Finite and Algorithmic Model Theory

Lecture 1 (Dresden 12.10.22, Revised version)

Lecturer: Bartosz "Bart" Bednarczyk

Technische Universität Dresden & Uniwersytet Wrocławski

TECHNISCHE UNIVERSITÄT DRESDEN

Uniwersytet Wrocławski

DeciGUT

European Research Council

Established by the European Commission

Today's agenda

Today's agenda

1. Basics information regarding the course.

Today's agenda

1. Basics information regarding the course.

2. An informal definition of a logic with examples.

Today's agenda

1. Basics information regarding the course.

2. An informal definition of a logic with examples.

3. Potential applications and further research options.

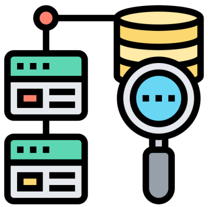
Today's agenda

1. Basics information regarding the course.

2. An informal definition of a logic with examples.

3. Potential applications and further research options.

Query languages?



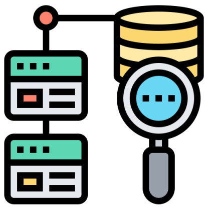
Today's agenda

1. Basics information regarding the course.

2. An informal definition of a logic with examples.

3. Potential applications and further research options.

Query languages?



Formal verification?

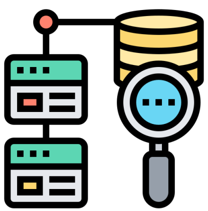
Today's agenda

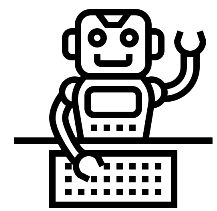
1. Basics information regarding the course.

2. An informal definition of a logic with examples.

3. Potential applications and further research options.

Query languages? Formal verification? Formal languages?

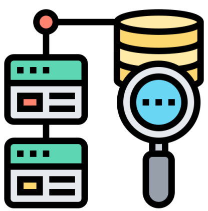




Today's agenda

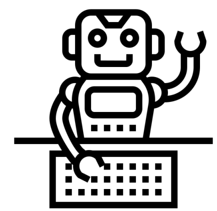
1. Basics information regarding the course.
2. An informal definition of a logic with examples.
3. Potential applications and further research options.

Query languages?

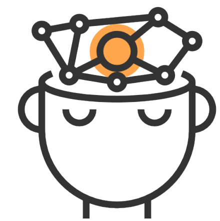


Formal verification?

Formal languages?



Complexity?



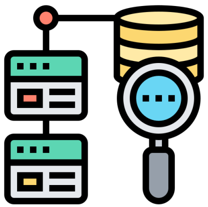
Today's agenda

1. Basics information regarding the course.

2. An informal definition of a logic with examples.

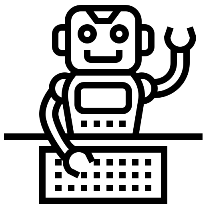
3. Potential applications and further research options.

Query languages?

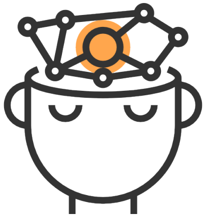


Formal verification?

Formal languages?



Complexity?



4. Recap from BSc studies: Syntax & Semantics of First-Order Logic (FO).

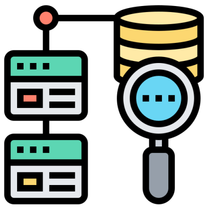
Today's agenda

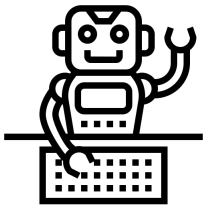
1. Basics information regarding the course.

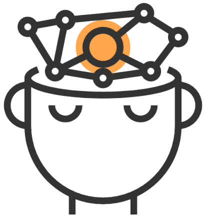
2. An informal definition of a logic with examples.

3. Potential applications and further research options.

Query languages? Formal verification? Formal languages? Complexity?







4. Recap from BSc studies: Syntax & Semantics of First-Order Logic (FO).

5. Basic notations, provability, and Gödel's theorem "$\models$ equals $\vdash$".

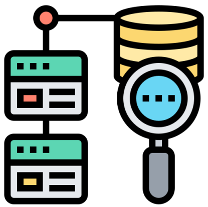
Today’s agenda

1. Basics information regarding the course.

2. An informal definition of a logic with examples.

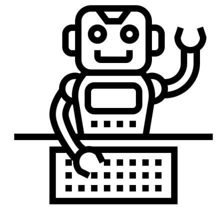
3. Potential applications and further research options.

Query languages?

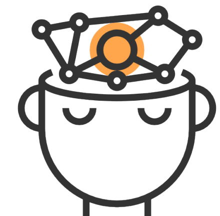


Formal verification?

Formal languages?



Complexity?



4. Recap from BSc studies: Syntax & Semantics of First-Order Logic (FO).

5. Basic notations, provability, and Gödel’s theorem “$\models$ equals $\vdash$”.

6. Gödel’s Compactness theorem with a proof and an application.

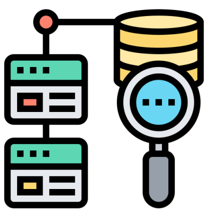
Today's agenda

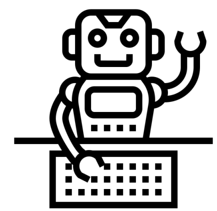
1. Basics information regarding the course.

2. An informal definition of a logic with examples.

3. Potential applications and further research options.

Query languages? Formal verification? Formal languages? Complexity?

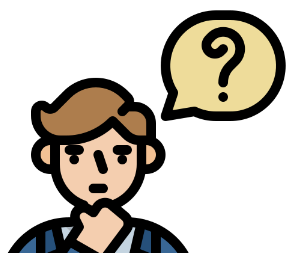





4. Recap from BSc studies: Syntax & Semantics of First-Order Logic (FO).

5. Basic notations, provability, and Gödel's theorem "$\models$ equals $\vdash$".

6. Gödel's Compactness theorem with a proof and an application.



Feel free to ask questions and interrupt me!

Don't be shy! If needed send me an email (bartosz.bednarczyk@cs.uni.wroc.pl) or approach me after the lecture!

Reminder: this is an advanced lecture. Target: people that had fun learning logic during BSc studies!

Course Information

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)

2. Course website: (at [ICCL]) ← check for slides, notes, and exercise lists.

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)
2. Course website: (at [ICCL]) ← check for slides, notes, and exercise lists.
3. Each week a new exercise list will be published. Do not worry if you can't solve all of them.

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)
2. Course website: (at [ICCL]) ← check for slides, notes, and exercise lists.
3. Each week a new exercise list will be published. Do not worry if you can't solve all of them.
4. Oral exam: question about the basic understanding + selected theorems. Intended to be easy!

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)
2. Course website: (at [ICCL]) ← check for slides, notes, and exercise lists.
3. Each week a new exercise list will be published. Do not worry if you can't solve all of them.
4. Oral exam: question about the basic understanding + selected theorems. Intended to be easy!
5. Goal: understand power/limitations of 1st-order logic and selected fragments (with a bit of complexity).

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)
2. Course website: (at [ICCL]) ← check for slides, notes, and exercise lists.
3. Each week a new exercise list will be published. Do not worry if you can't solve all of them.
4. Oral exam: question about the basic understanding + selected theorems. Intended to be easy!
5. Goal: understand power/limitations of 1st-order logic and selected fragments (with a bit of complexity).

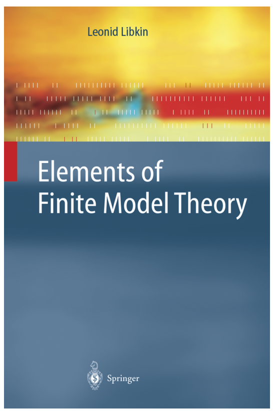
Books and literature.

Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

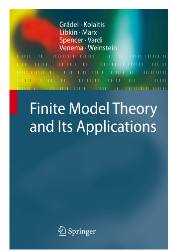
Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

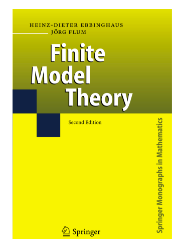
1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)
2. Course website: (at [ICCL]) ← check for slides, notes, and exercise lists.
3. Each week a new exercise list will be published. Do not worry if you can't solve all of them.
4. Oral exam: question about the basic understanding + selected theorems. Intended to be easy!
5. Goal: understand power/limitations of 1st-order logic and selected fragments (with a bit of complexity).

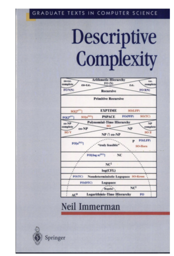


Books and literature.

+ Lecture notes by Martin Otto [HERE] and lecture notes by Erich Grädel [HERE]





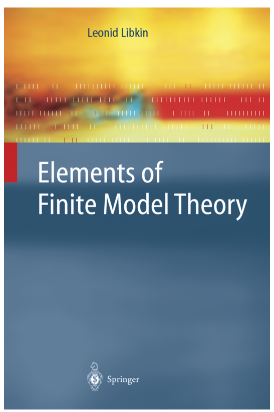


Course Information

https://iccl.inf.tu-dresden.de/web/Finite_and_algorithmic_model_theory_(22/23)_(WS2022)/en

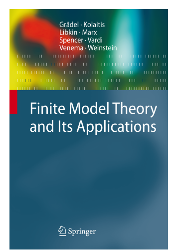
Contact me via email: bartosz.bednarczyk@cs.uni.wroc.pl

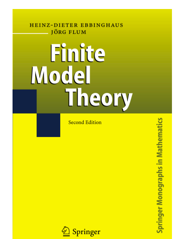
1. Lectures: Wednesday 14:50-16:20 (APB/E007), Tutorials: Thursday 13:00-14:50 (???) (important!)
2. Course website: (at [ICCL]) ← check for slides, notes, and exercise lists.
3. Each week a new exercise list will be published. Do not worry if you can't solve all of them.
4. Oral exam: question about the basic understanding + selected theorems. Intended to be easy!
5. Goal: understand power/limitations of 1st-order logic and selected fragments (with a bit of complexity).

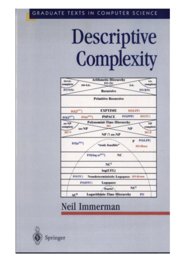


Books and literature.

+ Lecture notes by Martin Otto [HERE] and lecture notes by Erich Grädel [HERE]







Last but Not Least: I offer MSc/PHD research projects for motivated students!

What is a "logic"? A running example.

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory,

What is a "logic"? A running example.

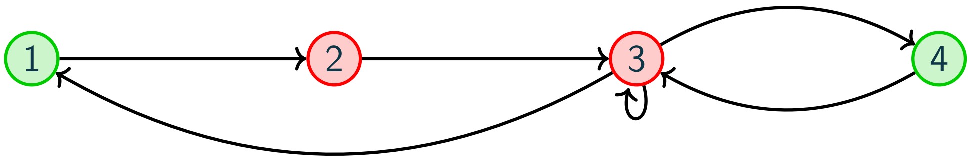
Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

What is a “logic”? A running example.

Naively: a “formal language” for expressing properties of relational structures ($\approx$ hypergraphs).

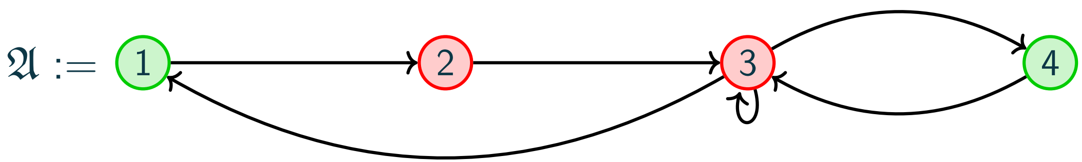
Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström’s theorems.

$\mathfrak{A} :=$ 

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

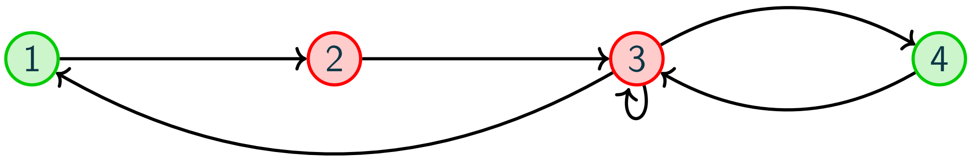


over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

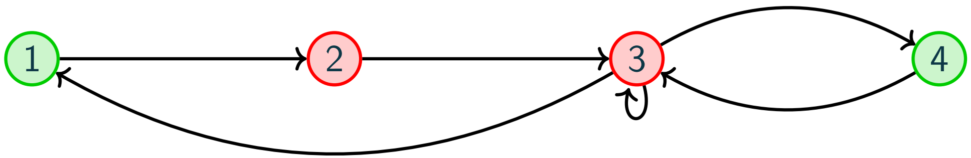
over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures (≈ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

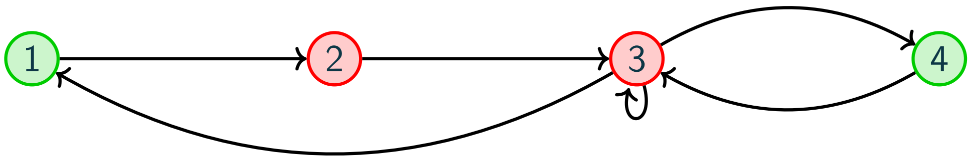
$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

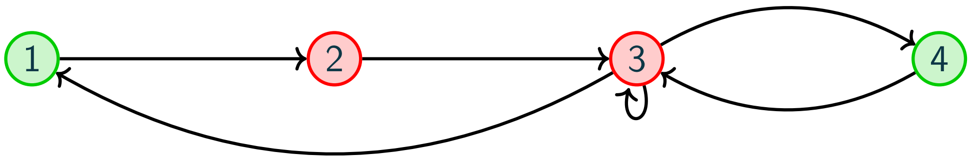
$E^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

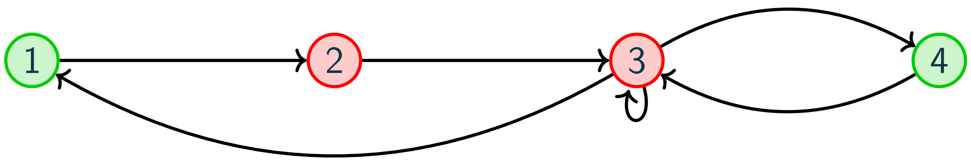
A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g.

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures (≈ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

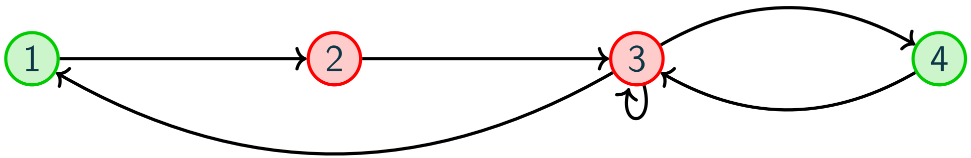
A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

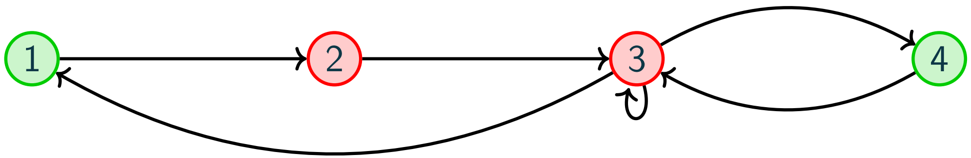
Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}$, $\quad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

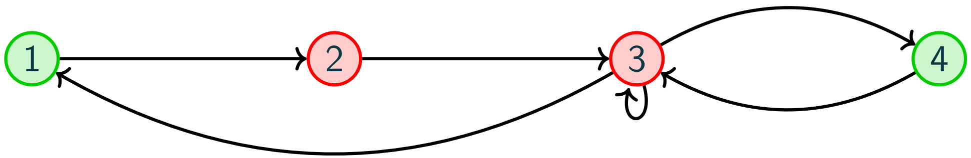
where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(G), \cdot^{\mathfrak{A}}(R), \cdot^{\mathfrak{A}}(E)$ are as above.

Constants $\approx$ elements, unary relations $\approx$ colours, binary (resp. higher-arity) relations $\approx$ (hyper)edges

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

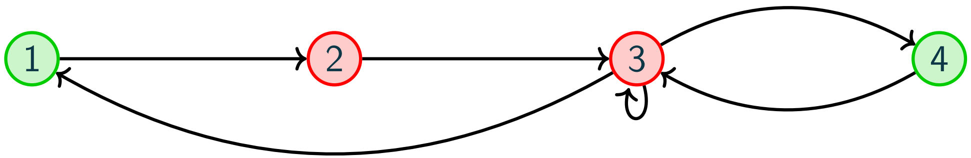
Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(G), \cdot^{\mathfrak{A}}(R), \cdot^{\mathfrak{A}}(E)$ are as above.

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1,2,3,4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

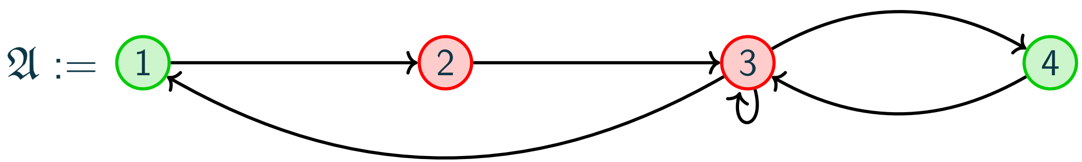
(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x \, (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.



over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

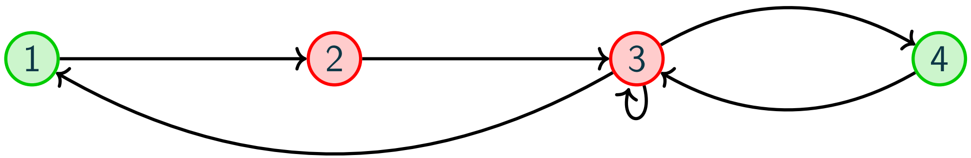
We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

What is a “logic”? A running example.

