TECHNISCHE UNIVERSITÄT DRESDEN

DATABASE THEORY

Lecture 11: Query Expressiveness

Markus Krötzsch

Knowledge-Based Systems

TU Dresden, 15th May 2019

Review

First-Order Query Expressiveness

Queries and Their Expressiveness

Recall:

- Syntax: a query expression q is a word from a query language (algebra expression, logical expression, etc.)
- Semantics: a query mapping $M[q]$ is a function that maps a database instance $\mathcal{I}$ to a database table $M[q](\mathcal{I})$
- We only study generic queries, which are closed under bijective renaming (isomorphism of databases)

Definition 11.1: The expressiveness of a query language is characterised by the set of query mappings that it can express.
Given a query language **L**, a query mapping M is **L**-definable if there is a query expression $q \in$ **L** such that $M[q] = M$.

We can study expressiveness for all query mappings over all possible databases, or we can restrict attention to a subset of query mappings or to a subset of databases.

Boolean Query Mappings

A Boolean query mapping is a query mapping that returns “true” (usually a database with one table with one empty row) or “false” (usually an empty database).

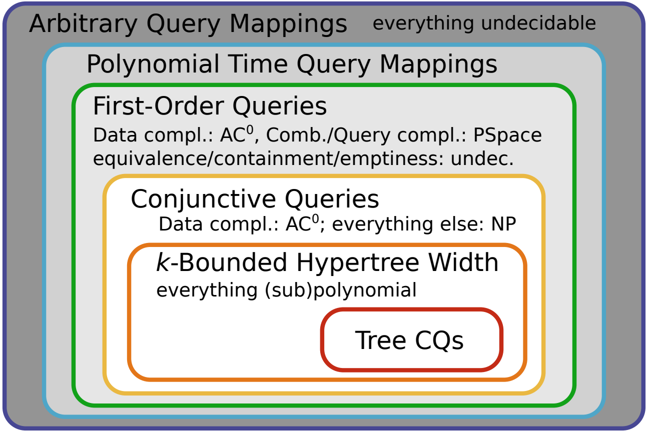
Every Boolean query mapping

- defines set of databases for which it is true
- defines a decision problem over the set of all databases
- could be decidable or undecidable
- if decidable, it may be characterised in terms of complexity

  Note: the “complexity of a mapping” is always “data complexity,” i.e., complexity w.r.t. the size of the input database; the mapping defines the decision problem and is fixed.

Expressivity vs. Complexity

All query mappings that can be expressed in first-order logic are of polynomial complexity, actually in AC^0.



The Limits of FO Queries

Are there polynomial query mappings that cannot be expressed in FO?

The Limits of FO Queries

Are there polynomial query mappings that cannot be expressed in FO?
$\leadsto$ yes!

We already knew this from previous lectures:

- We learned that $\mathsf{AC}^0 \subset \mathsf{NC}^1 \subseteq \ldots \subseteq \mathsf{P}$
- Hence, there is a problem X in NC^1 that is not in AC^0
- Therefore, the corresponding query mapping M_X is not FO-definable

$\mathsf{AC}^0 \subset \mathsf{NC}^1$ was first shown for the problem $X = \textsc{Parity}$:

- **Input:** finite relational structure $\mathcal{I}$
- **Output:** “true” if $\mathcal{I}$ has an even number of domain elements

The original proof is specific to this problem [Ajtai 1983].

Any Other FO-Undefinable Problems?

Any Other FO-Undefinable Problems?

Yes, many.

Any Other FO-Undefinable Problems?

Yes, many.

Strong evidence from complexity theory:

- If any P-complete problem X were FO-definable,

Any Other FO-Undefinable Problems?

Yes, many.

Strong evidence from complexity theory:

- If any P-complete problem X were FO-definable,
- then every problem in P could be LogSpace-reduced to X

Any Other FO-Undefinable Problems?

Yes, many.

Strong evidence from complexity theory:

- If any P-complete problem X were FO-definable,
- then every problem in P could be LogSpace-reduced to X
- and then solved in AC^0,

Any Other FO-Undefinable Problems?

Yes, many.

Strong evidence from complexity theory:

- If any P-complete problem X were FO-definable,
- then every problem in P could be LogSpace-reduced to X
- and then solved in AC^0,
- hence every problem in P could be solved in LogSpace,

Any Other FO-Undefinable Problems?

Yes, many.

Strong evidence from complexity theory:

- If any P-complete problem X were FO-definable,
- then every problem in P could be LogSpace-reduced to X
- and then solved in AC^0,
- hence every problem in P could be solved in LogSpace,
- that is, P $=$ L.
- Most experts do not think that this is the case.

Therefore, one would expect all P-hard and similarly all NL-hard problems to not be FO-definable.

$\leadsto$ How can we see this more directly?

Proving FO-Undefinability

How to show that a query mapping is FO-definable?

Proving FO-Undefinability

How to show that a query mapping is FO-definable?

$\leadsto$ Find an FO query that expresses the query mapping

Proving FO-Undefinability

How to show that a query mapping is FO-definable?
$\leadsto$ Find an FO query that expresses the query mapping

How to show that a query mapping is **not** FO-definable?

Proving FO-Undefinability

How to show that a query mapping is FO-definable?

$\rightsquigarrow$ Find an FO query that expresses the query mapping

How to show that a query mapping is **not** FO-definable?

