrow u = v) \\ &= \text{“}x\text{ has degree at most 1”}.\end{aligned}$$



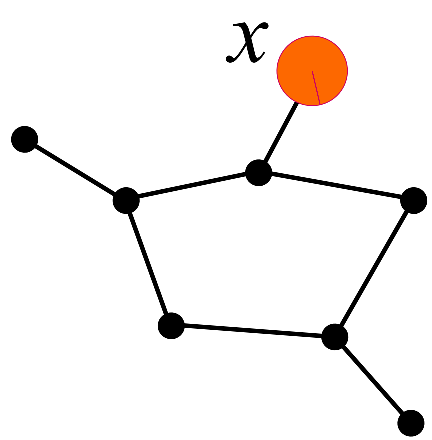
Example 1

$$\begin{aligned}\phi(x) &= \forall u \forall v((Exu \wedge Exv) \Rightarrow u = v)\\ &= \text{“}x\text{ has degree at most 1”}.\end{aligned}$$



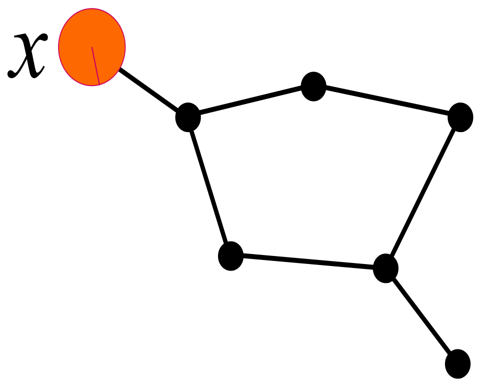
Example 1

$$\begin{aligned}\phi(x) &= \forall u \forall v((Exu \wedge Exv) \Rightarrow u = v) \\ &= \text{“}x \text{ has degree at most 1”.}\end{aligned}$$



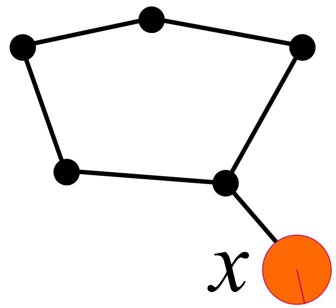
Example 1

$$\begin{aligned}\phi(x) &= \forall u \forall v((Exu \wedge Exv) \Rightarrow u = v) \\ &= \text{“}x\text{ has degree at most 1”}.\end{aligned}$$



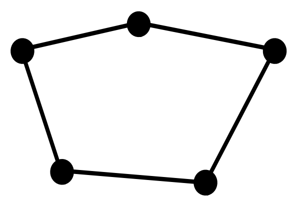
Example 1

$$\begin{aligned} \phi(x) &= \forall u \forall v ((Exu \wedge Exv) \Rightarrow u = v) \\ &= \text{“}x \text{ has degree at most 1”}. \end{aligned}$$



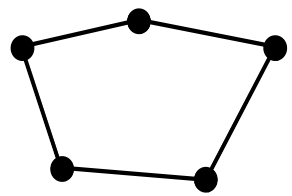
Example 1

$$
\begin{aligned}
\phi(x) &= \forall u \forall v ((Exu \wedge Exv) \Rightarrow u = v) \\
&= \text{“}x\text{ has degree at most 1”.}
\end{aligned}
$$



Example 1

$$
\begin{aligned}
\phi(x) &= \forall u \forall v ((Exu \wedge Exv) \Rightarrow u = v) \\
&= \text{“}x \text{ has degree at most 1”.}
\end{aligned}
$$

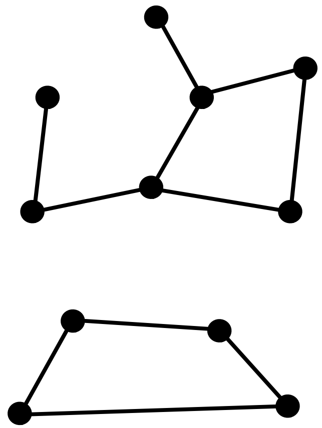


The graphs such that we can obtain the empty structure:

ACYCLIC GRAPHS.