Naively: a “formal language” for expressing properties of relational structures (≈ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström’s theorems.

$\mathfrak{A} :=$



over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

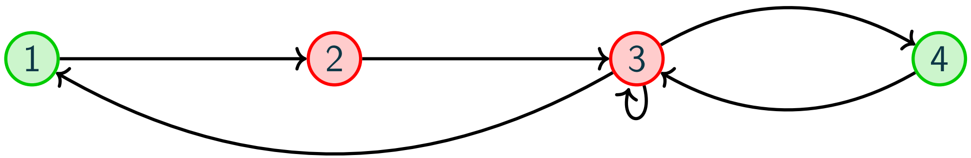
$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

What is a “logic”? A running example.

Naively: a “formal language” for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström’s theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

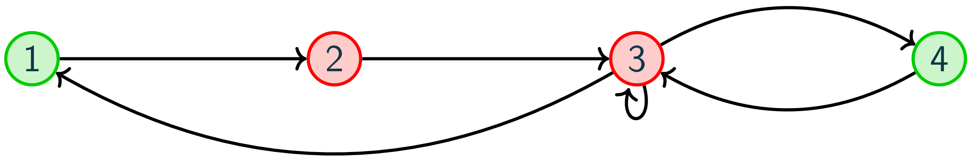
We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ:

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(G), \cdot^{\mathfrak{A}}(R), \cdot^{\mathfrak{A}}(E)$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x \, (G(x) \lor R(x)) \land (G(x) \leftrightarrow \neg R(x))$

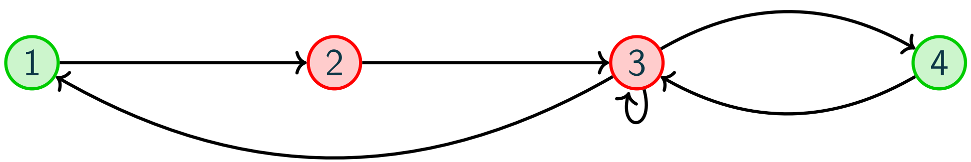
We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(G), \cdot^{\mathfrak{A}}(R), \cdot^{\mathfrak{A}}(E)$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\, (G(x) \vee R(x)) \wedge (G(x) \leftrightarrow \neg R(x))$

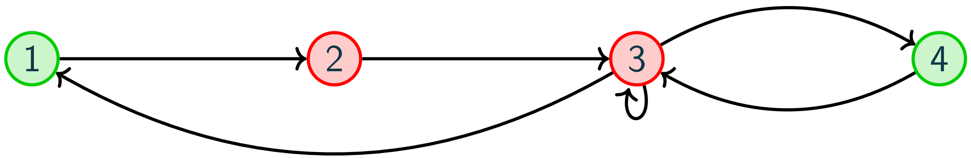
We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}$, $\quad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(G), \cdot^{\mathfrak{A}}(R), \cdot^{\mathfrak{A}}(E)$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\, (G(x) \vee R(x)) \wedge (G(x) \leftrightarrow \neg R(x))$

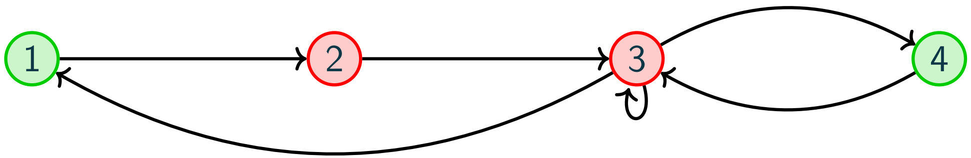
We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1,4\}, \qquad R^{\mathfrak{A}} := \{2,3\}$

$E^{\mathfrak{A}} := \{(1,2),(2,3),(3,1),(3,3)(3,4),(4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1,2,3,4\}$ and $\cdot^{\mathfrak{A}}(G), \cdot^{\mathfrak{A}}(R), \cdot^{\mathfrak{A}}(E)$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (G(x) \lor R(x)) \land (G(x) \leftrightarrow \neg R(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

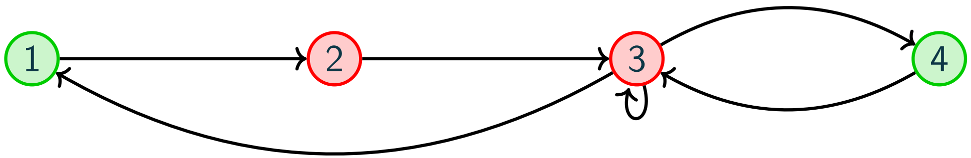
Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\land, \lor, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

What is a “logic”? A running example.

Naively: a “formal language” for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström’s theorems.

$\mathfrak{A} :=$



over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

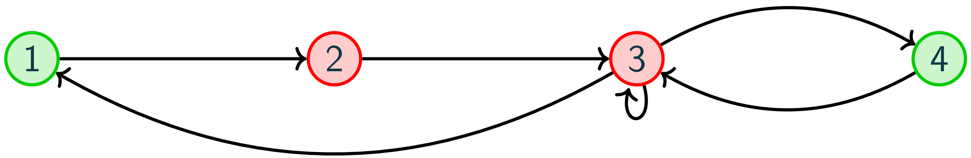
We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg\mathrm{R}(x))$

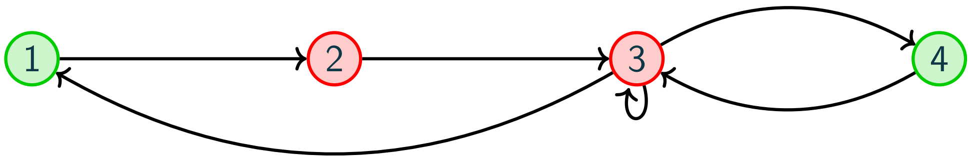
We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \vee_{i=0}^{\infty}, \ldots$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

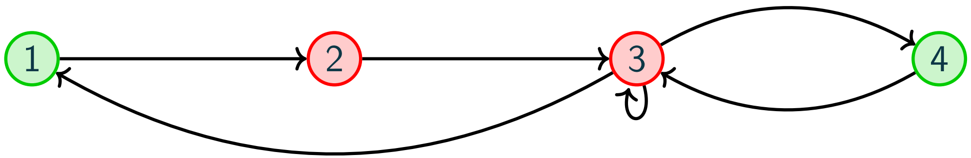
Quantifiers: $\forall, \exists, \exists^{even}, \exists^{=42}, \exists^{35\%}, \exists Set, \Diamond,$

What is a “logic”? A running example.

Naively: a “formal language” for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström’s theorems.

$\mathfrak{A} :=$



over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

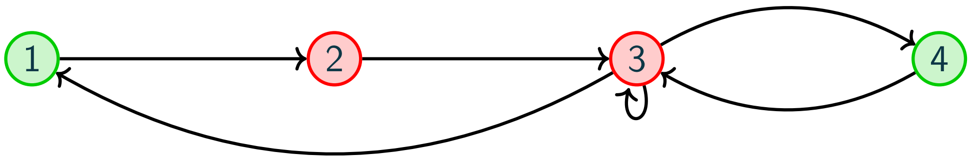
Quantifiers: $\forall, \exists, \exists^{even}, \exists^{=42}, \exists^{35\%}, \exists Set, \Diamond,$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$



over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\ (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

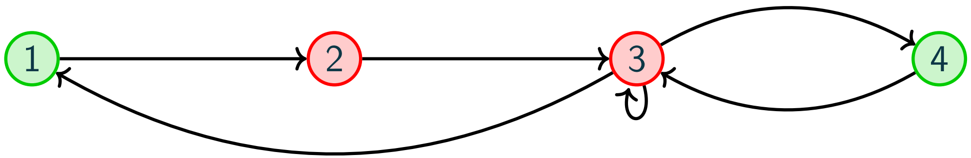
Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

Quantifiers: $\forall, \exists, \exists^{even}, \exists^{=42}, \exists^{35\%}, \exists Set, \Diamond,$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures (≈ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x\, (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

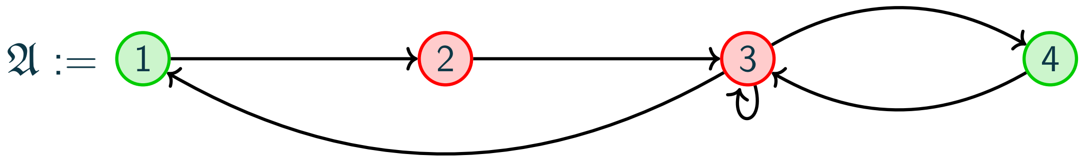
Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

Quantifiers: $\forall, \exists, \exists^{even}, \exists^{=42}, \exists^{35\%}, \exists Set, \Diamond$, Predicates (relational symbols): $\mathrm{P}, \in, =, \sim,$

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.



over a signature $\tau := \{G^{(1)}, R^{(1)}, E^{(2)}\}$

$G^{\mathfrak{A}} := \{1, 4\}, \qquad R^{\mathfrak{A}} := \{2, 3\}$

$E^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1,2,3,4\}$ and $\cdot^{\mathfrak{A}}(G), \cdot^{\mathfrak{A}}(R), \cdot^{\mathfrak{A}}(E)$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x \; (G(x) \vee R(x)) \wedge (G(x) \leftrightarrow \neg R(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

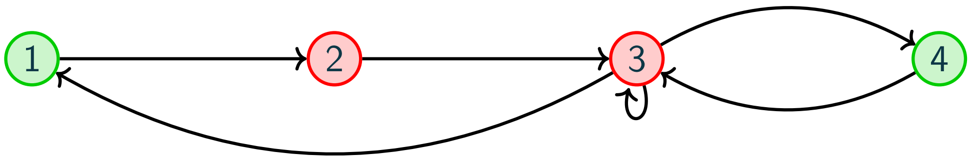
Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

Quantifiers: $\forall, \exists, \exists^{even}, \exists^{=42}, \exists^{35\%}, \exists Set, \Diamond$, Predicates (relational symbols): $P, \in, =, \sim$,

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1,2), (2,3), (3,1), (3,3)(3,4), (4,3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x \, (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

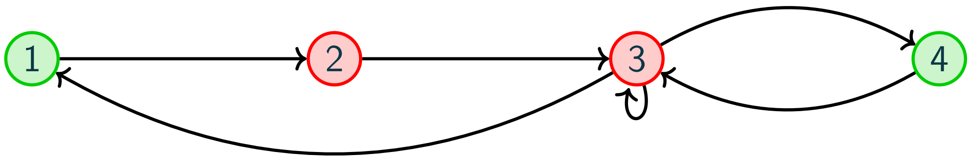
Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

Quantifiers: $\forall, \exists, \exists^{even}, \exists^{=42}, \exists^{35\%}, \exists Set, \Diamond$, Predicates (relational symbols): $\mathrm{P}, \in, =, \sim$,

What is a "logic"? A running example.

Naively: a "formal language" for expressing properties of relational structures ($\approx$ hypergraphs).

Made formal via abstract model theory, c.f. article at ncatlab.org and Lindström's theorems.

$\mathfrak{A} :=$ 

over a signature $\tau := \{\mathrm{G}^{(1)}, \mathrm{R}^{(1)}, \mathrm{E}^{(2)}\}$

$\mathrm{G}^{\mathfrak{A}} := \{1, 4\}, \qquad \mathrm{R}^{\mathfrak{A}} := \{2, 3\}$

$\mathrm{E}^{\mathfrak{A}} := \{(1, 2), (2, 3), (3, 1), (3, 3)(3, 4), (4, 3)\}$

A signature contains (at most countably* many) constant and relation symbols (each with a fixed arity).

Structure = Domain + interpretation of symbols, e.g. $\mathfrak{A} := (A, \cdot^{\mathfrak{A}})$ depicted above,

where $A = \{1, 2, 3, 4\}$ and $\cdot^{\mathfrak{A}}(\mathrm{G}), \cdot^{\mathfrak{A}}(\mathrm{R}), \cdot^{\mathfrak{A}}(\mathrm{E})$ are as above.

Example (of a First-Order Logic (FO) Formula)

(in a coloured graph:) Any node is either green or red.

$\varphi := \forall x \, (\mathrm{G}(x) \vee \mathrm{R}(x)) \wedge (\mathrm{G}(x) \leftrightarrow \neg \mathrm{R}(x))$

We write $\mathfrak{A} \models \varphi$ to indicate that $\mathfrak{A}$ satisfies φ or $\mathfrak{A}$ is a model of φ.

Formulae often employ: Variables: $x, y, z, X, Y, \ldots$ Boolean connectives: $\wedge, \vee, \neg, \leftrightarrow, \bigvee_{i=0}^{\infty}, \ldots$

Quantifiers: $\forall, \exists, \exists^{even}, \exists^{=42}, \exists^{35\%}, \exists Set, \Diamond$, Predicates (relational symbols): $\mathrm{P}, \in, =, \sim$, and more?

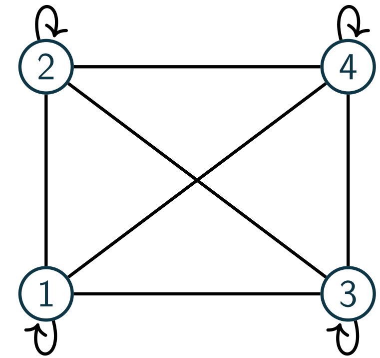
More examples I.

More examples I.

Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

More examples I.

Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

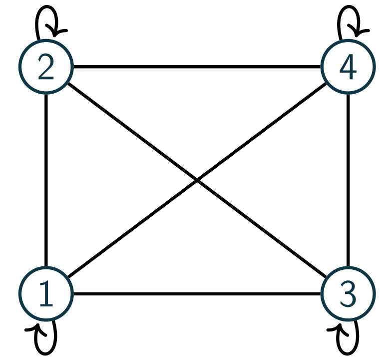


$\models \varphi_{\text{4clique}}$

More examples I.

Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here $\mathrm{E} =$ edge relation].)

1. There are precisely 4 elements . . .



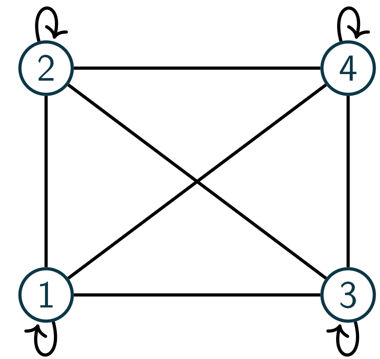
$\models \varphi_{\mathsf{4clique}}$

More examples I.

Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements . . .

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \Big(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4$$
$$\wedge \forall x \big[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \big] \Big)$$



$\models \varphi_{\text{4clique}}$

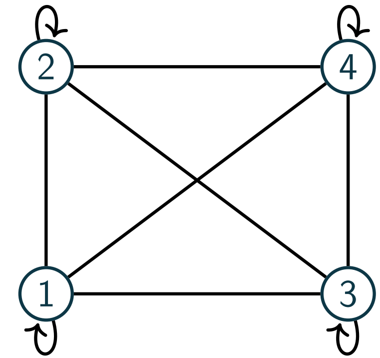
More examples I.

Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements . . .

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \left(x_1 \neq x_2 \land x_1 \neq x_3 \land x_1 \neq x_4 \land x_2 \neq x_3 \land x_2 \neq x_4 \land x_3 \neq x_4 \right.$$
$$\left. \land \forall x \big[x = x_1 \lor x = x_2 \lor x = x_3 \lor x = x_4 \big] \right)$$

2. and any two of them are linked by E.



$\models \varphi_{\text{4clique}}$

More examples I.

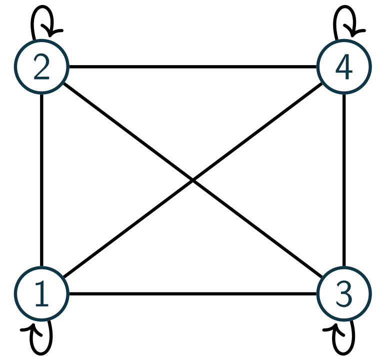
Exercise (An FO$[\{\mathrm{E}^{(2)}\}]$ formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements . . .

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \left(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4 \right.$$
$$\left. \wedge \forall x \big[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \big] \right)$$

2. and any two of them are linked by E.

$$\wedge \ \forall x \forall y \ \mathrm{E}(x, y).$$



$\models \varphi_{\text{4clique}}$

More examples I.

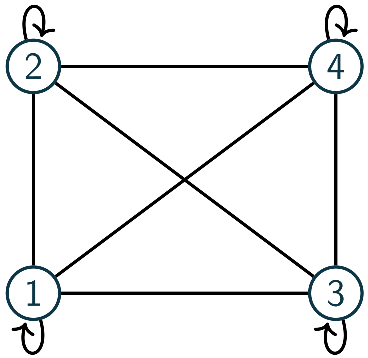
Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements ...

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \Big(x_1 \neq x_2 \land x_1 \neq x_3 \land x_1 \neq x_4 \land x_2 \neq x_3 \land x_2 \neq x_4 \land x_3 \neq x_4$$
$$\land \forall x \big[x = x_1 \lor x = x_2 \lor x = x_3 \lor x = x_4 \big] \Big)$$

2. and any two of them are linked by E.

$\land\ \forall x \forall y\ \mathrm{E}(x, y)$.



$\models \varphi_{4\text{clique}}$

More examples I.

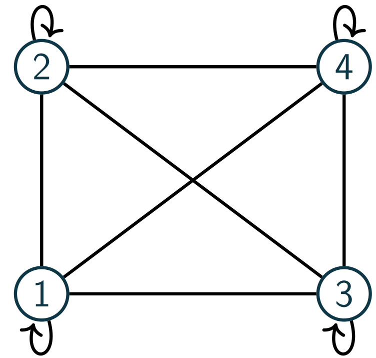
Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements . . .

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \big(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4$$
$$\wedge \forall x \big[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \big] \big)$$

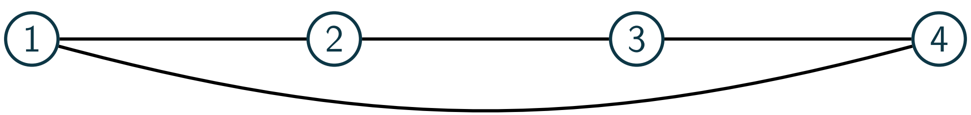
2. and any two of them are linked by E.

$\wedge\ \forall x \forall y\ \mathrm{E}(x, y)$.



$\models \varphi_{4\text{clique}}$

Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

$\mathfrak{G} :=$ 

More examples I.

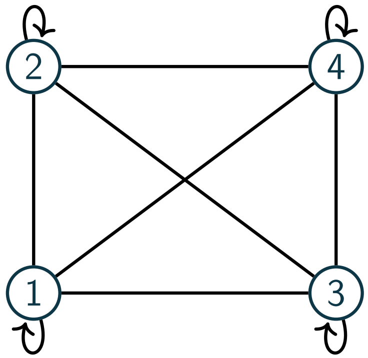
Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements . . .

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \Big(x_1 \neq x_2 \land x_1 \neq x_3 \land x_1 \neq x_4 \land x_2 \neq x_3 \land x_2 \neq x_4 \land x_3 \neq x_4$$
$$\land \forall x \big[x = x_1 \lor x = x_2 \lor x = x_3 \lor x = x_4 \big] \Big)$$

2. and any two of them are linked by E.

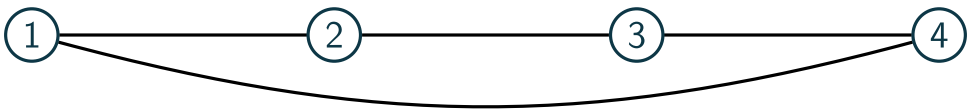
$$\land \forall x \forall y \ \mathrm{E}(x, y).$$



$\models \varphi_{\text{4clique}}$

Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

$\mathfrak{G} :=$



$$\varphi_{2COL} = \exists \mathrm{G} \exists \mathrm{R} \ (x \in \mathrm{G} \lor x \in \mathrm{R}) \land (x \in \mathrm{G} \leftrightarrow x \notin \mathrm{R}) \ \land \ \varphi_{ok}$$

More examples I.

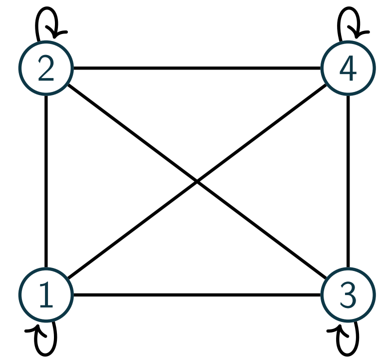
Exercise (An $\mathrm{FO}[\{\mathrm{E}^{(2)}\}]$ formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements . . .

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \Big(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4$$
$$\wedge \forall x \big[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \big] \Big)$$

2. and any two of them are linked by E.

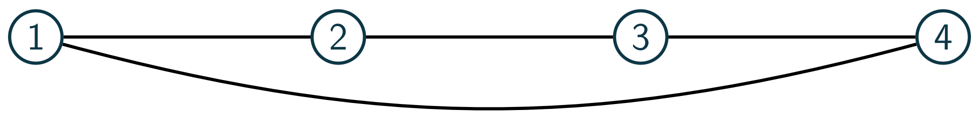
$$\wedge \; \forall x \forall y \; \mathrm{E}(x, y).$$



$\models \varphi_{\text{4clique}}$

Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

$\mathfrak{G} :=$



$$\varphi_{2COL} \; = \; \exists \mathrm{G} \exists \mathrm{R} \; (x \in \mathrm{G} \vee x \in \mathrm{R}) \wedge (x \in \mathrm{G} \leftrightarrow x \notin \mathrm{R}) \; \wedge \; \varphi_{ok}$$
$$\varphi_{ok} = \forall x (x \in \mathrm{G} \rightarrow (\forall y \; E(x, y) \rightarrow y \in \mathrm{R})) \wedge \forall x (x \in \mathrm{R} \rightarrow (\forall y \; E(x, y) \rightarrow y \in \mathrm{G}))$$

More examples I.

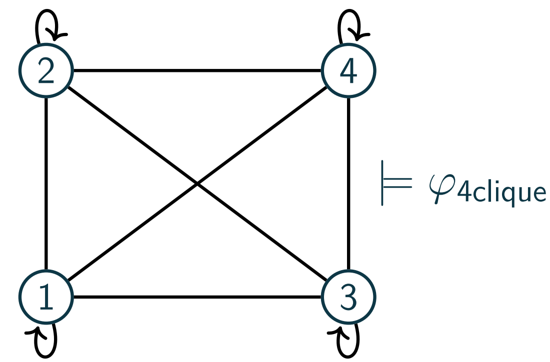
Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements . . .

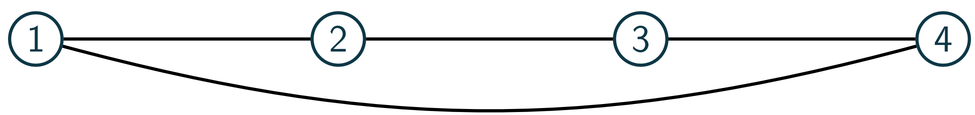
$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \left(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4 \right.$$
$$\left. \wedge \forall x \left[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \right] \right)$$

2. and any two of them are linked by E.

$\wedge\ \forall x \forall y\ \mathrm{E}(x, y)$.



Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

$\mathfrak{G} :=$ 

$$\varphi_{2COL} = \exists \mathrm{G} \exists \mathrm{R}\ (x \in \mathrm{G} \vee x \in \mathrm{R}) \wedge (x \in \mathrm{G} \leftrightarrow x \notin \mathrm{R})\ \wedge\ \varphi_{ok}$$
$$\varphi_{ok} = \forall x (x \in \mathrm{G} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{R})) \wedge \forall x (x \in \mathrm{R} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{G}))$$

Quantification over sets:

More examples I.

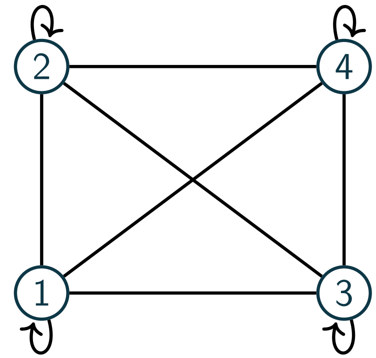
Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements ...

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \Big(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4$$
$$\wedge \forall x \big[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \big] \Big)$$

2. and any two of them are linked by E.

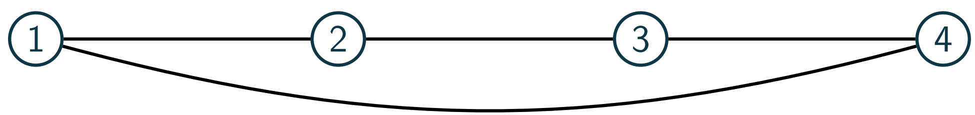
$\wedge \; \forall x \forall y \; \mathrm{E}(x, y)$.



$\models \varphi_{4\text{clique}}$

Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

$\mathfrak{G} :=$



$$\varphi_{2COL} \; = \; \exists \mathrm{G} \exists \mathrm{R} \; (x \in \mathrm{G} \vee x \in \mathrm{R}) \wedge (x \in \mathrm{G} \leftrightarrow x \notin \mathrm{R}) \; \wedge \; \varphi_{ok}$$
$$\varphi_{ok} = \forall x (x \in \mathrm{G} \to (\forall y \; E(x, y) \to y \in \mathrm{R})) \wedge \forall x (x \in \mathrm{R} \to (\forall y \; E(x, y) \to y \in \mathrm{G}))$$

More examples I.

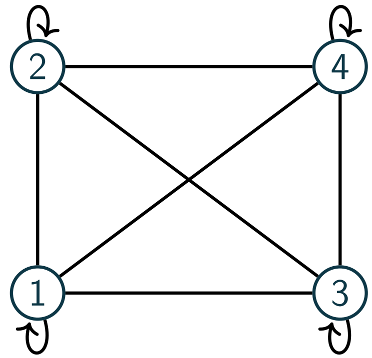
Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements ...

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \Big(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4$$
$$\wedge \forall x \big[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \big] \Big)$$

2. and any two of them are linked by E.

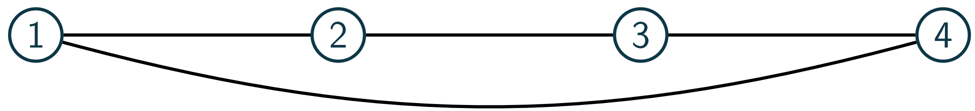
$$\wedge\ \forall x \forall y\ \mathrm{E}(x, y).$$



$\models \varphi_{4\text{clique}}$

Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

$\mathfrak{G} :=$



$$\varphi_{2COL} = \exists \mathrm{G} \exists \mathrm{R}\ (x \in \mathrm{G} \vee x \in \mathrm{R}) \wedge (x \in \mathrm{G} \leftrightarrow x \notin \mathrm{R}) \wedge \varphi_{ok}$$
$$\varphi_{ok} = \forall x (x \in \mathrm{G} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{R})) \wedge \forall x (x \in \mathrm{R} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{G}))$$

There exists a colouring with G and R and it is correct

More examples I.

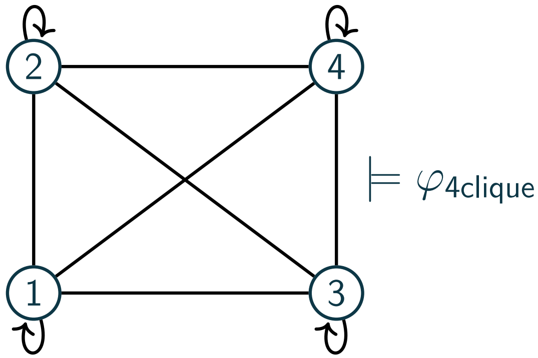
Exercise (An $\mathrm{FO}[\{\mathrm{E}^{(2)}\}]$ formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements ...

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \Big(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4$$
$$\wedge \forall x \big[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \big] \Big)$$

2. and any two of them are linked by E.

$$\wedge\ \forall x \forall y\ \mathrm{E}(x, y).$$



Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

$$\varphi_{2COL} = \exists \mathrm{G} \exists \mathrm{R}\ (x \in \mathrm{G} \vee x \in \mathrm{R}) \wedge (x \in \mathrm{G} \leftrightarrow x \notin \mathrm{R}) \wedge \varphi_{ok}$$
$$\varphi_{ok} = \forall x (x \in \mathrm{G} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{R})) \wedge \forall x (x \in \mathrm{R} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{G}))$$

More examples I.

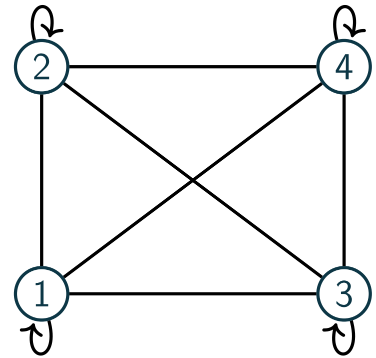
Exercise (An FO[$\{\mathrm{E}^{(2)}\}$] formula/query testing if a graph is a 4-element clique [here E = edge relation].)

1. There are precisely 4 elements ...

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \left(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4 \right.$$
$$\left. \wedge \forall x \left[x = x_1 \vee x = x_2 \vee x = x_3 \vee x = x_4 \right] \right)$$

2. and any two of them are linked by E.

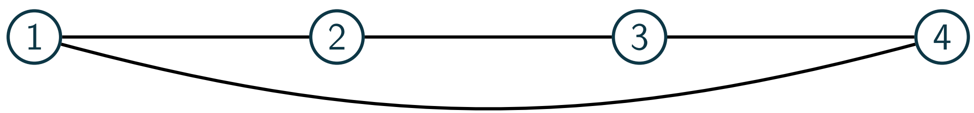
$$\wedge\ \forall x \forall y\ \mathrm{E}(x, y).$$



$\models \varphi_{\text{4clique}}$

Exercise (Write a formula over $\{\mathrm{E}^{(2)}\}$ checking if a graph is two-colorable.)

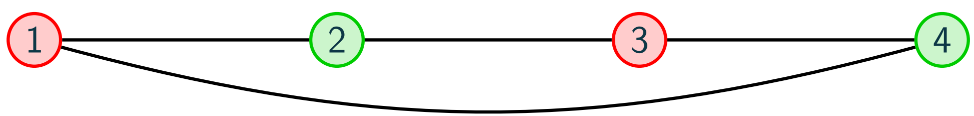
$\mathfrak{G} :=$



$\models \varphi_{2COL}$

$$\varphi_{2COL} = \exists \mathrm{G} \exists \mathrm{R}\ (x \in \mathrm{G} \vee x \in \mathrm{R}) \wedge (x \in \mathrm{G} \leftrightarrow x \notin \mathrm{R}) \wedge \varphi_{ok}$$
$$\varphi_{ok} = \forall x (x \in \mathrm{G} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{R})) \wedge \forall x (x \in \mathrm{R} \rightarrow (\forall y\ E(x, y) \rightarrow y \in \mathrm{G}))$$

$\mathfrak{G}' :=$



$\models \varphi_{ok}$

More examples II.

More examples II.

Exercise (Write an FO$[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial:

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$

More examples II.

Exercise (Write an FO$[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$

2. Case $k = 1$ is easy too:

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$

2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$
3. Case $k = 2$ is a tiny bit harder:

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$

2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$

3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, \mathtt{b})$

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$

2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$

3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, \mathtt{b})$

4. Case $k = 3$ is a similar:

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, \mathtt{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \exists x_2 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, x_2) \wedge \mathrm{E}(x_2, \mathtt{b})$

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \mathrm{E}(\mathtt{a}, x_1) \land \mathrm{E}(x_1, \mathtt{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathsf{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \exists x_2 \mathrm{E}(\mathtt{a}, x_1) \land \mathrm{E}(x_1, x_2) \land \mathrm{E}(x_2, \mathtt{b})$
5. So for any $k \geq 2$ just take:

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\mathrm{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathrm{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathrm{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathrm{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, \mathtt{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathrm{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \exists x_2 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, x_2) \wedge \mathrm{E}(x_2, \mathtt{b})$
5. So for any $k \geq 2$ just take: Take $\varphi_k^{\mathrm{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathtt{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathtt{b})$

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathsf{a}, \mathsf{b}\}]$ formula $\varphi_k^{\mathsf{reach(a,b)}}$ testing if there is a path from a to b of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach(a,b)}} := \mathsf{a} = \mathsf{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach(a,b)}} := \mathrm{E}(\mathsf{a}, \mathsf{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathsf{reach(a,b)}} := \exists x_1 \mathrm{E}(\mathsf{a}, x_1) \land \mathrm{E}(x_1, \mathsf{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathsf{reach(a,b)}} := \exists x_1 \exists x_2 \mathrm{E}(\mathsf{a}, x_1) \land \mathrm{E}(x_1, x_2) \land \mathrm{E}(x_2, \mathsf{b})$
5. So for any $k \geq 2$ just take: Take $\varphi_k^{\mathsf{reach(a,b)}} := \exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathsf{a}, x_1) \land \bigwedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \land \mathrm{E}(x_{k-1}, \mathsf{b})$

Question (Can we do better in terms the total number of quantifiers?)

More examples II.

Exercise (Write an FO$[\{\mathrm{E}^{(2)}, \mathtt{a}, \mathtt{b}\}]$ formula $\varphi_k^{\text{reach}(\mathtt{a},\mathtt{b})}$ testing if there is a path from $\mathtt{a}$ to $\mathtt{b}$ of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\text{reach}(\mathtt{a},\mathtt{b})} := \mathtt{a} = \mathtt{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\text{reach}(\mathtt{a},\mathtt{b})} := \mathrm{E}(\mathtt{a}, \mathtt{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\text{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, \mathtt{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\text{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \exists x_2 \mathrm{E}(\mathtt{a}, x_1) \wedge \mathrm{E}(x_1, x_2) \wedge \mathrm{E}(x_2, \mathtt{b})$
5. So for any $k \geq 2$ just take: Take $\varphi_k^{\text{reach}(\mathtt{a},\mathtt{b})} := \exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathtt{a}, x_1) \wedge \bigwedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathtt{b})$

Question (Can we do better in terms the total number of quantifiers?)

Current state of the art: $\log_2(k) - \mathcal{O}(1) \leq ??? \leq 3\log_3(k) + \mathcal{O}(1)$ by Fagin at al. [MFCS 2022]

More examples II.

Exercise (Write an FO$[\{\mathrm{E}^{(2)}, \mathrm{a}, \mathrm{b}\}]$ formula $\varphi_k^{\mathrm{reach}(\mathrm{a},\mathrm{b})}$ testing if there is a path from a to b of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathrm{reach}(\mathrm{a},\mathrm{b})} := \mathrm{a} = \mathrm{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathrm{reach}(\mathrm{a},\mathrm{b})} := \mathrm{E}(\mathrm{a}, \mathrm{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathrm{reach}(\mathrm{a},\mathrm{b})} := \exists x_1 \mathrm{E}(\mathrm{a}, x_1) \wedge \mathrm{E}(x_1, \mathrm{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathrm{reach}(\mathrm{a},\mathrm{b})} := \exists x_1 \exists x_2 \mathrm{E}(\mathrm{a}, x_1) \wedge \mathrm{E}(x_1, x_2) \wedge \mathrm{E}(x_2, \mathrm{b})$
5. So for any $k \geq 2$ just take: Take $\varphi_k^{\mathrm{reach}(\mathrm{a},\mathrm{b})} := \exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathrm{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathrm{b})$

Question (Can we do better in terms the total number of quantifiers?)