$\rightsquigarrow$ Not so easy . . . important tools:

- Ehrenfeucht-Fraïssé games
- Locality theorems

Ehrenfeucht-Fraïssé Games

A method for showing that certain finite structures cannot be distinguished by certain FO formulas

General idea:

- A game is played on two databases $\mathcal{I}$ and $\mathcal{J}$
- There are two players: the Spoiler and the Duplicator
- The players select elements from $\mathcal{I}$ and $\mathcal{J}$ in each round
- Spoiler wants to show that the two databases are different
- Duplicator wants make the databases appear to be the same

We will always play on finite structures without constant symbols

(remember that one can simulate constants by unary relations with one row)

Playing One Run of an EF Game

A single run of the game has a fixed number r of rounds

Spoiler starts each round, and Duplicator answers:

- Spoiler picks a domain element from $\mathcal{I}$ or from $\mathcal{J}$
- Duplicator picks an element from the other database ($\mathcal{J}$ or $\mathcal{I}$)

$\leadsto$ One element gets picked from each $\mathcal{I}$ and $\mathcal{J}$ per round
$\leadsto$ Run of game ends with two lists of elements:
$a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$

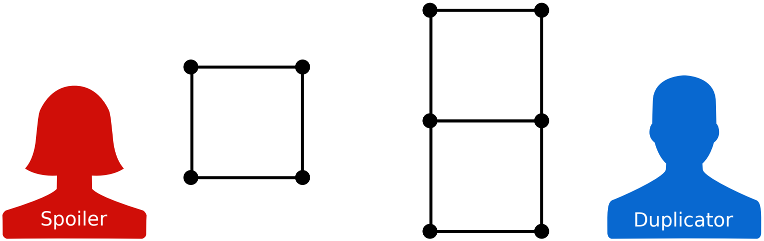
Duplicator wins the run if:

- For all indices i and j, we have $a_i = a_j$ if and only if $b_i = b_j$.
- For all lists of indices $i_1, \ldots, i_n$ and n-ary relation names R, we have $\langle a_{i_1}, \ldots, a_{i_n} \rangle \in R^{\mathcal{I}}$ if and only if $\langle b_{i_1}, \ldots, b_{i_n} \rangle \in R^{\mathcal{J}}$.

  "The substructures induced by the selected elements are isomorphic"

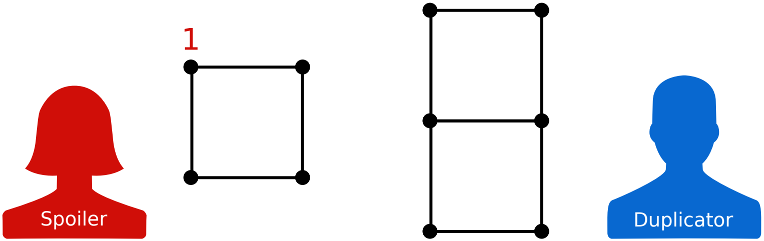
Otherwise Spoiler wins the run.

Example: Run of a Two-Turn EF Game



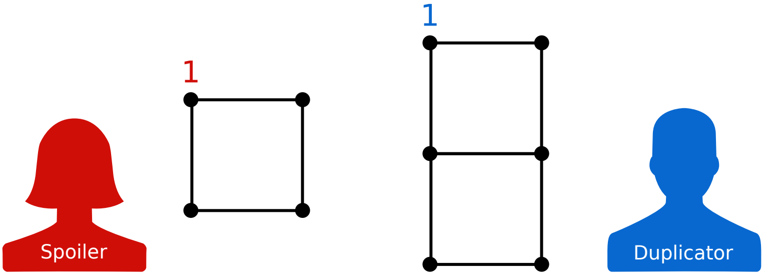
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Two-Turn EF Game



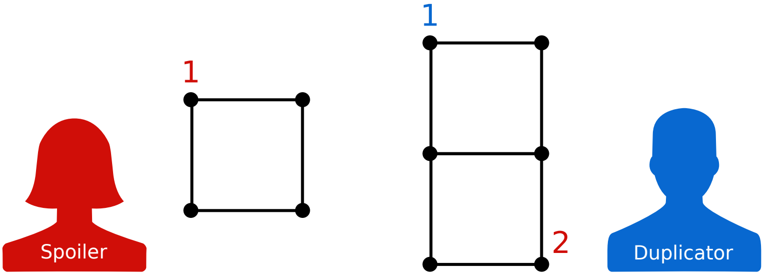
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Two-Turn EF Game



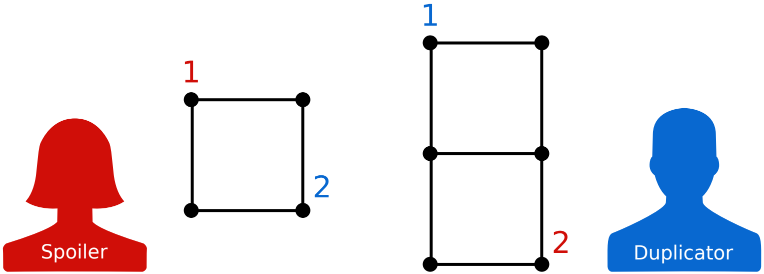
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Two-Turn EF Game



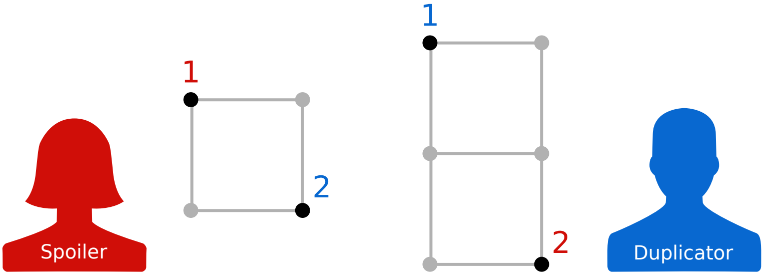
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Two-Turn EF Game