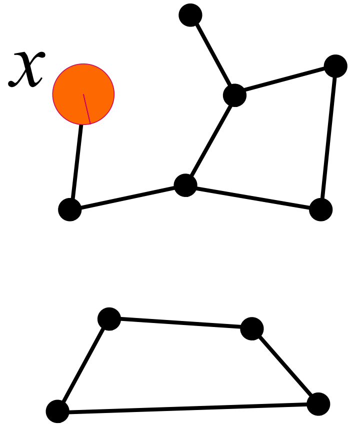
Example 2

$$
\begin{aligned}
\phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\
&= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}.
\end{aligned}
$$



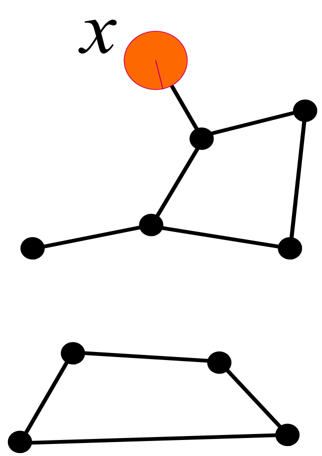
Example 2

$$\begin{aligned}\phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x\text{ is linked to another vertex or }x\text{ is alone”}.\end{aligned}$$



Example 2

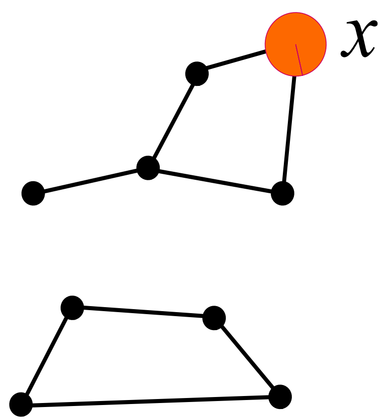
$$\phi(x) = \exists u\ Exu \vee \forall u\ x = u.$$
$$= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}.$$



Example 2

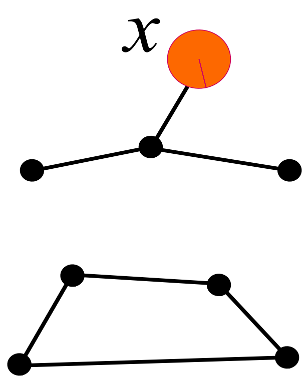
$$\phi(x) = \exists u\; Exu \vee \forall u\; x = u.$$

$$= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}.$$



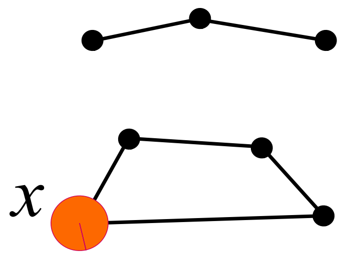
Example 2

$$\begin{aligned}\phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x\text{ is linked to another vertex or }x\text{ is alone”}.\end{aligned}$$



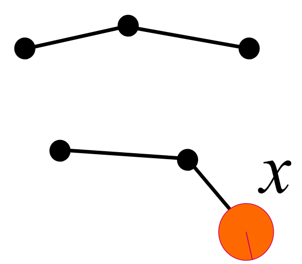
Example 2

$$\begin{aligned} \phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}. \end{aligned}$$



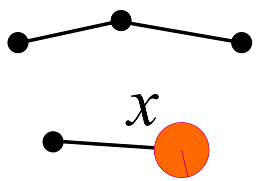
Example 2

$$\begin{aligned}\phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}.\end{aligned}$$