Current state of the art: $\log_2(k) - \mathcal{O}(1) \leq ??? \leq 3\log_3(k) + \mathcal{O}(1)$ by Fagin at al. [MFCS 2022]

Exercise (Write a formula φ^{conn} over $\{\mathrm{E}^{(2)}\}$ testing if a structure is E-connected.)

More examples II.

Exercise (Write an FO[$\{\mathrm{E}^{(2)}, \mathsf{a}, \mathsf{b}\}$] formula $\varphi_k^{\mathsf{reach}(\mathsf{a},\mathsf{b})}$ testing if there is a path from a to b of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \mathsf{a} = \mathsf{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \mathrm{E}(\mathsf{a}, \mathsf{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \exists x_1 \mathrm{E}(\mathsf{a}, x_1) \wedge \mathrm{E}(x_1, \mathsf{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \exists x_1 \exists x_2 \mathrm{E}(\mathsf{a}, x_1) \wedge \mathrm{E}(x_1, x_2) \wedge \mathrm{E}(x_2, \mathsf{b})$
5. So for any $k \geq 2$ just take: Take $\varphi_k^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathsf{a}, x_1) \wedge \bigwedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathsf{b})$

Question (Can we do better in terms the total number of quantifiers?)

Current state of the art: $\log_2(k) - \mathcal{O}(1) \leq ??? \leq 3\log_3(k) + \mathcal{O}(1)$ by Fagin at al. [MFCS 2022]

Exercise (Write a formula φ^{conn} over $\{\mathrm{E}^{(2)}\}$ testing if a structure is E-connected.)

$$\varphi^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \forall x \forall y \ \bigvee_{i=0}^{\infty} \varphi_k^{\mathsf{reach}(\mathsf{a},\mathsf{b})}[\mathsf{a}/x, \mathsf{b}/y].$$

More examples II.

Exercise (Write an FO$[\{\mathrm{E}^{(2)}, \mathsf{a}, \mathsf{b}\}]$ formula $\varphi_k^{\mathsf{reach}(\mathsf{a},\mathsf{b})}$ testing if there is a path from a to b of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \mathsf{a} = \mathsf{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \mathrm{E}(\mathsf{a}, \mathsf{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \exists x_1 \mathrm{E}(\mathsf{a}, x_1) \land \mathrm{E}(x_1, \mathsf{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \exists x_1 \exists x_2 \mathrm{E}(\mathsf{a}, x_1) \land \mathrm{E}(x_1, x_2) \land \mathrm{E}(x_2, \mathsf{b})$
5. So for any $k \geq 2$ just take: Take $\varphi_k^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathsf{a}, x_1) \land \bigwedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \land \mathrm{E}(x_{k-1}, \mathsf{b})$

Question (Can we do better in terms the total number of quantifiers?)

Current state of the art: $\log_2(k) - \mathcal{O}(1) \leq ??? \leq 3\log_3(k) + \mathcal{O}(1)$ by Fagin at al. [MFCS 2022]

Exercise (Write a formula φ^{conn} over $\{\mathrm{E}^{(2)}\}$ testing if a structure is E-connected.)

Is there a chance to get an FO formula?

$$\varphi^{\mathsf{reach}(\mathsf{a},\mathsf{b})} := \forall x \forall y \ \bigvee_{i=0}^{\infty} \varphi_k^{\mathsf{reach}(\mathsf{a},\mathsf{b})}[\mathsf{a}/x, \mathsf{b}/y].$$

More examples II.

Exercise (Write an $\mathrm{FO}[\{\mathrm{E}^{(2)}, \mathrm{a}, \mathrm{b}\}]$ formula $\varphi_k^{\mathrm{reach(a,b)}}$ testing if there is a path from a to b of length k.)

1. Case $k = 0$ is trivial: Take $\varphi_0^{\mathrm{reach(a,b)}} := \mathrm{a} = \mathrm{b}$
2. Case $k = 1$ is easy too: Take $\varphi_1^{\mathrm{reach(a,b)}} := \mathrm{E}(\mathrm{a}, \mathrm{b})$
3. Case $k = 2$ is a tiny bit harder: Take $\varphi_2^{\mathrm{reach(a,b)}} := \exists x_1 \mathrm{E}(\mathrm{a}, x_1) \wedge \mathrm{E}(x_1, \mathrm{b})$
4. Case $k = 3$ is a similar: Take $\varphi_3^{\mathrm{reach(a,b)}} := \exists x_1 \exists x_2 \mathrm{E}(\mathrm{a}, x_1) \wedge \mathrm{E}(x_1, x_2) \wedge \mathrm{E}(x_2, \mathrm{b})$
5. So for any $k \geq 2$ just take: Take $\varphi_k^{\mathrm{reach(a,b)}} := \exists x_1 \ldots \exists x_{k-1}\ \mathrm{E}(\mathrm{a}, x_1) \wedge \bigwedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathrm{b})$

Question (Can we do better in terms the total number of quantifiers?)

Current state of the art: $\log_2(k) - \mathcal{O}(1) \leq ??? \leq 3\log_3(k) + \mathcal{O}(1)$ by Fagin at al. [MFCS 2022]

Exercise (Write a formula φ^{conn} over $\{\mathrm{E}^{(2)}\}$ testing if a structure is E-connected.)

$\varphi^{\mathrm{reach(a,b)}} := \forall x \forall y\ \bigvee_{i=0}^{\infty} \varphi_k^{\mathrm{reach(a,b)}}[\mathrm{a}/x, \mathrm{b}/y].$

Is there a chance to get an FO formula?

No. And we will show it today!

Motivations I: why do we care about logic?

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for “computer science”.

```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \quad \varphi(i)$

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \varphi(i)$

$$\varphi(i) = \exists n \exists s\ \textsc{Candidate}(i, n, s) \land \textsc{Appl}(\texttt{"Computer Science"}, i)$$

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

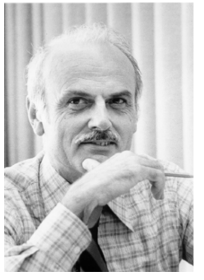
```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \varphi(i)$

$$\varphi(i) = \exists n \exists s\ \textsc{Candidate}(i, n, s) \land \textsc{Appl}(\texttt{"Computer Science"}, i)$$

Theorem (Codd 1971)

Basic SQL ≈ First-Order Logic



Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

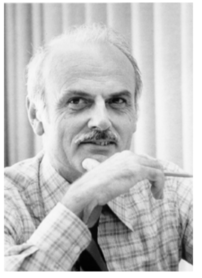
```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \varphi(i)$

$$\varphi(i) = \exists n \exists s\ \textsc{Candidate}(i, n, s) \land \textsc{Appl}(\texttt{"Computer Science"}, i)$$

Theorem (Codd 1971)

Basic SQL $\approx$ First-Order Logic



Other useful logic: Datalog $\approx$ SQL + recursion

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

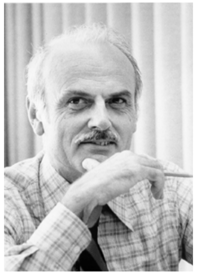
```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \quad \varphi(i)$

$$\varphi(i) = \exists n \exists s \, \textsc{Candidate}(i, n, s) \land \textsc{Appl}(\texttt{"Computer Science"}, i)$$

Theorem (Codd 1971)

Basic SQL $\approx$ First-Order Logic



Other useful logic: Datalog $\approx$ SQL + recursion

1. VLog: a rule engine for querying data graphs

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

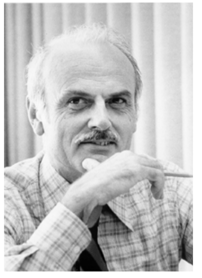
```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \varphi(i)$

$$\varphi(i) = \exists n \exists s\ \textsc{Candidate}(i, n, s) \land \textsc{Appl}(\texttt{"Computer Science"}, i)$$

Theorem (Codd 1971)

Basic SQL $\approx$ First-Order Logic



Other useful logic: Datalog $\approx$ SQL + recursion

1. VLog: a rule engine for querying data graphs
2. Vadalog: querying data graphs based on Datalog

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for "computer science".

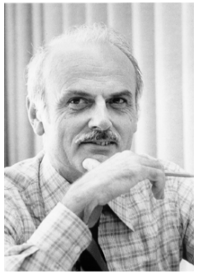
```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \varphi(i)$

$$\varphi(i) = \exists n \exists s\ \textsc{Candidate}(i, n, s) \land \textsc{Appl}(\texttt{"Computer Science"}, i)$$

Theorem (Codd 1971)

Basic SQL $\approx$ First-Order Logic



Other useful logic: Datalog $\approx$ SQL + recursion

1. VLog: a rule engine for querying data graphs
2. Vadalog: querying data graphs based on Datalog

Nice lecture on VadaLog by Gottlob [here], and a course on knowledge graphs by Krötzsch [here].

Motivations I: why do we care about logic?

Query: Give me IDs of all candidates who applied for “computer science”.

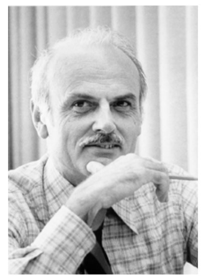
```
SELECT CandID
FROM Candidate
WHERE Major = "Computer Science"
```

$\rightsquigarrow \quad \varphi(i)$

$$\varphi(i) = \exists n \exists s \ \textsc{Candidate}(i, n, s) \land \textsc{Appl}(\texttt{"Computer Science"}, i)$$

Theorem (Codd 1971)

Basic SQL ≈ First-Order Logic



Other useful logic: Datalog ≈ SQL + recursion

1. VLog: a rule engine for querying data graphs
2. Vadalog: querying data graphs based on Datalog

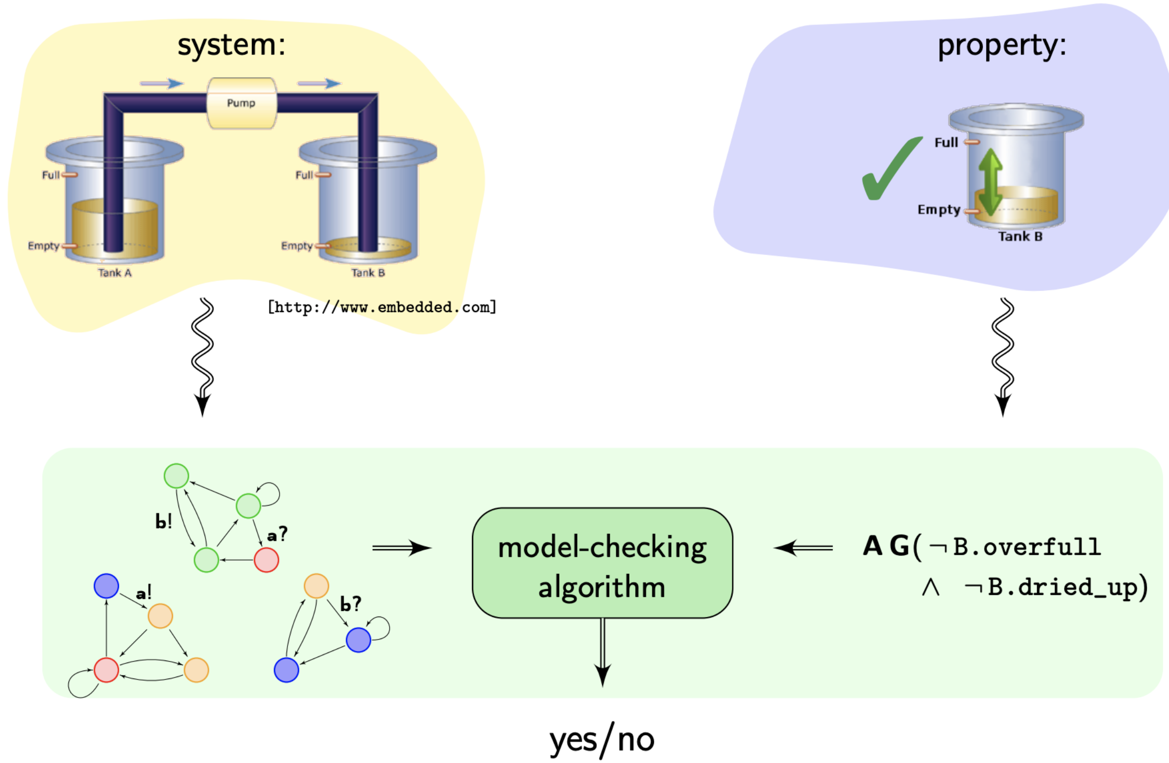
Nice lecture on VadaLog by Gottlob [here], and a course on knowledge graphs by Krötzsch [here].

Description logics: a family of logics for knowledge representation.

W3C® SNOMED CT The global language of healthcare protégé Dublin Core Metadata Initiative Making it easier to find information

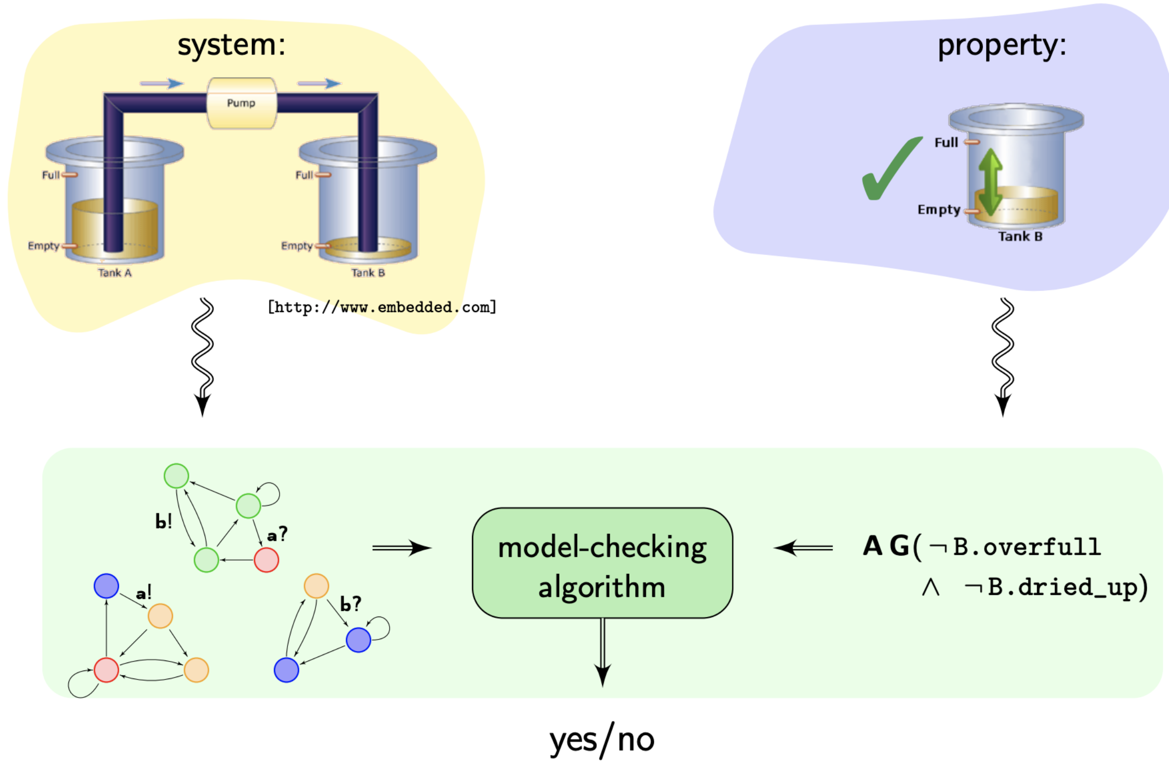
Motivations II: why do we care about logic?

Motivations II: why do we care about logic?



Motivations II: why do we care about logic?

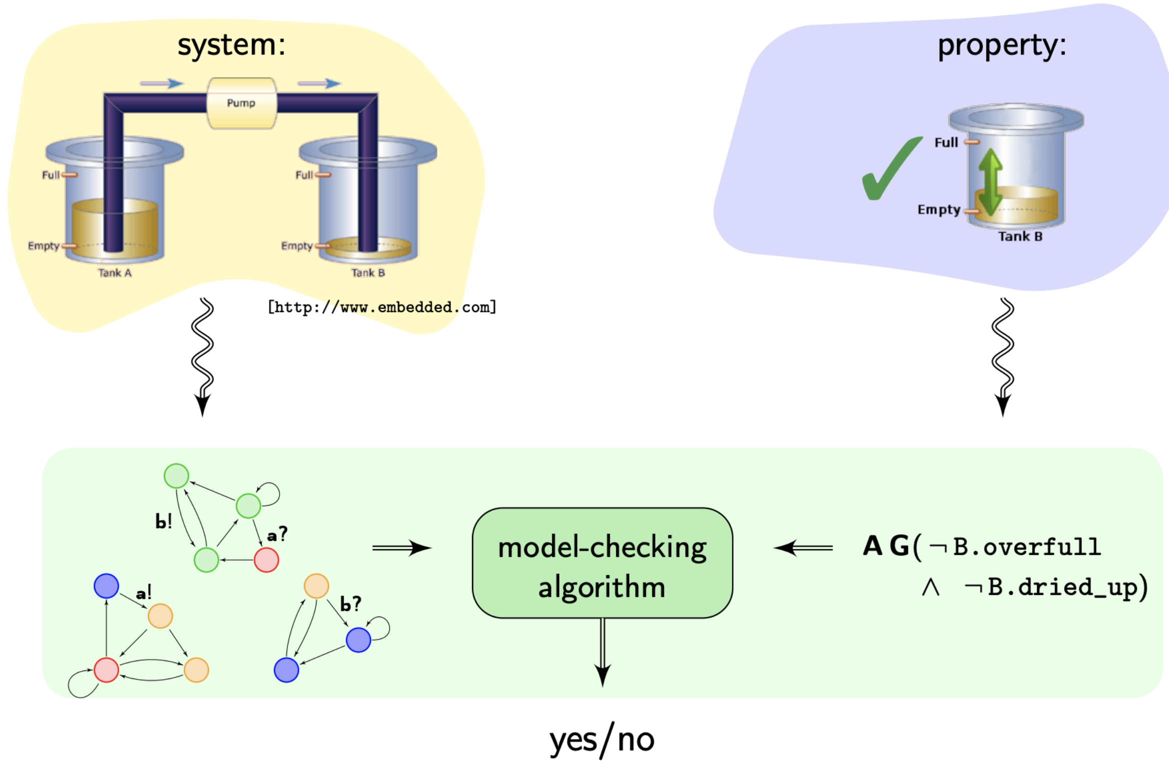
1. Temporal logics as specification languages



Motivations II: why do we care about logic?