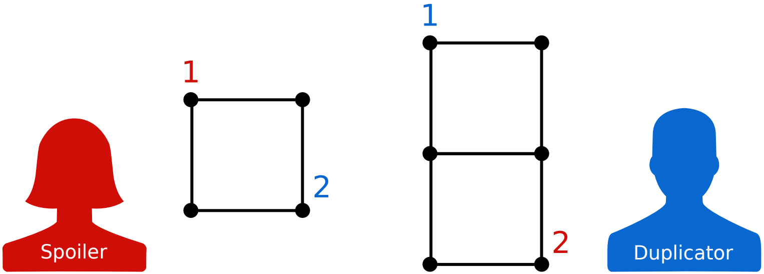
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Two-Turn EF Game



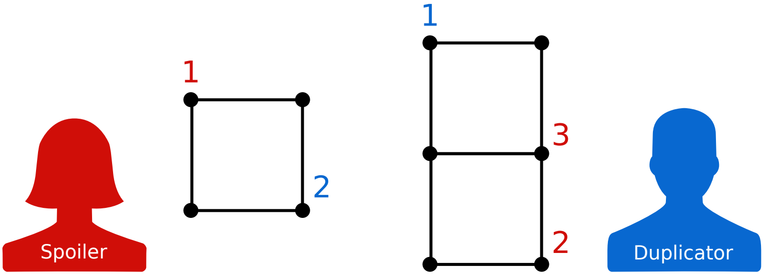
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Three-Turn EF Game



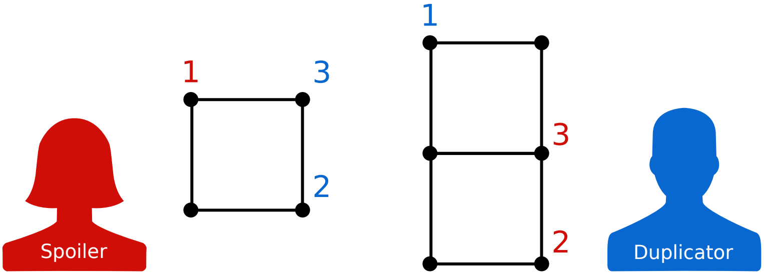
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Three-Turn EF Game



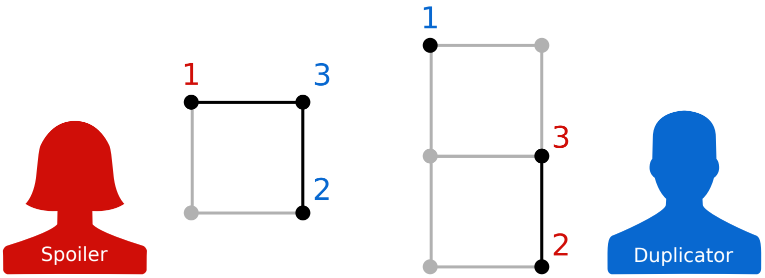
- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Three-Turn EF Game



- edges denote a bi-directional binary predicate
- all edges are the same predicate

Example: Run of a Three-Turn EF Game



- edges denote a bi-directional binary predicate
- all edges are the same predicate

Winning the EF Game

The game is won by whoever has a winning strategy:

A player has a winning strategy if he/she can make sure that he/she will win, whatever the other player is doing.

In other words:

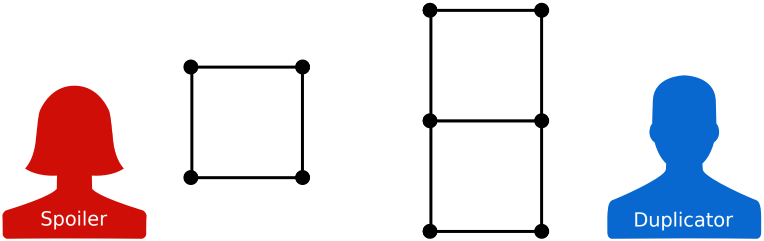
- Duplicator wins if he can duplicate any move that the spoiler makes.
- Spoiler wins if she can spoil any attempt to duplicate her moves.

We write $\mathcal{I} \sim_r \mathcal{J}$ if Duplicator wins the r-round EF game on $\mathcal{I}$ and $\mathcal{J}$.

Observation: given enough moves, the spoiler will always win, unless the structures are isomorphic

Example

Who wins the 2-round game?
Who wins the 3-round game?



- edges denote a bi-directional binary predicate
- all edges are the same predicate

Quantifier Rank

EF games characterise expressivity of FO formulae based on the nesting depth of quantifiers:

Definition 11.2: The quantifier rank of a FO formula is the maximal nesting level of quantifiers within the formula.

Example 11.3:

- A formula without quantifiers has quantifier rank 0
- $\exists x.(C(x) \wedge \forall y.(R(x, y) \to x \approx y) \wedge \exists v.S(x, v))$ has quantifier rank 2

Quantifier Rank

EF games characterise expressivity of FO formulae based on the nesting depth of quantifiers:

Definition 11.2: The quantifier rank of a FO formula is the maximal nesting level of quantifiers within the formula.

Example 11.3:

- A formula without quantifiers has quantifier rank 0
- $\exists x.(C(x) \wedge \forall y.(R(x, y) \rightarrow x \approx y) \wedge \exists v.S(x, v))$ has quantifier rank 2

Definition 11.4: We write $\mathcal{I} \equiv_r \mathcal{J}$ if $\mathcal{I}$ and $\mathcal{J}$ satisfy the same FO sentences of rank r (or less).

Significance of EF Games

Theorem 11.5: For every r, $\mathcal{I}$ and $\mathcal{J}$, the following are equivalent:

- $\mathcal{I} \equiv_r \mathcal{J}$, that is, $\mathcal{I}$ and $\mathcal{J}$ satisfy the same FO sentences of rank r (or less).
- $\mathcal{I} \sim_r \mathcal{J}$, that is, the Duplicator wins the r-round EF game on $\mathcal{I}$ and $\mathcal{J}$.