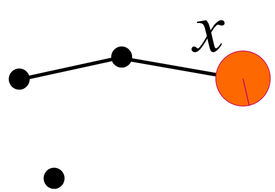
Example 2

$$\begin{aligned} \phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{"}x\text{ is linked to another vertex or }x\text{ is alone".} \end{aligned}$$



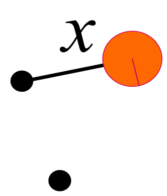
Example 2

$$\begin{aligned} \phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}. \end{aligned}$$



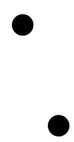
Example 2

$$\begin{aligned}\phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x\text{ is linked to another vertex or }x\text{ is alone”}.\end{aligned}$$



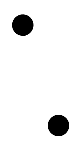
Example 2

$$\begin{aligned}\phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}.\end{aligned}$$



Example 2

$$\begin{aligned} \phi(x) &= \exists u\ Exu \vee \forall u\ x = u. \\ &= \text{“}x \text{ is linked to another vertex or } x \text{ is alone”}. \end{aligned}$$

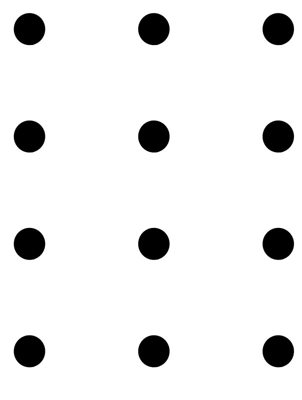


The graphs such that we can obtain the empty structure:

CONNECTED GRAPHS.

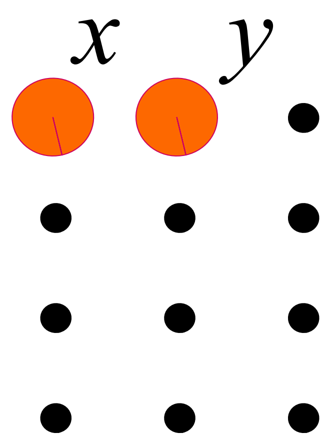
Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$



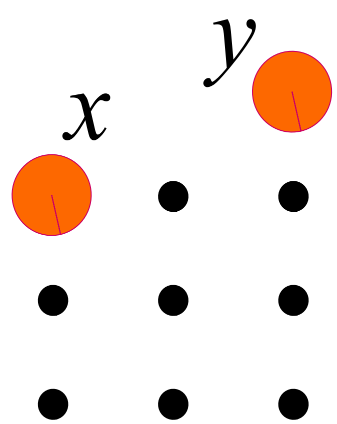
Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$



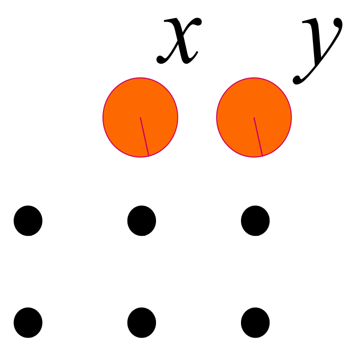
Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$



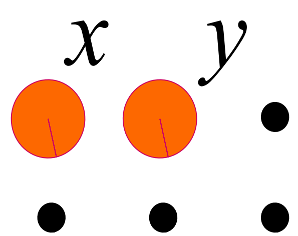
Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$



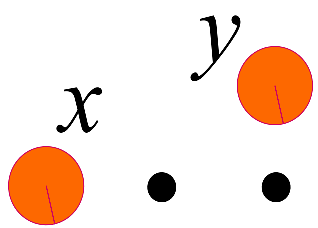
Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$



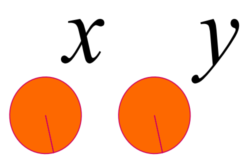
Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$



Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$



Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$

$$\emptyset$$

Example 3

$$\phi(x, y) \quad = \quad x \neq y.$$

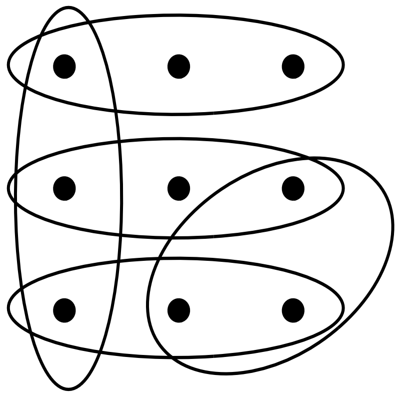
$$\emptyset$$

The structures such that we obtain the empty structure:

STRUCTURES OF EVEN SIZE.