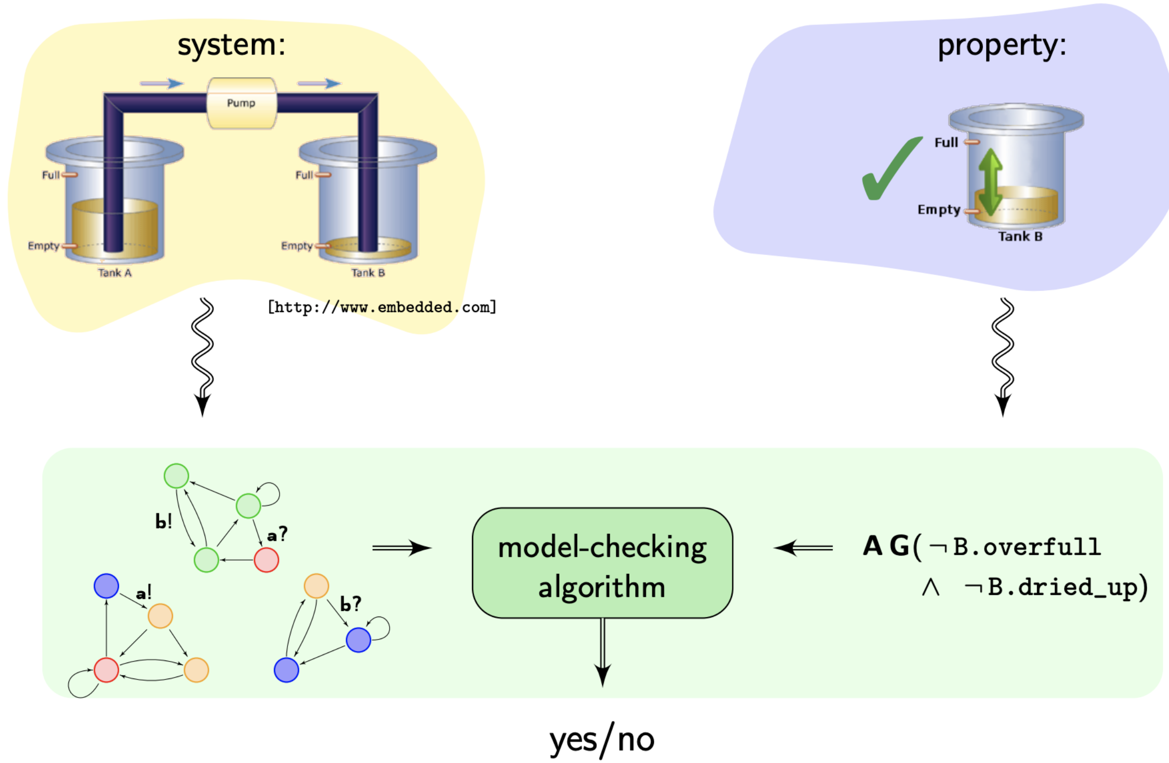
1. Temporal logics as specification languages

2. COQ: verified algorithms!, c.f. [here]



Motivations II: why do we care about logic?

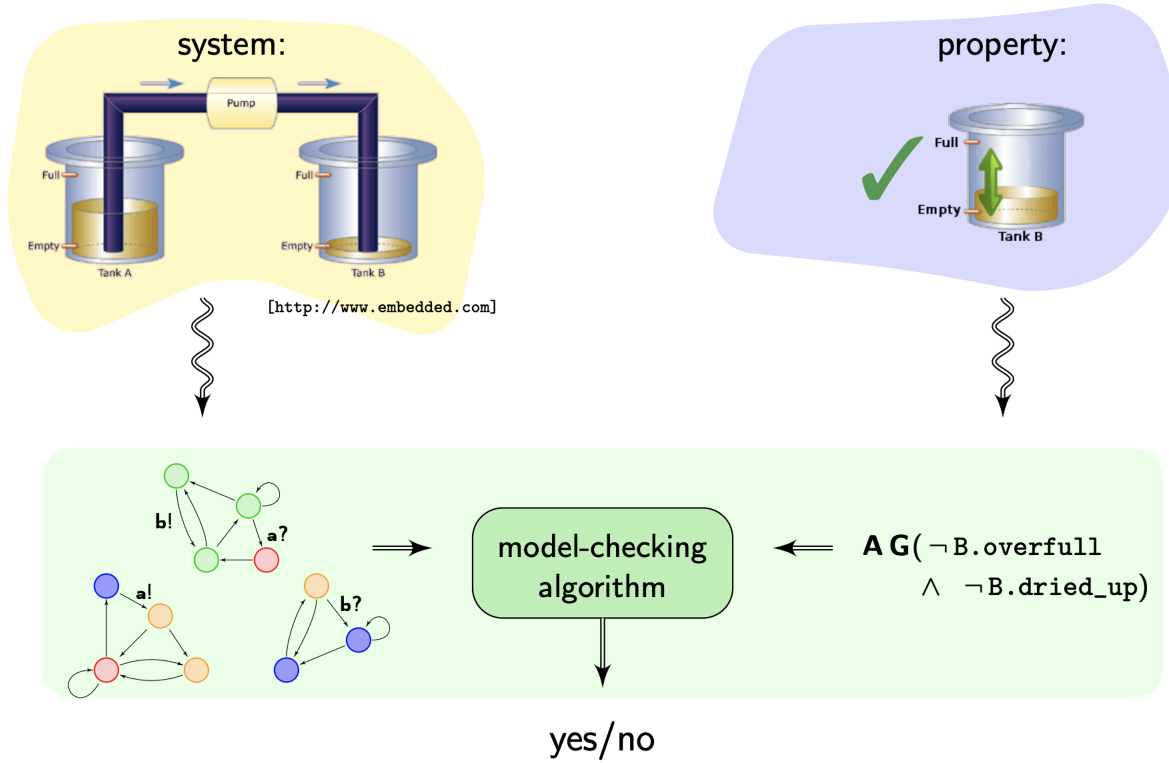
1. Temporal logics as specification languages
2. COQ: verified algorithms!, c.f. [here]
3. Separation logic: verifying Cpp/Java



Motivations II: why do we care about logic?

1. Temporal logics as specification languages
2. COQ: verified algorithms!, c.f. [here]
3. Separation logic: verifying Cpp/Java

Nice lecture [here].



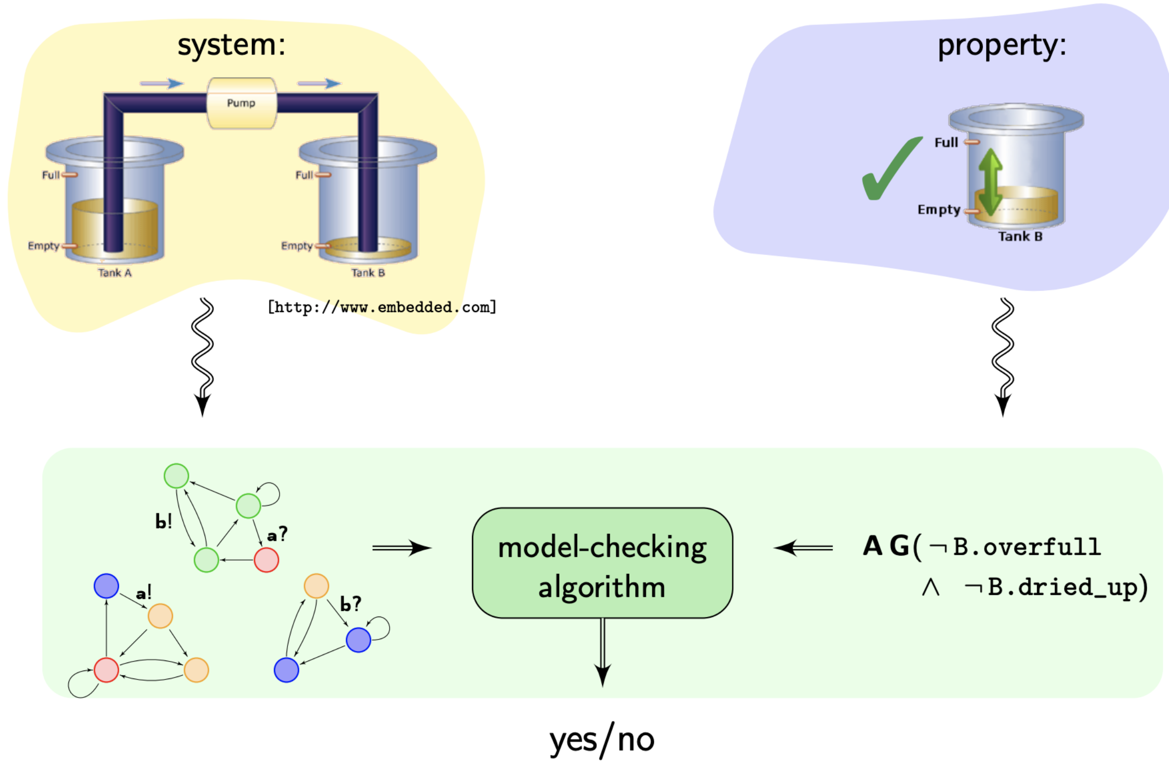
Motivations II: why do we care about logic?

1. Temporal logics as specification languages

2. COQ: verified algorithms!, c.f. [here]

3. Separation logic: verifying Cpp/Java

Nice lecture [here].(I'm there running with a mic!)

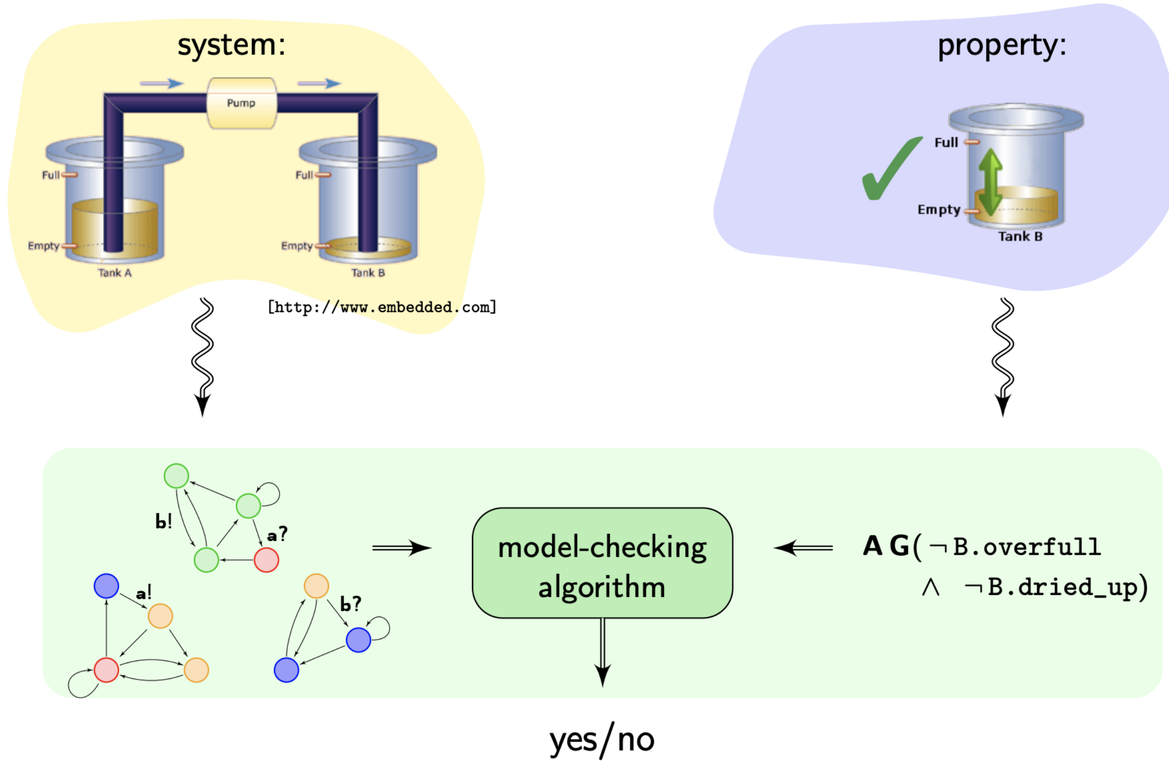


Motivations II: why do we care about logic?

1. Temporal logics as specification languages
2. COQ: verified algorithms!, c.f. [here]
3. Separation logic: verifying Cpp/Java

Nice lecture [here].(I'm there running with a mic!)

Check also Infer tool by Facebook!

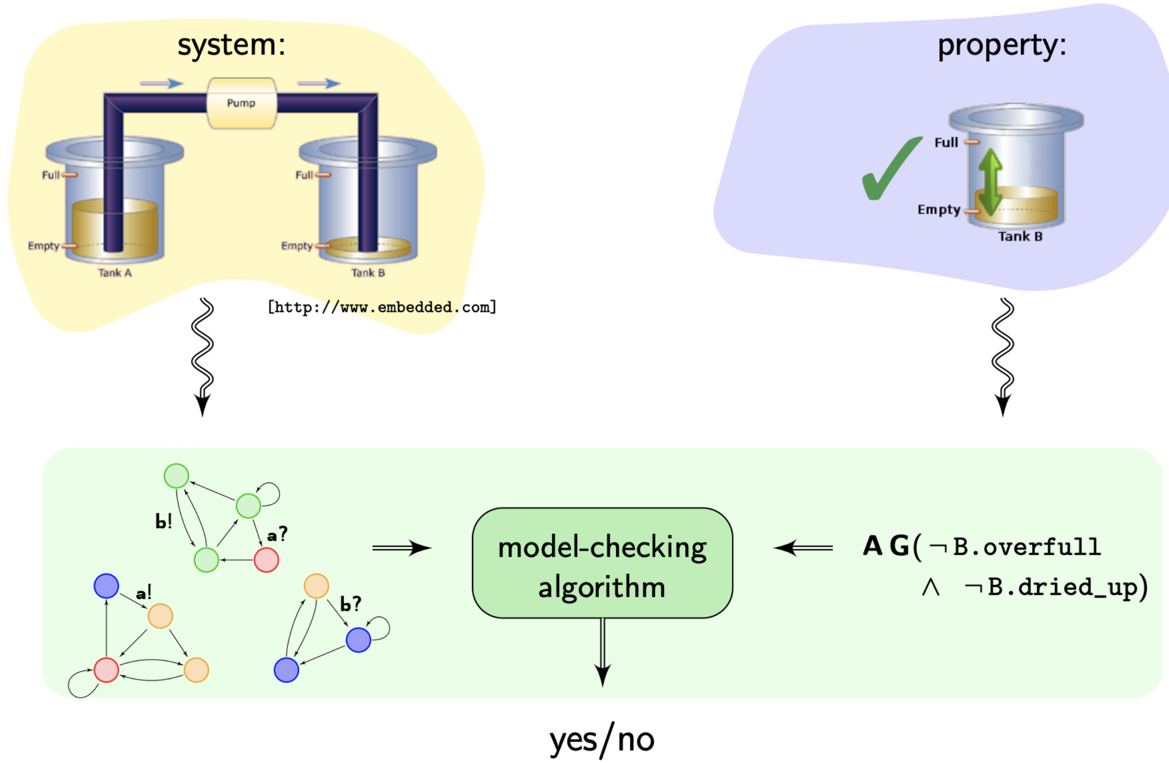


Motivations II: why do we care about logic?

1. Temporal logics as specification languages
2. COQ: verified algorithms!, c.f. [here]
3. Separation logic: verifying Cpp/Java

Nice lecture [here]. (I'm there running with a mic!)

Check also Infer tool by Facebook!

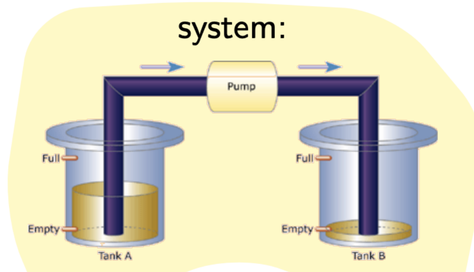


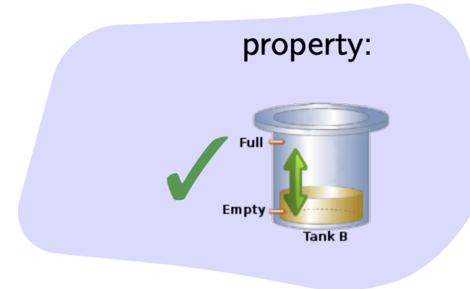
Motivations II: why do we care about logic?

1. Temporal logics as specification languages
2. COQ: verified algorithms!, c.f. [here]
3. Separation logic: verifying Cpp/Java

Nice lecture [here].(I'm there running with a mic!)