Therefore, the following are equivalent:

- The query mapping M is FO-definable
- There is an FO sentence φ that defines M
- There is a number r such that, for every $\mathcal{I}$ accepted by M and every $\mathcal{J}$ not accepted by M, the Spoiler wins the r-round EF game on $\mathcal{I}$ and $\mathcal{J}$

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \nsim_r \mathcal{J}$.

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \nsim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \nsim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

We sketch the idea for the case that φ_r is in prenex normal form $\varphi_r = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_r.x_r.\psi$ with $\mathsf{Q}_i \in \{\exists, \forall\}$ and ψ a quantifier-free formula

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \not\sim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

We sketch the idea for the case that φ_r is in prenex normal form $\varphi_r = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_r.x_r.\psi$ with $\mathsf{Q}_i \in \{\exists, \forall\}$ and ψ a quantifier-free formula:

- Then $\neg\varphi_r$ is equivalent to $\bar{\mathsf{Q}}_1 x_1 \ldots \bar{\mathsf{Q}}_r.x_r.\neg\psi$, where $\bar{\exists} = \forall$ and $\bar{\forall} = \exists$

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \not\sim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

We sketch the idea for the case that φ_r is in prenex normal form $\varphi_r = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_r.x_r.\psi$ with $\mathsf{Q}_i \in \{\exists, \forall\}$ and ψ a quantifier-free formula:

- Then $\neg\varphi_r$ is equivalent to $\bar{\mathsf{Q}}_1 x_1 \ldots \bar{\mathsf{Q}}_r.x_r.\neg\psi$, where $\bar{\exists} = \forall$ and $\bar{\forall} = \exists$
- Spoiler will enforce a selection of elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that, after i steps of the game, $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_i \mapsto a_i\} \models \mathsf{Q}_{i+1} x_{i+1} \ldots \mathsf{Q}_r x_r.\psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_i \mapsto b_i\} \not\models \mathsf{Q}_{i+1} x_{i+1} \cdots \mathsf{Q}_r x_r.\psi$ $(*)$

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \not\sim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

We sketch the idea for the case that φ_r is in prenex normal form $\varphi_r = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_r.x_r.\psi$ with $\mathsf{Q}_i \in \{\exists, \forall\}$ and ψ a quantifier-free formula:

- Then $\neg\varphi_r$ is equivalent to $\bar{\mathsf{Q}}_1 x_1 \ldots \bar{\mathsf{Q}}_r.x_r.\neg\psi$, where $\bar{\exists} = \forall$ and $\bar{\forall} = \exists$
- Spoiler will enforce a selection of elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that, after i steps of the game, $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_i \mapsto a_i\} \models \mathsf{Q}_{i+1} x_{i+1} \ldots \mathsf{Q}_r x_r.\psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_i \mapsto b_i\} \not\models \mathsf{Q}_{i+1} x_{i+1}. \cdots \mathsf{Q}_r x_r.\psi$ $(*)$:
  - Property $(*)$ holds initially $(i = 0)$ by assumption.

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \not\sim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

We sketch the idea for the case that φ_r is in prenex normal form $\varphi_r = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_r.x_r.\psi$ with $\mathsf{Q}_i \in \{\exists, \forall\}$ and ψ a quantifier-free formula:

- Then $\neg\varphi_r$ is equivalent to $\bar{\mathsf{Q}}_1 x_1 \ldots \bar{\mathsf{Q}}_r.x_r.\neg\psi$, where $\bar{\exists} = \forall$ and $\bar{\forall} = \exists$
- Spoiler will enforce a selection of elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that, after i steps of the game, $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_i \mapsto a_i\} \models \mathsf{Q}_{i+1} x_{i+1} \ldots \mathsf{Q}_r x_r.\psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_i \mapsto b_i\} \not\models \mathsf{Q}_{i+1} x_{i+1} \cdots \mathsf{Q}_r x_r.\psi$ $(*)$:
  - Property $(*)$ holds initially $(i = 0)$ by assumption.
  - In step $i+1$, if $\mathsf{Q}_{i+1} = \exists$, Spoiler selects $a_{i+1} \in \Delta^{\mathcal{I}}$ such that $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_{i+1} \mapsto a_{i+1}\} \models \mathsf{Q}_{i+2} x_{i+2} \ldots \mathsf{Q}_r x_r.\psi$ – this exists because of $(*)$.

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \nsim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

We sketch the idea for the case that φ_r is in prenex normal form $\varphi_r = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_r.x_r.\psi$ with $\mathsf{Q}_i \in \{\exists, \forall\}$ and ψ a quantifier-free formula:

- Then $\neg\varphi_r$ is equivalent to $\bar{\mathsf{Q}}_1 x_1 \ldots \bar{\mathsf{Q}}_r.x_r.\neg\psi$, where $\bar{\exists} = \forall$ and $\bar{\forall} = \exists$
- Spoiler will enforce a selection of elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that, after i steps of the game, $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_i \mapsto a_i\} \models \mathsf{Q}_{i+1} x_{i+1} \ldots \mathsf{Q}_r x_r.\psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_i \mapsto b_i\} \not\models \mathsf{Q}_{i+1} x_{i+1} \cdots \mathsf{Q}_r x_r.\psi$ $(*)$:
  - Property $(*)$ holds initially $(i = 0)$ by assumption.
  - In step $i+1$, if $\mathsf{Q}_{i+1} = \exists$, Spoiler selects $a_{i+1} \in \Delta^{\mathcal{I}}$ such that $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_{i+1} \mapsto a_{i+1}\} \models \mathsf{Q}_{i+2} x_{i+2} \ldots \mathsf{Q}_r x_r.\psi$ – this exists because of $(*)$.
  - Any choice b_{i+1} of Duplicator will be such that $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_{i+1} \mapsto b_{i+1}\} \not\models \mathsf{Q}_{i+2} x_{i+2} \ldots \mathsf{Q}_r x_r.\psi$, since $\bar{\mathsf{Q}}_{i+1} = \forall$.