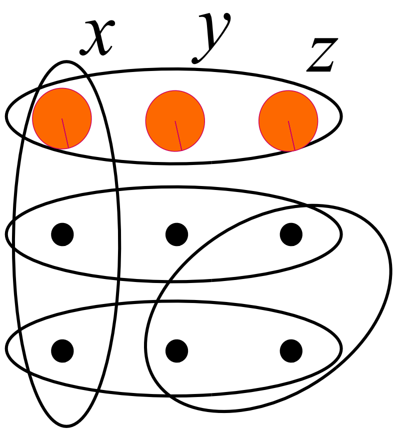
Example 4

$$\phi(x, y, z) \quad = \quad Txyz.$$



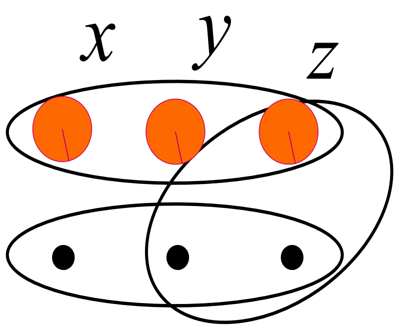
Example 4

$$\phi(x,y,z) \quad = \quad Txyz.$$



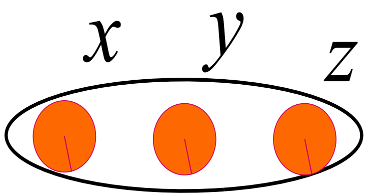
Example 4

$$\phi(x, y, z) \quad = \quad Txyz.$$



Example 4

$$\phi(x, y, z) \quad = \quad Txyz.$$



Example 4

$$\phi(x, y, z) \quad = \quad Txyz.$$

$$\emptyset$$

Example 4

$$\phi(x, y, z) \quad = \quad Txyz.$$

$$\emptyset$$

The collections of 3-sets such that we obtain the empty structure:

those having an EXACT COVER BY 3-SETS (X3C).

Definition

- Let $\phi(x_1, ..., x_k)$ be a first-order formula with k free variables x_1, ..., x_k and let $\mathcal{A}$ be a finite relational structure.

Definition

Let $\phi(x_1, ..., x_k)$ be a first-order formula with k free variables x_1, ..., x_k and let $\mathcal{A}$ be a finite relational structure.

► We consider the following rule: if there exist elements a_1, ..., a_k of A such that $\mathcal{A} \models \phi(a_1, ..., a_k)$ then remove a_1, ..., a_k from $\mathcal{A}$, i.e. replace $\mathcal{A}$ with the substructure $\mathcal{A} \setminus \{a_1, ..., a_k\}$.

Definition

Let $\phi(x_1, ..., x_k)$ be a first-order formula with k free variables x_1, ..., x_k and let $\mathcal{A}$ be a finite relational structure.

We consider the following rule: if there exist elements a_1, ..., a_k of A such that $\mathcal{A} \models \phi(a_1, ..., a_k)$ then remove a_1, ..., a_k from $\mathcal{A}$, i.e. replace $\mathcal{A}$ with the substructure $\mathcal{A} \setminus \{a_1, ..., a_k\}$.

► We call $\mathbf{DR}(\phi(x_1, ..., x_k))$ the set of finite structures $\mathcal{A}$ such that there is a way to apply the rule to $\mathcal{A}$ successively until we obtain the empty structure.