Check also Infer tool by Facebook!






```
vim hello.c
// hello.c
#include <stdlib.h>

void test() {
  int *s = NULL;
  *s = 42;
}
```

```
bartoszbednarczyk@Minsky-Machine: ~/Downloads/Infer
$ infer run -- gcc -c hello.c

Capturing in make/cc mode...
Found 1 source file to analyze in /Users/bartoszbednarczyk/Downloads/Infer/infer-out


Analysis finished in 775mss

Found 1 issue

hello.c:6: error: NULL_DEREFERENCE
  pointer `s` last assigned on line 5 could be null and is dereferenced at line 6, column 3.
  4.   void test() {
  5.     int *s = NULL;
  6. >   *s = 42;
  7.   }


Summary of the reports

  NULL_DEREFERENCE: 1
```

Motivations III: why do we care about logic?

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

$O(n)$ time

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

$\Theta(n \log(n))$ memory?

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

solvable in PSPACE?

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

decidable?

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Motivations III: why do we care about logic?

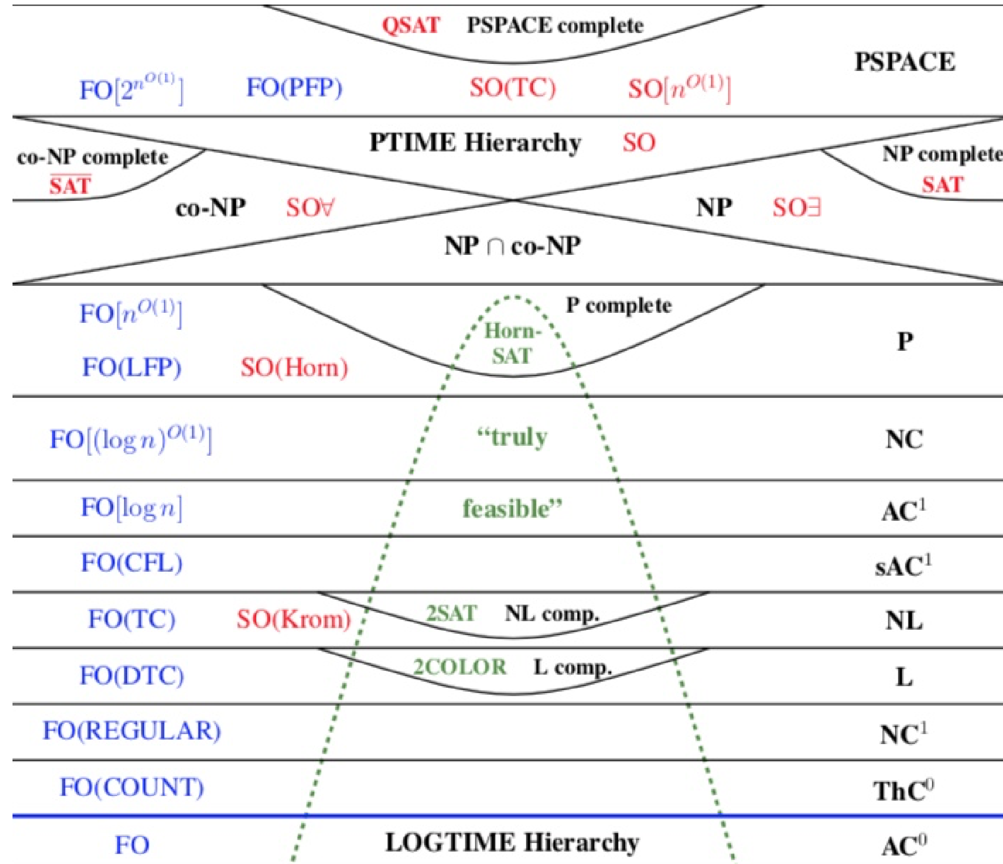
In "standard" computational complexity we measure resources, e.g. space and time.

Descriptive complexity: how strong the language must be to describe the problem?

Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Descriptive complexity: how strong the language must be to describe the problem?

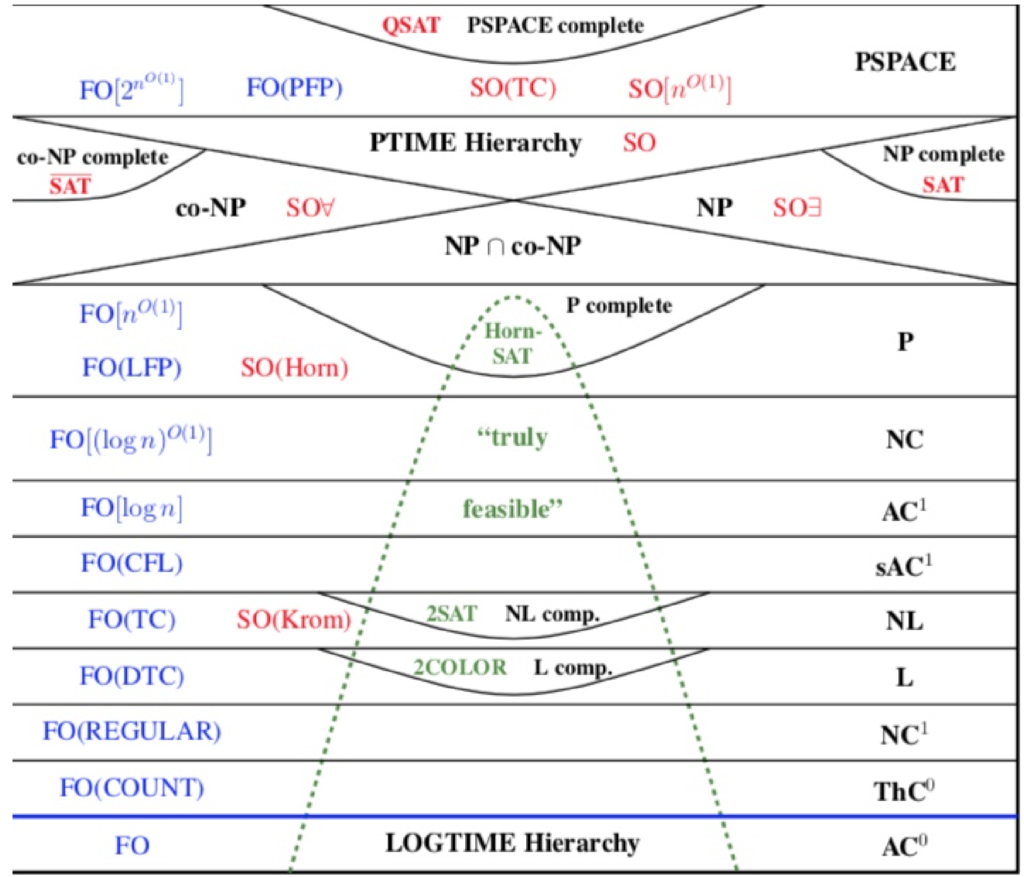


Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Descriptive complexity: how strong the language must be to describe the problem?

A logic $\mathcal{L}$ characterises the complexity class $\mathcal{C}$ if for every property of finite structures $\mathcal{P}$:



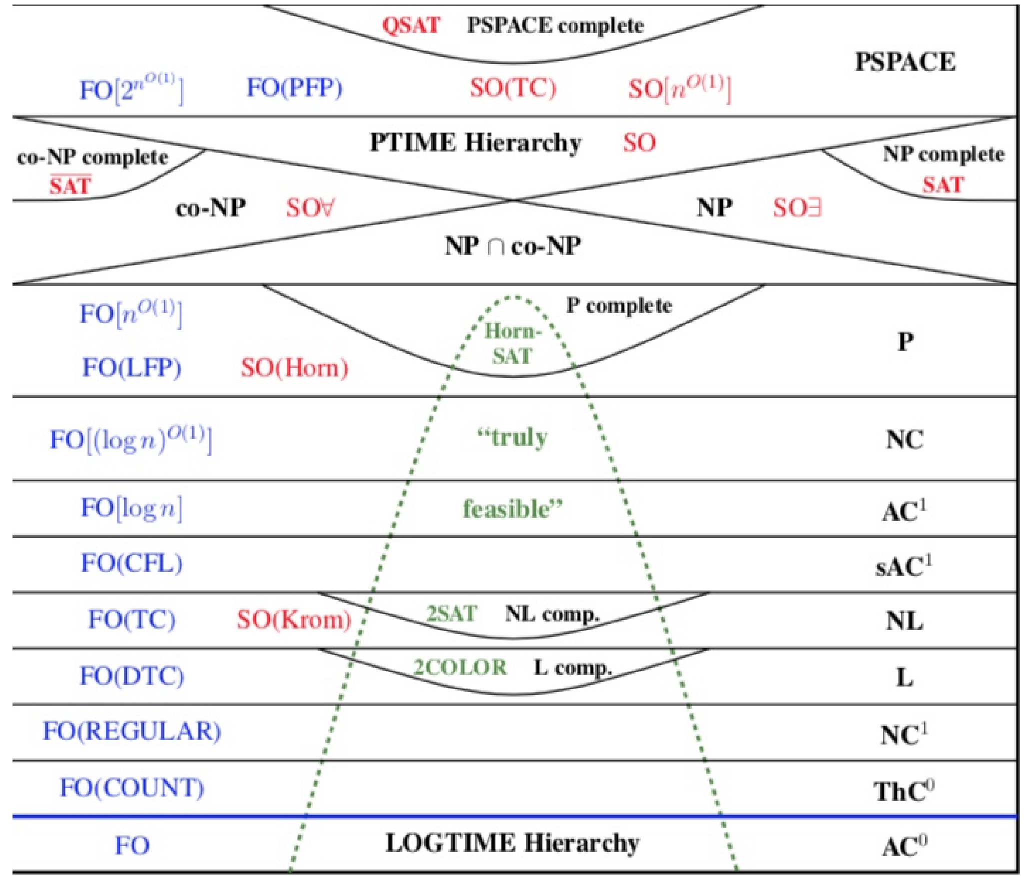
Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Descriptive complexity: how strong the language must be to describe the problem?

A logic $\mathcal{L}$ characterises the complexity class $\mathcal{C}$ if for every property of finite structures $\mathcal{P}$:

1. $\mathcal{P}$ is expressible in $\mathcal{L}$ if and only if



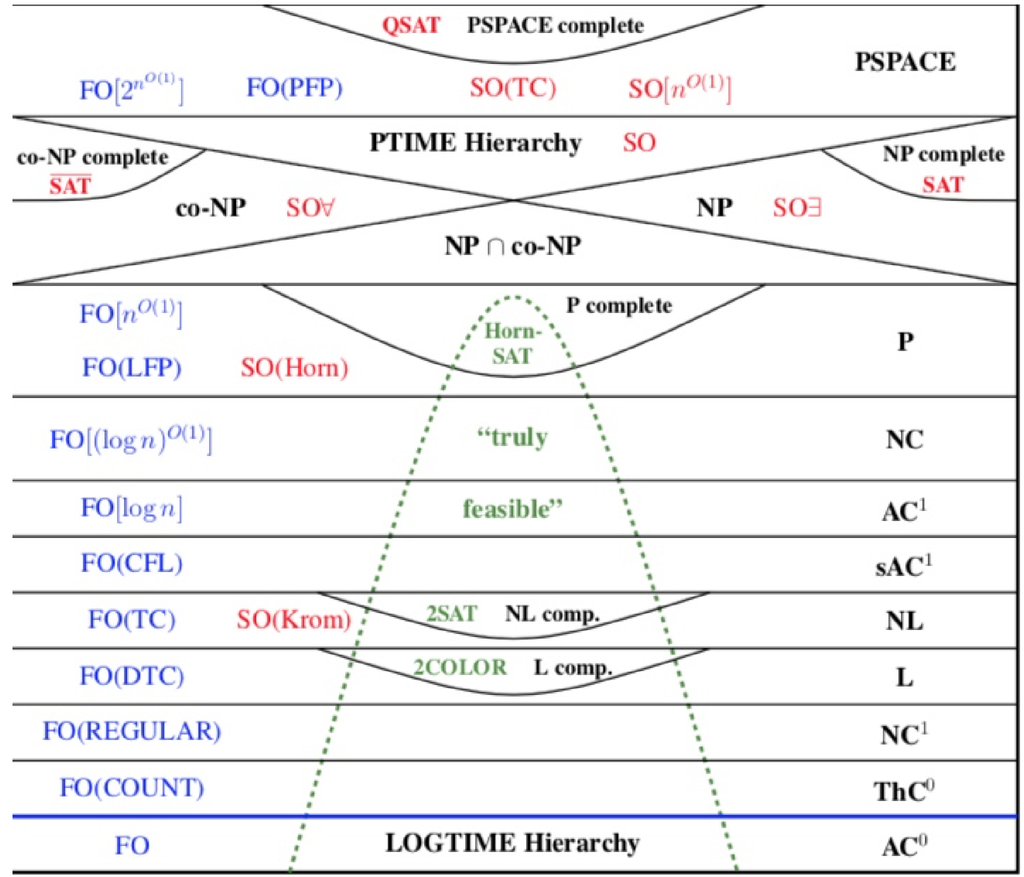
Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

Descriptive complexity: how strong the language must be to describe the problem?

A logic $\mathcal{L}$ characterises the complexity class $\mathcal{C}$ if for every property of finite structures $\mathcal{P}$:

1. $\mathcal{P}$ is expressible in $\mathcal{L}$ if and only if
2. There is an algorithm in $\mathcal{C}$ deciding $\mathcal{P}$.



Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

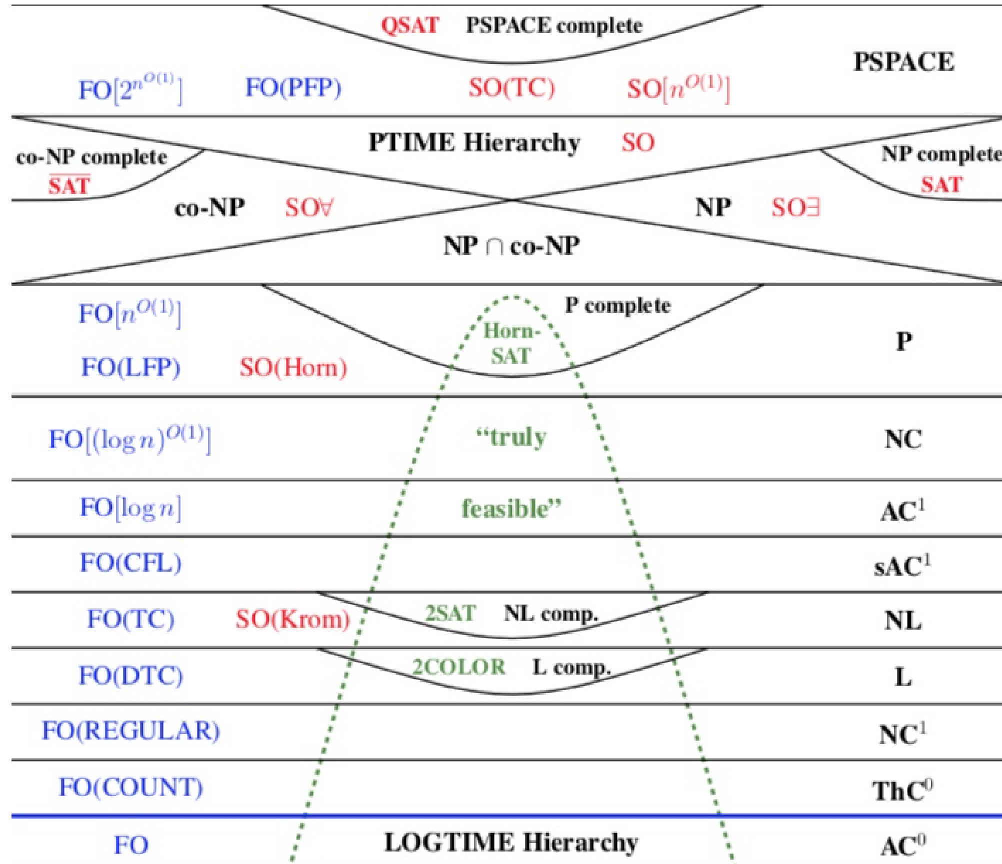
Descriptive complexity: how strong the language must be to describe the problem?

A logic $\mathcal{L}$ characterises the complexity class $\mathcal{C}$ if for every property of finite structures $\mathcal{P}$:

1. $\mathcal{P}$ is expressible in $\mathcal{L}$ if and only if
2. There is an algorithm in $\mathcal{C}$ deciding $\mathcal{P}$.

Theorem (Fagin'1973)

Existential Second Order Logic characterises NP.



Motivations III: why do we care about logic?

In "standard" computational complexity we measure resources, e.g. space and time.

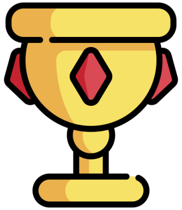
Descriptive complexity: how strong the language must be to describe the problem?

A logic $\mathcal{L}$ characterises the complexity class $\mathcal{C}$ if for every property of finite structures $\mathcal{P}$:

1. $\mathcal{P}$ is expressible in $\mathcal{L}$ if and only if
2. There is an algorithm in $\mathcal{C}$ deciding $\mathcal{P}$.