Proof idea (1)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof: We show the contrapositive: if $\mathcal{I} \not\equiv_r \mathcal{J}$ then $\mathcal{I} \not\sim_r \mathcal{J}$. Hence, suppose there is a formula φ_r of quantifier depth r such that (w.l.o.g.) $\mathcal{I} \models \varphi_r$ and $\mathcal{J} \models \neg\varphi_r$.

We sketch the idea for the case that φ_r is in prenex normal form $\varphi_r = \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_r.x_r.\psi$ with $\mathsf{Q}_i \in \{\exists, \forall\}$ and ψ a quantifier-free formula:

- Then $\neg\varphi_r$ is equivalent to $\bar{\mathsf{Q}}_1 x_1 \ldots \bar{\mathsf{Q}}_r.x_r.\neg\psi$, where $\bar{\exists} = \forall$ and $\bar{\forall} = \exists$
- Spoiler will enforce a selection of elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that, after i steps of the game, $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_i \mapsto a_i\} \models \mathsf{Q}_{i+1} x_{i+1} \ldots \mathsf{Q}_r x_r.\psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_i \mapsto b_i\} \not\models \mathsf{Q}_{i+1} x_{i+1} \cdots \mathsf{Q}_r x_r.\psi$ $(*)$:
  - Property $(*)$ holds initially $(i = 0)$ by assumption.
  - In step $i + 1$, if $\mathsf{Q}_{i+1} = \exists$, Spoiler selects $a_{i+1} \in \Delta^{\mathcal{I}}$ such that $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_{i+1} \mapsto a_{i+1}\} \models \mathsf{Q}_{i+2} x_{i+2} \ldots \mathsf{Q}_r x_r.\psi$ – this exists because of $(*)$.
  - Any choice b_{i+1} of Duplicator will be such that $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_{i+1} \mapsto b_{i+1}\} \not\models \mathsf{Q}_{i+2} x_{i+2} \ldots \mathsf{Q}_r x_r.\psi$, since $\bar{\mathsf{Q}}_{i+1} = \forall$.
  - The case $\mathsf{Q}_{i+1} = \forall$ is similar: now Spoiler selects b_{i+1}.

Proof idea (2)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof (continued): Therefore, by $(*)$, after r rounds we have selected elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_r \mapsto a_r\} \models \psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_r \mapsto b_r\} \not\models \psi$.

Proof idea (2)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof (continued): Therefore, by $(*)$, after r rounds we have selected elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_r \mapsto a_r\} \models \psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_r \mapsto b_r\} \not\models \psi$.

Hence, the substructures induced by the selected elements are not isomorphic (if they were, we would find that ψ evaluates to the same in both cases)

Proof idea (2)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof (continued): Therefore, by $(*)$, after r rounds we have selected elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_r \mapsto a_r\} \models \psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_r \mapsto b_r\} \not\models \psi$.

Hence, the substructures induced by the selected elements are not isomorphic (if they were, we would find that ψ evaluates to the same in both cases)

$\leadsto$ Spoiler wins

Proof idea (2)

We outline the proof for the direction that is more important to us:

Lemma 11.6: For every r, we find $\sim_r \subseteq \equiv_r$.

Proof (continued): Therefore, by $(*)$, after r rounds we have selected elements $a_1, \ldots, a_r \in \Delta^{\mathcal{I}}$ and $b_1, \ldots, b_r \in \Delta^{\mathcal{J}}$, such that $\mathcal{I}, \{x_1 \mapsto a_1, \ldots, x_r \mapsto a_r\} \models \psi$ and $\mathcal{J}, \{x_1 \mapsto b_1, \ldots, x_r \mapsto b_r\} \not\models \psi$.

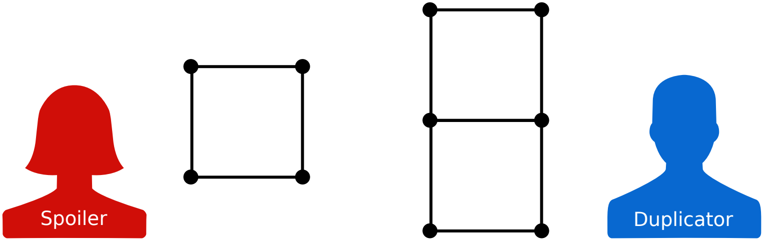
Hence, the substructures induced by the selected elements are not isomorphic (if they were, we would find that ψ evaluates to the same in both cases)

$\leadsto$ Spoiler wins

The idea can be generalised to formulae φ_r that are not in prenex normal form (by interleaving the choice of the quantifier and the evaluation of the formula) $\square$

Example

Let's assume all edges denote the (bi-directional) predicate r:

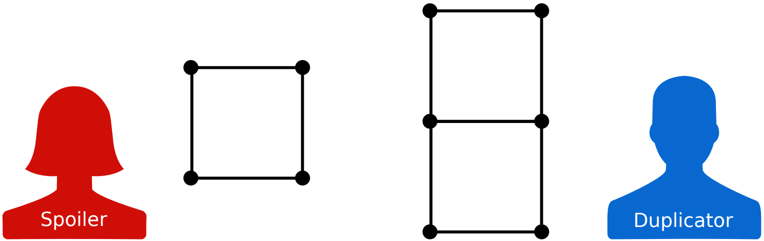


$\mathcal{I}$ $\mathcal{J}$

Which formula distinguishes the two structures?