More formally

$\mathcal{A} \in \mathbf{DR}(\phi(x_1, ..., x_k))$ if there exist pairwise disjoint subsets of A $\{a_1^1, ..., a_k^1\}$, ..., $\{a_1^n, ..., a_k^n\}$ such that:

- $\bigcup_{l=1}^{n} \{a_1^l, ..., a_k^l\} = A$, and
- for every $i < n$, $\mathcal{A} \setminus \bigcup_{l=1}^{i} \{a_1^l, ..., a_k^l\} \models \phi(a_1^{i+1}, ..., a_k^{i+1})$.

Summary of the preceding examples

- $\mathbf{DR}(\forall u \forall v((Exu \wedge Exv) \Rightarrow u = v)) = \text{ACYCLIC GRAPHS}$
- $\mathbf{DR}(\exists u\ Exu \vee \forall u\ x = u) = \text{CONNECTED GRAPHS}$
- $\mathbf{DR}(x \neq y) = \text{EVEN}$
- $\mathbf{DR}(Txyz) = \text{X3C}$

Other examples

- $\mathbf{DR}(Exy \wedge Eyx \wedge x \neq y) =$ PERFECT MATCHING
- $\mathbf{DR}(\forall u \forall v((x \in u \wedge x \in v) \Rightarrow \forall t(t \in u \Leftrightarrow t \in v)) \vee$
  $\forall u(\forall t(t \in x \Rightarrow t \in u) \vee \forall t(t \in x \Rightarrow \neg(t \in u))))$
  $=$ γ-ACYCLIC HYPERGRAPHS
- $\mathbf{DR}(\forall u \forall v((x \in u \wedge x \in v) \Rightarrow (\forall t(t \in u \Rightarrow t \in v)$
  $\vee \; \forall t(t \in v \Rightarrow t \in u))) \;\vee\; \forall t \neg(t \in x))$
  $=$ β-ACYCLIC HYPERGRAPHS
- $\mathbf{DR}(\forall u \forall v((x \in u \wedge x \in v) \Rightarrow u = v) \vee \exists w \forall t(t \in x \Rightarrow t \in w))$
  $=$ α-ACYCLIC OF HYPERGRAPHS

Complexity

- Every $\mathbf{DR}(\phi(x_1, ..., x_k))$ is in $\mathbf{NP}$ (data complexity).

Complexity

Every $\mathbf{DR}(\phi(x_1, ..., x_k))$ is in **NP** (data complexity).

- $\mathbf{DR}(Txyz)$ is **NP**-complete.

Complexity

Every $\mathbf{DR}(\phi(x_1, ..., x_k))$ is in **NP** (data complexity).

$\mathbf{DR}(Txyz)$ is **NP**-complete.

▶ What restrictions can ensure polynomial time recognition?

Syntactical restrictions

- Let Q_1, ..., Q_n be quantifier symbols in $\{\forall, \exists, \forall^*, \exists^*\}$.

Syntactical restrictions

Let Q_1, ..., Q_n be quantifier symbols in $\{\forall, \exists, \forall^*, \exists^*\}$.

► We call $Q_1 ... Q_n \mathbf{DR}^k$ the class of properties in the form $\mathbf{DR}(\phi(x_1, ..., x_k))$ where $\phi(x_1, ..., x_k) \in Q_1 ... Q_n \mathbf{FO}$.

Syntactical restrictions

Let Q_1, ..., Q_n be quantifier symbols in $\{\forall, \exists, \forall^*, \exists^*\}$.

We call $Q_1 ... Q_n \mathbf{DR}^k$ the class of properties in the form $\mathbf{DR}(\phi(x_1, ..., x_k))$ where $\phi(x_1, ..., x_k) \in Q_1 ... Q_n \mathbf{FO}$.

Example: GRAPH ACYCLICITY
$(= \mathbf{DR}(\forall u \forall v((Exu \wedge Exv) \Rightarrow u = v)))$ belongs to $\forall\forall\mathbf{DR}^1$.