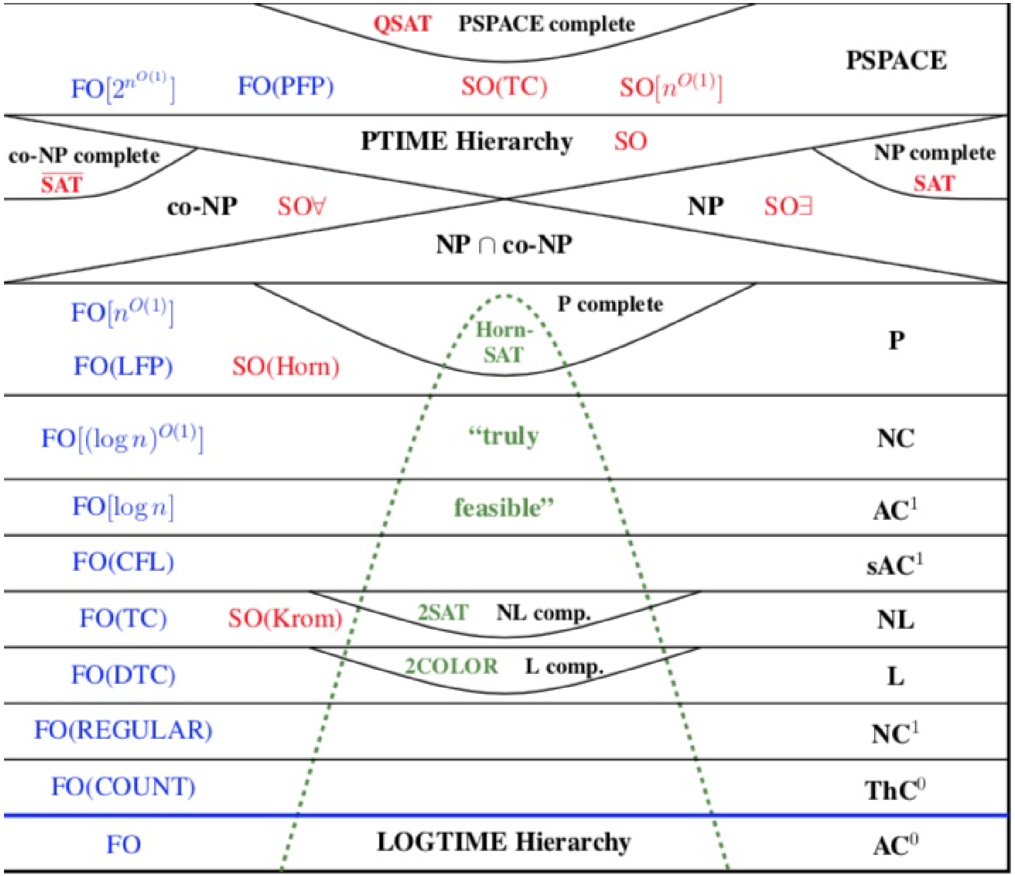
Theorem (Fagin'1973)

Existential Second Order Logic characterises NP.



Is there a logic for PTime?

No idea since 1988.



Motivations IV: why do we care about logic?

Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

Is every property of graphs expressible in FO is

Motivations IV: why do we care about logic?

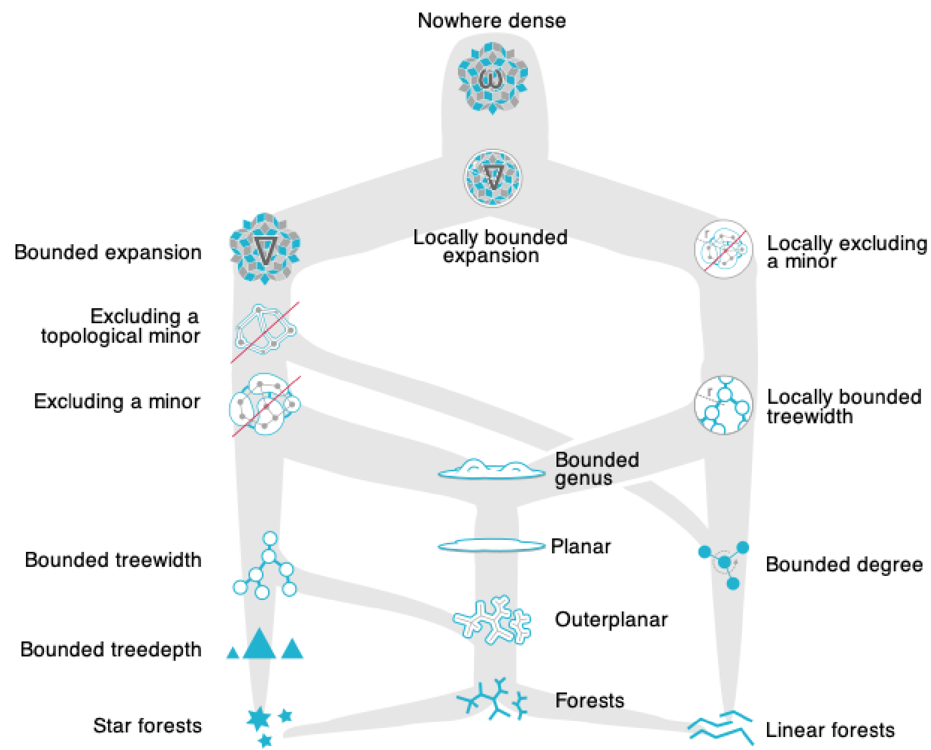
Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

> Is every property of graphs expressible in FO is checkable in linear time for all graphs from class $\mathcal{C}$?

Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

Is every property of graphs expressible in FO is checkable in linear time for all graphs from class $\mathcal{C}$?



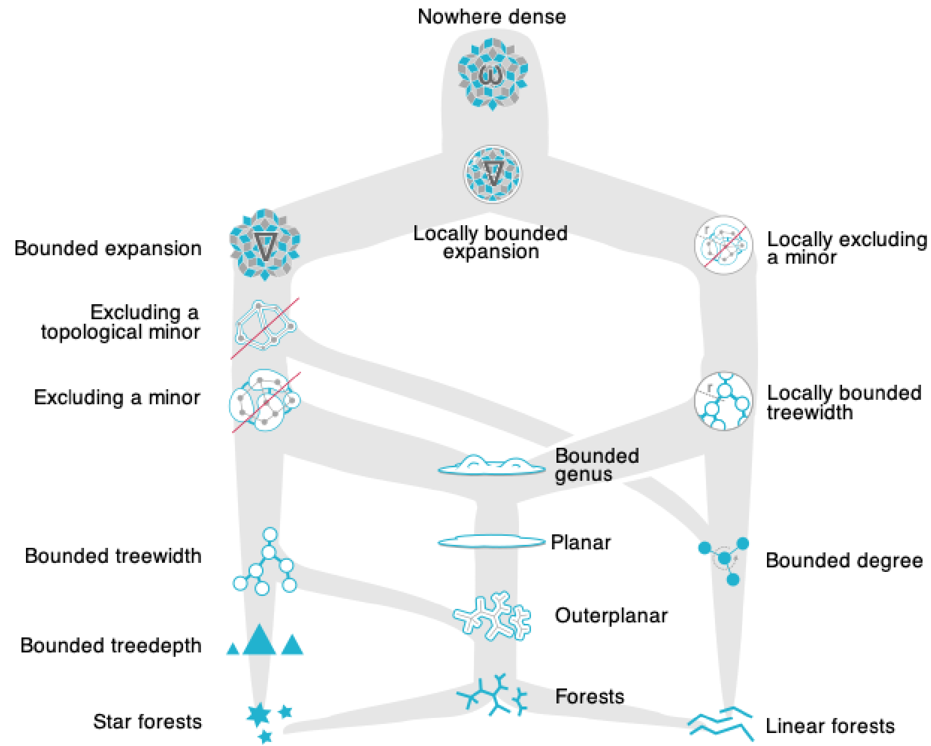
Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

Is every property of graphs expressible in FO is checkable in linear time for all graphs from class $\mathcal{C}$?

Theorem (Courcelle 1990)

$\mathcal{C}$:= graphs of bounded-treewidth.



Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

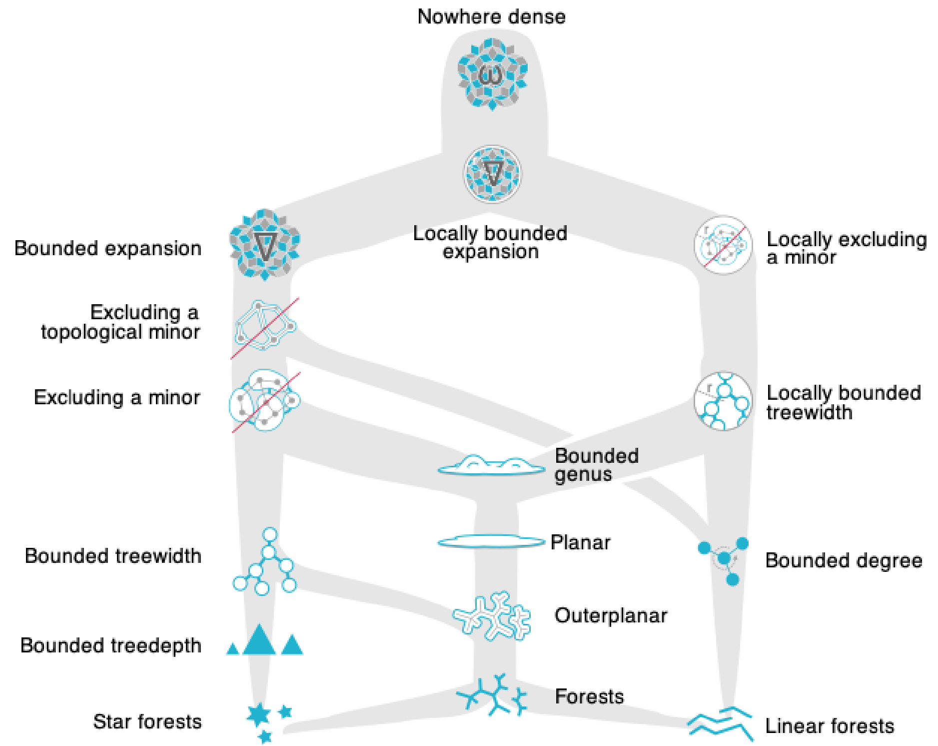
Is every property of graphs expressible in FO is checkable in linear time for all graphs from class $\mathcal{C}$?

Theorem (Courcelle 1990)

$\mathcal{C}$:= graphs of bounded-treewidth.

Theorem (Seese 1996)

$\mathcal{C}$:= graphs of bounded-degree.



Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

> Is every property of graphs expressible in FO is checkable in linear time for all graphs from class $\mathcal{C}$?

Theorem (Courcelle 1990)

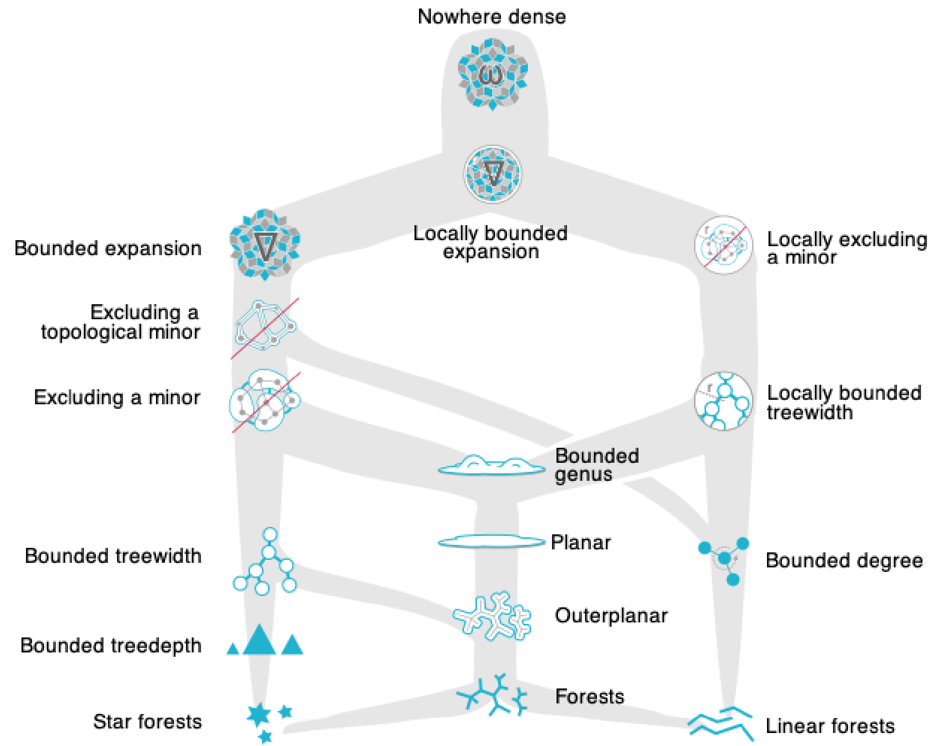
$\mathcal{C} :=$ graphs of bounded-treewidth.

Theorem (Seese 1996)

$\mathcal{C} :=$ graphs of bounded-degree.

Theorem (Dvorák et al. 2010)

$\mathcal{C} :=$ graphs of bounded-expansion.



Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

Is every property of graphs expressible in FO is checkable in linear time for all graphs from class $\mathcal{C}$?

Theorem (Courcelle 1990)

$\mathcal{C} :=$ graphs of bounded-treewidth.

Theorem (Seese 1996)

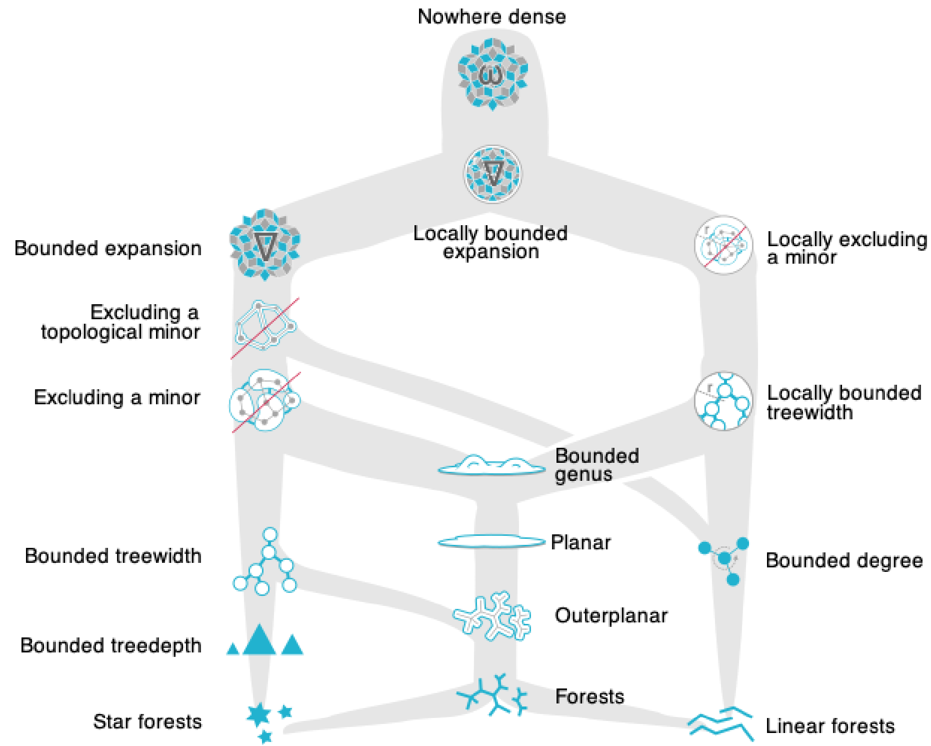
$\mathcal{C} :=$ graphs of bounded-degree.

Theorem (Dvorák et al. 2010)

$\mathcal{C} :=$ graphs of bounded-expansion.

Theorem (Bonnet et al. 2022)

$\mathcal{C} :=$ graphs of bounded-twinwidth.



Motivations IV: why do we care about logic?

Meta algorithms: say what you want instead of writing a code! Hot topic nowadays!

> Is every property of graphs expressible in FO is
> checkable in linear time for all graphs from class $\mathcal{C}$?

Theorem (Courcelle 1990)

$\mathcal{C} :=$ graphs of bounded-treewidth.

Theorem (Seese 1996)

$\mathcal{C} :=$ graphs of bounded-degree.

Theorem (Dvorák et al. 2010)

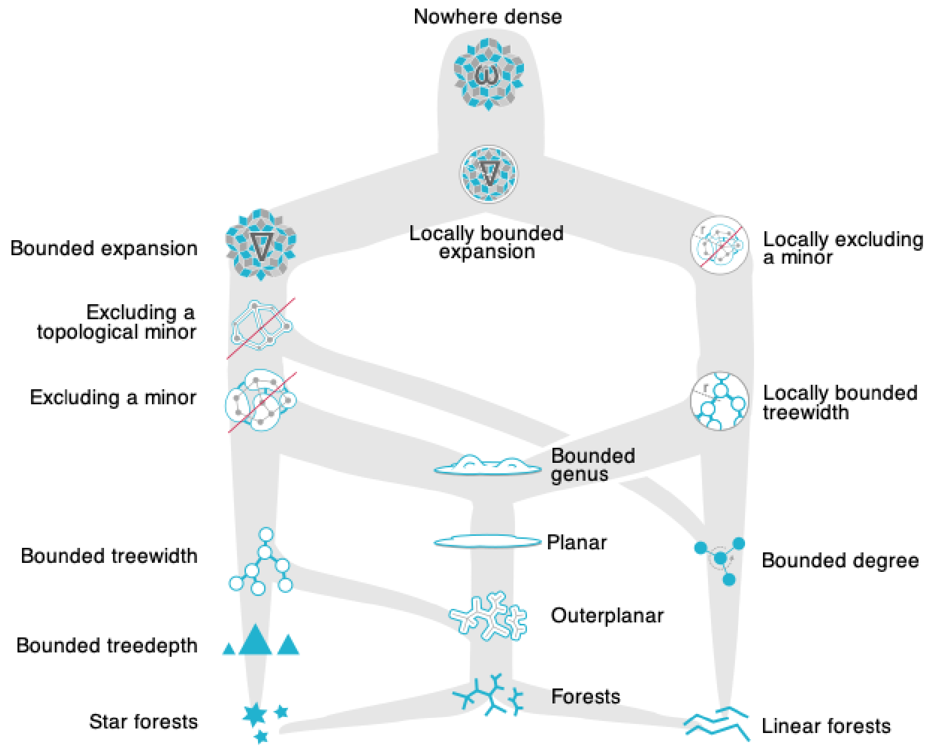
$\mathcal{C} :=$ graphs of bounded-expansion.