Example

Let's assume all edges denote the (bi-directional) predicate r:



$\mathcal{I}$ $\mathcal{J}$

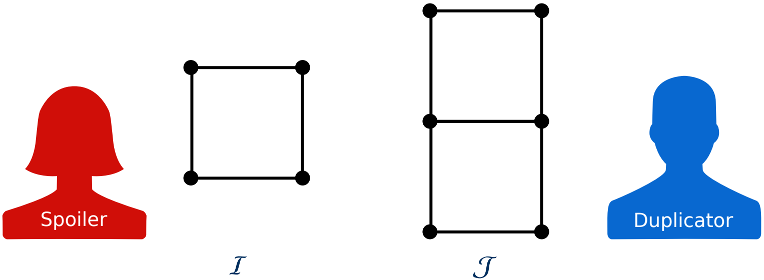
Which formula distinguishes the two structures?

For example: $\varphi_3 = \exists x.\exists y.\forall z.r(x,z) \leftrightarrow r(y,z)$

- $\mathcal{I} \models \varphi_3$
- $\mathcal{J} \not\models \varphi_3$

Example

Let's assume all edges denote the (bi-directional) predicate r:



Which formula distinguishes the two structures?

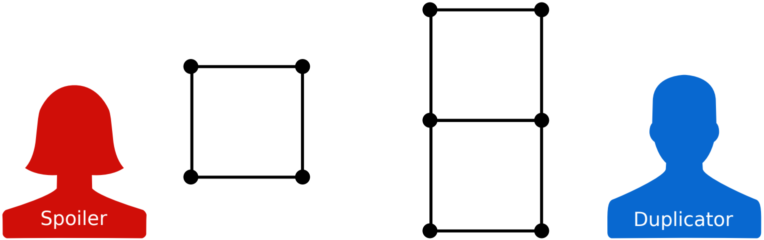
For example: $\varphi_3 = \exists x.\exists y.\forall z.r(x,z) \leftrightarrow r(y,z)$

- $\mathcal{I} \models \varphi_3$
- $\mathcal{J} \not\models \varphi_3$

The formula corresponds to 3-move a winning strategy for Spoiler

Example

Let's assume all edges denote the (bi-directional) predicate r:



$\mathcal{I}$ $\mathcal{J}$

Which formula distinguishes the two structures?

For example: $\varphi_3 = \exists x.\exists y.\forall z.r(x, z) \leftrightarrow r(y, z)$

- $\mathcal{I} \models \varphi_3$
- $\mathcal{J} \not\models \varphi_3$

The formula corresponds to 3-move a winning strategy for Spoiler:

- first select opposing corners in $\mathcal{I}$
- then select an element in $\mathcal{J}$ that neighbours exactly one of the elements selected by Duplicator

Using EF Games to Show FO-Undefinability

How to show that a query mapping M can not be FO-defined:

- Let C_M be the class of all databases recognised by M
- Find sequences of databases $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \ldots \in C_M$ and databases $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3, \ldots \notin C_M$, such that $\mathcal{I}_i \sim_i \mathcal{J}_i$

$\rightsquigarrow$ for any formula φ (however large its quantifier rank r), there is a counterexample $\mathcal{I}_r \in C_M$ and $\mathcal{J}_r \notin C_M$ that φ cannot distinguish

Using EF Games to Show FO-Undefinability

How to show that a query mapping M can not be FO-defined:

- Let C_M be the class of all databases recognised by M
- Find sequences of databases $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \ldots \in C_M$ and databases $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3, \ldots \notin C_M$, such that $\mathcal{I}_i \sim_i \mathcal{J}_i$

$\leadsto$ for any formula φ (however large its quantifier rank r), there is a counterexample $\mathcal{I}_r \in C_M$ and $\mathcal{J}_r \notin C_M$ that φ cannot distinguish

Problems:

- How to find such sequences of $\mathcal{I}_i$ and $\mathcal{J}_i$?
  $\leadsto$ No general strategy exists
- Given suitable sequences, how to show that $\mathcal{I}_i \sim_i \mathcal{J}_i$?
  $\leadsto$ Can be difficult, but doable for some special cases

Expressiveness on Linear Orders

Let's look at some very simple structures:

Definition 11.7: A structure $\mathcal{I}$ is a linear order if it has a single binary predicate $\leq$ interpreted as a total, transitive, reflexive and asymmetric relation.

Expressiveness on Linear Orders

Let's look at some very simple structures:

Definition 11.7: A structure $\mathcal{I}$ is a linear order if it has a single binary predicate $\leq$ interpreted as a total, transitive, reflexive and asymmetric relation.

Example 11.8: Consider the following structures:

$\mathcal{L}_6 : 1 \leq 2 \leq 3 \leq 4 \leq 5 \leq 6$
$\mathcal{L}_7 : 1 \leq 2 \leq 3 \leq 4 \leq 5 \leq 6 \leq 7$

Spoiler can win the 3-round EF game as follows:

Spoiler plays 4 in $\mathcal{L}_7$
- Duplicator plays 4 in $\mathcal{L}_6$: Spoiler plays 6 in $\mathcal{L}_7$
  - Duplicator plays 5 in $\mathcal{L}_6$: Spoiler plays 5 in $\mathcal{L}_7$ and wins
  - Duplicator plays 6 in $\mathcal{L}_6$: Spoiler plays 7 in $\mathcal{L}_7$ and wins
- Duplicator plays 3 in $\mathcal{L}_6$: symmetric game (flipped horizontally)

Expressiveness on Linear Orders

Let's look at some very simple structures:

Definition 11.7: A structure $\mathcal{I}$ is a linear order if it has a single binary predicate $\leq$ interpreted as a total, transitive, reflexive and asymmetric relation.