Influence on the complexity

Classes containing **NP**-complete properties:	Classes included in **PTIME**:
• $\exists\forall\mathbf{DR}^1$ • $\forall\exists\mathbf{DR}^1$ • $\exists\mathbf{DR}^2$ • $\forall\mathbf{DR}^2$ • Quantifier-free $\mathbf{DR}^3$	• $\exists^*\mathbf{DR}^1$ • $\forall^*\mathbf{DR}^1$ • Quantifier-free $\mathbf{DR}^2$

NP-complete cases

Are **NP**-complete:

- $\mathbf{DR}(\exists u \forall v((Eux \wedge ((Euv \wedge Evu) \Rightarrow v = u)) \vee Exx))$
- $\mathbf{DR}(\forall u \exists v(((\neg Evv \wedge (Evx \vee Exv))$
  $\wedge((Euu \wedge Eux) \Rightarrow (Evx \wedge Euv \wedge Evu))$
  $\wedge((Euu \wedge Exu) \Rightarrow (Exv \wedge Euv \wedge Evu)))$
  $\vee(Exv \wedge Evx)))$
- $\mathbf{DR}(\exists t((Exx \wedge Exy \wedge \neg Eyx \wedge Eyt \wedge Ety)$
  $\vee(Ett \wedge x \neq t \wedge Etx \wedge Exy \wedge \neg Eyx)$
  $\vee(Ett \wedge Etx \wedge x \neq t \wedge x = y)$
  $\vee(Exx \wedge Exy \wedge Eyx \wedge x \neq y)))$
- $\mathbf{DR}(\forall u((Exy \wedge \neg(u \neq x \wedge Eux \wedge Exu)) \vee (Exy \wedge x = y)))$
- $\mathbf{DR}(Exy \wedge Eyz \wedge Ezx)$

PTIME cases

Quantifier-free $\mathbf{DR}^2$

$$\mathcal{A} \in \mathbf{DR}(\phi(x, y))?$$

Quantifier-free $\mathbf{DR}^2$

$$\mathcal{A} \in \mathbf{DR}(\phi(x,y))$$

$$\Longleftrightarrow$$

PERFECT MATCHING in the graph

$$G := (A, \{\{a,b\} \mid \mathcal{A} \models \phi(a,b)\}).$$

$\forall^*$**DR**^{1} **and** $\exists^*$**DR**^{1}

Confluent algorithms.

$\forall^*\mathbf{DR}^1$ and preservation under substructure

$\forall^*\mathbf{DR}^1$ and preservation under substructure

- Preservation under substructure:

If $\mathcal{A} \in \mathcal{P}$ and $\mathcal{B} \subset \mathcal{A}$, then $\mathcal{B} \in \mathcal{P}$.

$\forall^*$**DR**^{1} and preservation under substructure

Preservation under substructure:
If $\mathcal{A} \in \mathcal{P}$ and $\mathcal{B} \subset \mathcal{A}$, then $\mathcal{B} \in \mathcal{P}$.

- Failure of the preservation theorem:

$$\forall^*\mathbf{FO} \subsetneq \text{preserved under substructure } \mathbf{FO}.$$

$\forall^*$**DR**^{1} and preservation under substructure

Preservation under substructure:
If $\mathcal{A} \in \mathcal{P}$ and $\mathcal{B} \subset \mathcal{A}$, then $\mathcal{B} \in \mathcal{P}$.

- Refinement of the failure:

$$\forall^*\mathbf{FO} \subsetneq \forall^*\mathbf{DR}^1 \cap \mathbf{FO} \subsetneq \text{preserved under substructure } \mathbf{FO}.$$

Undefinability example

GRAPH PLANARITY $\notin$ **DR**.