Example 11.9: Consider the following structures:

$\mathcal{L}_7 : 1 \leq 2 \leq 3 \leq 4 \leq 5 \leq 6 \leq 7$
$\mathcal{L}_8 : 1 \leq 2 \leq 3 \leq 4 \leq 5 \leq 6 \leq 7 \leq 8$

Spoiler cannot win the 3-round EF game:

Spoiler plays 4 in $\mathcal{L}_8$: Duplicator plays 4 in $\mathcal{L}_7$
  Spoiler plays 6 in $\mathcal{L}_8$: Duplicator plays 6 in $\mathcal{L}_7$; spoiler cannot win
  Spoiler plays 7 in $\mathcal{L}_8$: Duplicator plays 6 in $\mathcal{L}_7$; spoiler cannot win
Other cases similar: Spoiler never wins

EF Games and Linear Orders

Theorem 11.10: The following are equivalent:

- $\mathcal{L}_m \sim_r \mathcal{L}_n$
- either (1) $m = n$, or (2) $m \geq 2^r - 1$ and $n \geq 2^r - 1$

Proof: See board. □

FO-Definability of Parity

Theorem 11.11: Parity is not FO-definable for linear orders, hence it is not FO-definable for arbitrary databases.

FO-Definability of Parity

Theorem 11.11: Parity is not FO-definable for linear orders, hence it is not FO-definable for arbitrary databases.

Proof:

- Suppose for a contradiction that Parity is FO-definable by some query φ.
- Let r be the quantifier rank of φ.
- Consider databases $\mathcal{L}_m$ and $\mathcal{L}_n$ with $m = 2^r$ and $n = 2^r + 1$.
- We know that $\mathcal{L}_m \sim_r \mathcal{L}_n$, and therefore $\mathcal{L}_m \equiv_r \mathcal{L}_n$.
- Hence, $\mathcal{L}_m \models \varphi$ if and only if $\mathcal{L}_n \models \varphi$.
- But $\mathcal{L}_m \in$ Parity while $\mathcal{L}_n \notin$ Parity.
- Therefore, φ does not FO-define Parity. Contradiction. $\square$

FO-Definability of CONNECTIVITY

The CONNECTIVITY problem over finite graphs is as follows:

CONNECTIVITY

- Input: A finite graph (relational structure with one binary relation "edge")
- Output: "true" if there is an (undirected) path between any pair of vertices

FO-Definability of CONNECTIVITY

The CONNECTIVITY problem over finite graphs is as follows:

CONNECTIVITY

- Input: A finite graph (relational structure with one binary relation "edge")
- Output: "true" if there is an (undirected) path between any pair of vertices

Theorem 11.12: CONNECTIVITY is not FO-definable.

Proof:

- Suppose for a contradiction that CONNECTIVITY is FO-definable using a query φ.
- We show that this would make PARITY FO-definable on linear orders.
- For a linear order $\mathcal{L}$ with order predicate $\leq$, we define a finite graph $\mathcal{G}(\mathcal{L})$ over a binary predicate "edge" such that $\mathcal{G}(\mathcal{L})$ is connected if and only if $\mathcal{L}$ has an odd number of elements.

Defining a Graph From a Linear Order

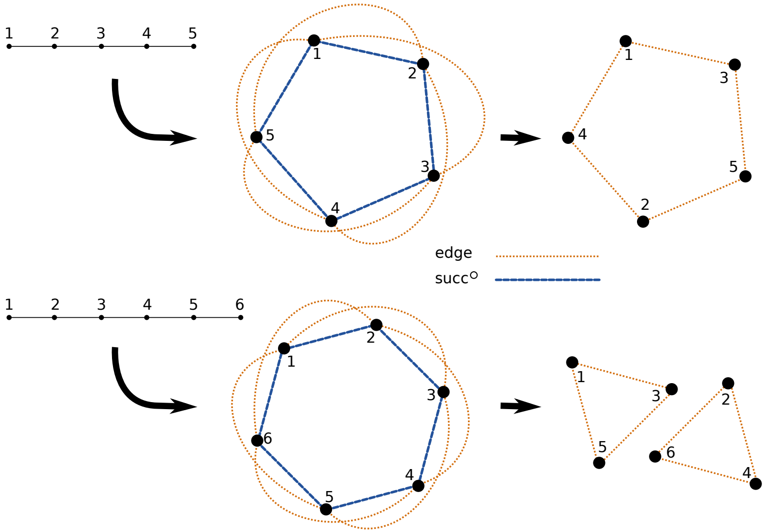
We use abbreviations for the following FO formulas:

$$\begin{aligned}
\text{succ}[x, y] &= (x \le y) \land \neg(y \le x) \land && \quad y \text{ is the successor of } x \\
&\quad \forall z.(z \le x \lor y \le z) && \\
\text{min}[x] &= \forall z.x \le z && \quad x \text{ is the first element} \\
\text{max}[x] &= \forall z.z \le x && \quad x \text{ is the last element} \\
\text{succ}^{\circ}[x, y] &= \text{succ}[x, y] \lor (\text{max}[x] \land \text{min}[y]) && \quad \text{circular version of succ}
\end{aligned}$$

We now define the formula ψ that derives edges from a linear order:

$$\forall x, y.\text{edge}(x, y) \leftrightarrow \exists z.\text{succ}^{\circ}[x, z] \land \text{succ}^{\circ}[z, y]$$

Illustration: Graphs From Linear Orders



Completing the Proof

Observation:
The graph $\mathcal{G}(\mathcal{L})$ is connected if and only if $\mathcal{L}$ has odd parity.

Therefore, if φ FO-defines $\textsc{Connectivity}$ on graphs with predicate edge, then $\neg(\varphi \wedge \psi)$ FO-defines $\textsc{Parity}$ on linear orders.