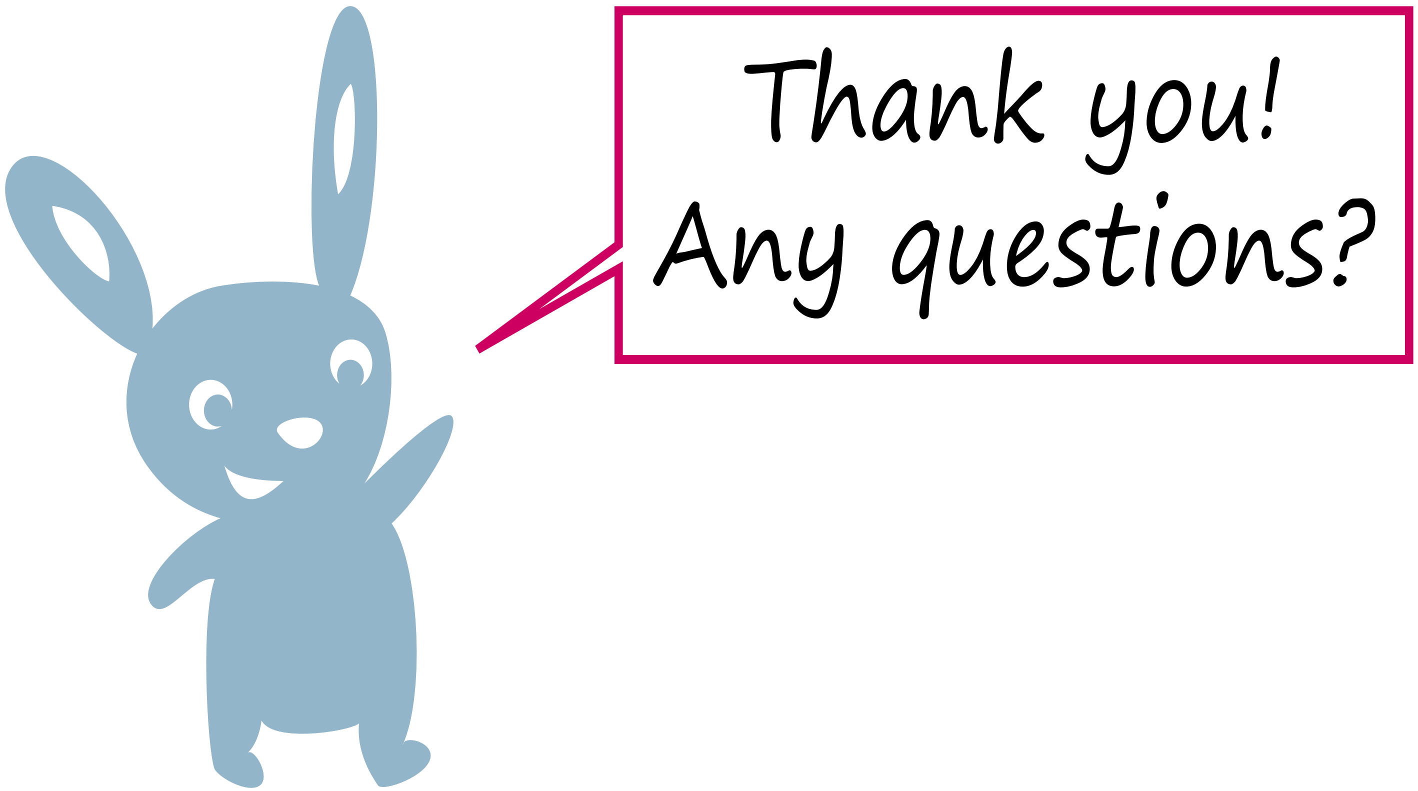
Objectives

- Capturing more **PTIME** properties by finding other conditions on the formula.

Objectives

- Capturing more **PTIME** properties by finding other conditions on the formula.

- Complexity classification for special digraphs: simple graphs, digraphs representing a hypergraph (i.e. bipartite digraphs of the signature $\{\in\}$).


Thank you!
Any questions?