Since $\textsc{Parity}$ is not FO-definable, no such φ can exist. $\square$

Beyond Linear Orders: Locality

Intuition: Duplicator can win an EF game if selected nodes have the same "neighbourhood"
$\rightsquigarrow$ let's define this for graphs (structures with binary predicates)

Definition 11.13: Consider a graph $\mathcal{G}$. For a natural number $d \geq 0$ and a vertex v, the d-neighbourhood of v, $N(v, d)$, is defined inductively:

- $N(v, 0) = \{v\}$
- $N(v, d + 1) = N(v, d) \cup$
  $\{w \mid w$ is a direct neighbour of some $w' \in N(v, d)\}$

Two vertices v and w have the same d-type if the subgraphs $\mathcal{G}|_{N(v,d)}$ and $\mathcal{G}|_{N(w,d)}$ are isomorphic.
Two graphs are d-equivalent if, for every d-type, they have the same number of d-neighbourhoods of this type.

Locality and FO-definability

A special case of Gaifman's Locality Theorem of first-order logic:

Theorem 11.14: For every integer $r \geq 1$:

- if $\mathcal{G}_1$ is 3^{r-1}-equivalent to $\mathcal{G}_2$
- then $\mathcal{G}_1 \sim_r \mathcal{G}_2$, and thus $\mathcal{G}_1 \equiv_r \mathcal{G}_2$

$\leadsto$ Intuition: FO can only express local properties

How to show that a query mapping M can not be FO-defined:

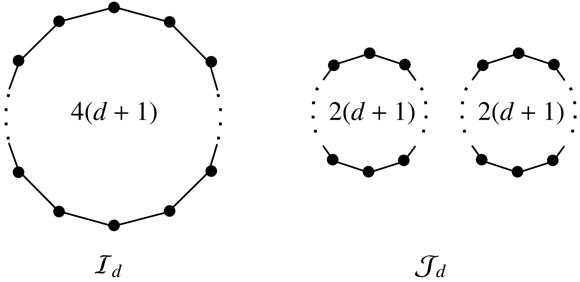
- Let C_M be the class of all databases recognised by M
- Find sequences of graphs $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \ldots \in C_M$ and graphs $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3, \ldots \notin C_M$, such that $\mathcal{I}_i$ is i-equivalent to $\mathcal{J}_i$

$\leadsto$ for any formula φ (however large its quantifier rank r), there is a counterexample $\mathcal{I}_{3^{r-1}} \in C_M$ and $\mathcal{J}_{3^{r-1}} \notin C_M$ that φ cannot distinguish

Connectivity is not FO-definable (Proof 2)

Theorem 11.15: Connectivity is not FO-definable.

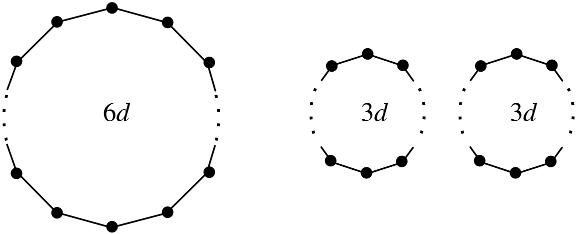
Proof: counterexample for quantifier rank r: set $d = 3^r$



- the only d-type is a path of $2d + 1$ nodes
- $\mathcal{I}_d$ and $\mathcal{J}_d$ are d-equivalent $\square$

Theorem 11.16: 2-COLOURABILITY is not FO-definable.

Proof: counterexample for quantifier rank r: set $d = 3^r$ (odd number)

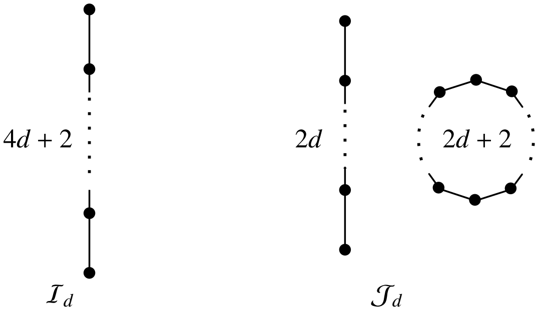


$\mathcal{I}_d$ $\mathcal{J}_d$

- the only d-type is a path of $2d + 1$ nodes
- $\mathcal{I}_d$ and $\mathcal{J}_d$ are d-equivalent $\square$

Theorem 11.17: Acyclicity is not FO-definable.

Proof: counterexample for quantifier rank r: set $d = 3^r$



- d-types are paths of $\leq 2d + 1$ nodes
- $\mathcal{I}_d$ and $\mathcal{J}_d$ are d-equivalent □

Summary: Limits of FO-Queries

FO queries (and hence Relational Calculus) cannot express properties that require a "global" view:

- properties where one needs to follow paths
- properties where one needs to count elements

Remember Lecture 1?

"Stops at distance 2 from Helmholtzstr."

$$R_2 = \delta_{\mathsf{To}\to\mathsf{From}}(\pi_{\mathsf{To}}(\mathsf{Connect} \bowtie R_1))$$

What about all stops reachable from Helmholtzstr.?

Summary: Limits of FO-Queries

FO queries (and hence Relational Calculus) cannot express properties that require a "global" view:

- properties where one needs to follow paths
- properties where one needs to count elements

Remember Lecture 1?

"Stops at distance 2 from Helmholtzstr."

$$R_2 = \delta_{\mathsf{To}\to\mathsf{From}}(\pi_{\mathsf{To}}(\mathsf{Connect} \bowtie R_1))$$

What about all stops reachable from Helmholtzstr.?

$\leadsto$ Not expressible in Relational Calculus

Yet, all examples we saw are in P

$\leadsto$ Is there another query language that could help us?

Summary and Outlook

FO-queries (and thus CQs) cannot express even all tractable query mappings $\leadsto$ FO-definability

Showing that a query is not FO-definable requires some creativity
$\leadsto$ Ehrenfeucht-Fraïssé Games as one approach

FO-queries can only express “local” properties

Possible proof techniques:

- Ehrenfeucht-Fraïssé Games
- Locality Theorems
- For more approaches see
  Chapter 17 of [Abiteboul, Hull, Vianu 1994]

Open questions:

- If FO cannot express all tractable queries, what can?