Theorem (Bonnet et al. 2022)

$\mathcal{C} :=$ graphs of bounded-twinwidth.

Theorem (Grohe, Kreutzer, Siebertz 2014)

$\mathcal{O}(|\varphi|^{1+\varepsilon})$ for $\mathcal{C} :=$ nowhere-dense graphs.



Signatures (vocabularies)

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \mathtt{Bartek}$ $\bullet\!\!\longrightarrow$

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$ •→

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ •→

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \mathtt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol $\mathtt{c}$, we have $\cdot^{\mathfrak{A}} : \mathtt{c} \mapsto (\mathtt{c}^{\mathfrak{A}} \in A)$

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

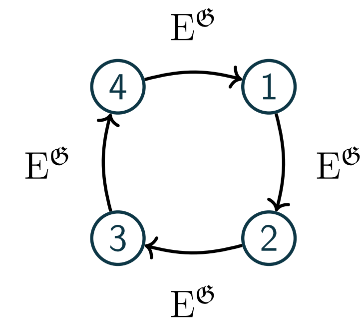
Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \mathtt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures



Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol $\mathtt{c}$, we have $\cdot^{\mathfrak{A}} : \mathtt{c} \mapsto (\mathtt{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$ •

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ • with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

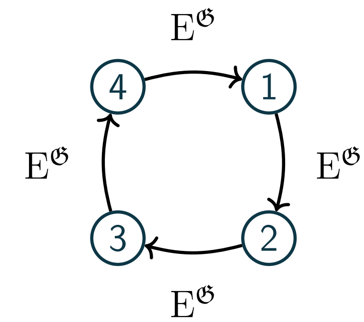
Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$



Morphisms

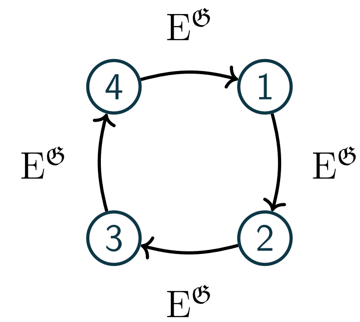
Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures



Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol $\mathtt{c}$, we have $\cdot^{\mathfrak{A}} : \mathtt{c} \mapsto (\mathtt{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures.

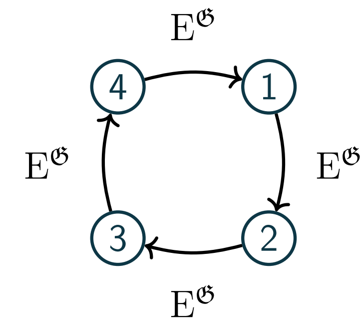
Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures



Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures. A homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is $\mathfrak{h} : A \to B$ satisfying:

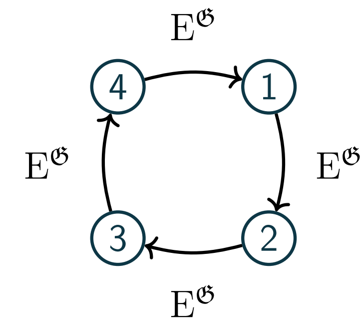
Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures



Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures. A homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is $\mathfrak{h} : A \to B$ satisfying:

- For all constant symbols $\mathsf{c} \in \sigma$ we have $\mathfrak{h}(\mathsf{c}^{\mathfrak{A}}) = \mathsf{c}^{\mathfrak{B}}$, and

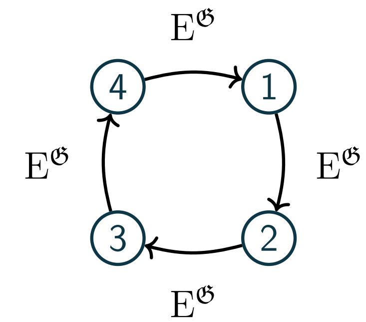
Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures



Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures. A homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is $\mathfrak{h} : A \to B$ satisfying:

- For all constant symbols $\mathsf{c} \in \sigma$ we have $\mathfrak{h}(\mathsf{c}^{\mathfrak{A}}) = \mathsf{c}^{\mathfrak{B}}$, and
- For all relational symbols $\mathrm{R} \in \sigma$, $\mathrm{R}^{\mathfrak{A}}(a_1, \ldots, a_{\mathsf{ar}(\mathrm{R})})$ implies $\mathrm{R}^{\mathfrak{B}}(\mathfrak{h}(a_1), \ldots, \mathfrak{h}(a_{\mathsf{ar}(\mathrm{R})}))$.

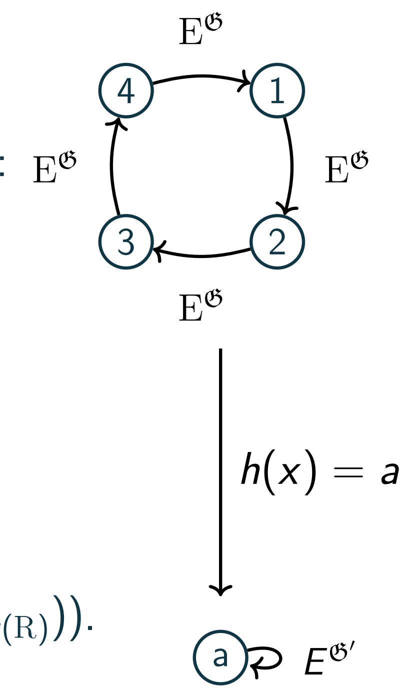
Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures



Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures. A homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is $\mathfrak{h} : A \to B$ satisfying:

- For all constant symbols $\mathsf{c} \in \sigma$ we have $\mathfrak{h}(\mathsf{c}^{\mathfrak{A}}) = \mathsf{c}^{\mathfrak{B}}$, and
- For all relational symbols $\mathrm{R} \in \sigma$, $\mathrm{R}^{\mathfrak{A}}(a_1, \ldots, a_{\mathsf{ar}(\mathrm{R})})$ implies $\mathrm{R}^{\mathfrak{B}}(\mathfrak{h}(a_1), \ldots, \mathfrak{h}(a_{\mathsf{ar}(\mathrm{R})}))$.

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

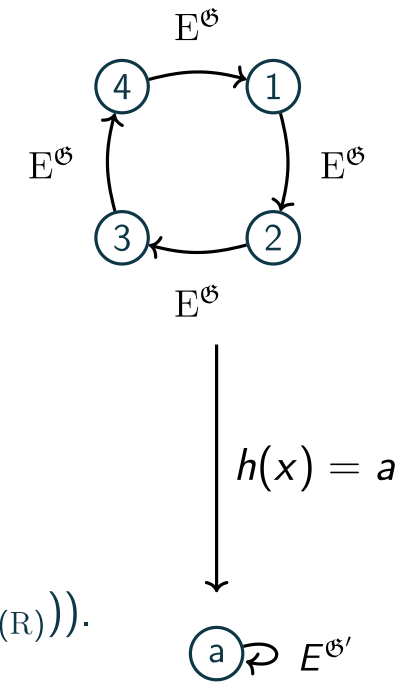
Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathrm{c} \mapsto (\mathrm{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$



Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures. A homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is $\mathfrak{h} : A \to B$ satisfying:

- For all constant symbols $\mathrm{c} \in \sigma$ we have $\mathfrak{h}(\mathrm{c}^{\mathfrak{A}}) = \mathrm{c}^{\mathfrak{B}}$, and
- For all relational symbols $\mathrm{R} \in \sigma$, $\mathrm{R}^{\mathfrak{A}}(a_1, \ldots, a_{\mathsf{ar}(\mathrm{R})})$ implies $\mathrm{R}^{\mathfrak{B}}(\mathfrak{h}(a_1), \ldots, \mathfrak{h}(a_{\mathsf{ar}(\mathrm{R})}))$.

An isomorphism $\mathfrak{h}$ between $\mathfrak{A}$ and $\mathfrak{B}$ is a bijection s.t. $\mathfrak{h}, \mathfrak{h}^{-1}$ are homomorphisms.

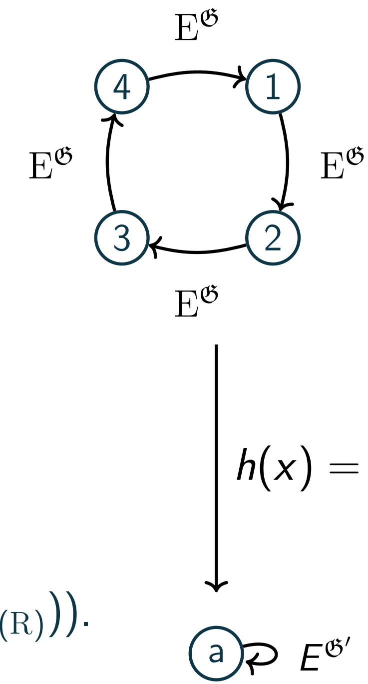
Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

Structures



Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$

Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures. A homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is $\mathfrak{h} : A \to B$ satisfying:

- For all constant symbols $\mathsf{c} \in \sigma$ we have $\mathfrak{h}(\mathsf{c}^{\mathfrak{A}}) = \mathsf{c}^{\mathfrak{B}}$, and
- For all relational symbols $\mathrm{R} \in \sigma$, $\mathrm{R}^{\mathfrak{A}}(a_1, \ldots, a_{\mathsf{ar}(\mathrm{R})})$ implies $\mathrm{R}^{\mathfrak{B}}(\mathfrak{h}(a_1), \ldots, \mathfrak{h}(a_{\mathsf{ar}(\mathrm{R})}))$.

An isomorphism $\mathfrak{h}$ between $\mathfrak{A}$ and $\mathfrak{B}$ is a bijection s.t. $\mathfrak{h}, \mathfrak{h}^{-1}$ are homomorphisms.

In this case we write: $\mathfrak{A} \cong \mathfrak{B}$.

Signatures (vocabularies)

Signature σ is a (countable) collection of symbols: $(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathrm{R}_1, \mathrm{R}_2, \ldots)$

Constant symbols, e.g. $\emptyset, 7, \texttt{Bartek}$

Relational symbols, e.g. $\in, \subseteq, \mathrm{isEven}$ with an associated arity, e.g. $\mathsf{ar}(\subseteq) = 2, \mathsf{ar}(\mathrm{isEven}) = 1$

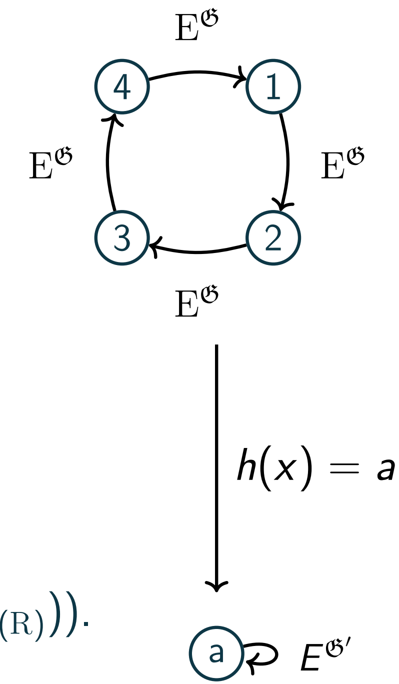
Structures

Over a signature σ we define σ-structures $\mathfrak{A} = (A, \cdot^{\mathfrak{A}})$ composed of:

- Non-empty set A called the domain of $\mathfrak{A}$ + Interpretation function $\cdot^{\mathfrak{A}}$ such that:

1. For each constant symbol c, we have $\cdot^{\mathfrak{A}} : \mathsf{c} \mapsto (\mathsf{c}^{\mathfrak{A}} \in A)$

2. For each relational symbol R, we have $\cdot^{\mathfrak{A}} : \mathrm{R} \mapsto (\mathrm{R}^{\mathfrak{A}} \subseteq A^{\mathsf{ar}(\mathrm{R})})$



Morphisms

Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures. A homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is $\mathfrak{h} : A \to B$ satisfying:

- For all constant symbols $\mathsf{c} \in \sigma$ we have $\mathfrak{h}(\mathsf{c}^{\mathfrak{A}}) = \mathsf{c}^{\mathfrak{B}}$, and
- For all relational symbols $\mathrm{R} \in \sigma$, $\mathrm{R}^{\mathfrak{A}}(a_1, \ldots, a_{\mathsf{ar}(\mathrm{R})})$ implies $\mathrm{R}^{\mathfrak{B}}(\mathfrak{h}(a_1), \ldots, \mathfrak{h}(a_{\mathsf{ar}(\mathrm{R})}))$.

An isomorphism $\mathfrak{h}$ between $\mathfrak{A}$ and $\mathfrak{B}$ is a bijection s.t. $\mathfrak{h}, \mathfrak{h}^{-1}$ are homomorphisms.

In this case we write: $\mathfrak{A} \cong \mathfrak{B}$. Important! $\mathfrak{A} \cong \mathfrak{B}$ implies $\mathfrak{A} \models \varphi \Leftrightarrow \mathfrak{B} \models \varphi$ for all formulae φ.

Syntax of FO[σ]

Syntax of FO[σ]

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

Syntax of FO[σ]

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set FO[σ] of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c}$ is a constant from $\sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set $\mathsf{FO}[\sigma]$ of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under

$\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Syntax of FO[σ]

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c}$ is a constant from $\sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set FO[σ] of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under $\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c}$ is a constant from $\sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set $\mathsf{FO}[\sigma]$ of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under $\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

$\exists x \ (E(x, y) \wedge \forall z \ (E(z, y) \rightarrow x = z))$

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set $\mathsf{FO}[\sigma]$ of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under

$\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

$\exists x \; (E(x, y) \wedge \forall z \, (E(z, y) \rightarrow x = z))$

$\exists x \; (E(x, y) \wedge \exists y \, \neg E(y, x))$

Syntax of FO[σ]

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is Terms(σ) $:= \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c}$ is a constant from $\sigma\}$.
- The set of atomic formulae Atoms(σ) is the smallest set such that:

1. If t_1, t_2 are terms from Terms(σ) then $t_1 = t_2$ belongs to Atoms(σ).

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in$ Terms(σ), and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in$ Atoms(σ).

- The set FO[σ] of First-Order formulae over σ is the closure of Atoms(σ) under

$\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

$\exists x\ (E(x, y) \wedge \forall z\, (E(z, y) \rightarrow x = z))$

$\exists x\ (E(x, y) \wedge \exists y\, \neg E(y, x))$

Formally, we define the set of free variables of φ, denoted with FVar(φ), as follows:

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set $\mathsf{FO}[\sigma]$ of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under $\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

$\exists x \; (E(x, y) \wedge \forall z \, (E(z, y) \rightarrow x = z))$

$\exists x \; (E(x, y) \wedge \exists y \, \neg E(y, x))$

Formally, we define the set of free variables of φ, denoted with $\mathsf{FVar}(\varphi)$, as follows:

- $\mathsf{FVar}(x) = \{x\}, \mathsf{FVar}(\mathsf{c}) = \emptyset$ for all $x \in \mathrm{Var}$ and constant symbols c from σ.

Syntax of $\mathrm{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathrm{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathrm{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathrm{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathrm{Atoms}(\sigma)$.
2. If $t_1, \ldots, t_{\mathrm{ar}(\mathrm{R})} \in \mathrm{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathrm{ar}(\mathrm{R})}) \in \mathrm{Atoms}(\sigma)$.

- The set $\mathrm{FO}[\sigma]$ of First-Order formulae over σ is the closure of $\mathrm{Atoms}(\sigma)$ under

$\land, \lor, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

$\exists x \ (E(x, y) \land \forall z \ (E(z, y) \rightarrow x = z))$ $\qquad$ $\exists x \ (E(x, y) \land \exists y \ \neg E(y, x))$

Formally, we define the set of free variables of φ, denoted with $\mathrm{FVar}(\varphi)$, as follows:

- $\mathrm{FVar}(x) = \{x\}, \mathrm{FVar}(\mathsf{c}) = \emptyset$ for all $x \in \mathrm{Var}$ and constant symbols c from σ.
- $\mathrm{FVar}(t_1 = t_2) = \mathrm{FVar}(t_1) \cup \mathrm{FVar}(t_2)$ for all $t_1, t_2 \in \mathrm{Terms}(\sigma)$.

Syntax of FO[σ]

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c}$ is a constant from $\sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set FO[σ] of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under $\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

$\exists x \ (E(x, y) \wedge \forall z \ (E(z, y) \rightarrow x = z))$

$\exists x \ (E(x, y) \wedge \exists y \ \neg E(y, x))$

Formally, we define the set of free variables of φ, denoted with $\mathsf{FVar}(\varphi)$, as follows:

- $\mathsf{FVar}(x) = \{x\}, \mathsf{FVar}(\mathsf{c}) = \emptyset$ for all $x \in \mathrm{Var}$ and constant symbols c from σ.
- $\mathsf{FVar}(t_1 = t_2) = \mathsf{FVar}(t_1) \cup \mathsf{FVar}(t_2)$ for all $t_1, t_2 \in \mathsf{Terms}(\sigma)$.
- $\mathsf{FVar}(\neg\varphi) = \mathsf{FVar}(\varphi)$ and $\mathsf{FVar}(\varphi \wedge \psi) = \mathsf{FVar}(\varphi) \cup \mathsf{FVar}(\psi)$. (and similarly for $\rightarrow, \leftrightarrow, \vee, \top, \bot$)

Syntax of $\mathsf{FO}[\sigma]$

- Let $\mathrm{Var} := \{x, y, z, u, v, \ldots\}$ be a countably-infinite set of variables.
- The set of terms is $\mathsf{Terms}(\sigma) := \mathrm{Var} \cup \{\mathsf{c} \mid \mathsf{c} \text{ is a constant from } \sigma\}$.
- The set of atomic formulae $\mathsf{Atoms}(\sigma)$ is the smallest set such that:

1. If t_1, t_2 are terms from $\mathsf{Terms}(\sigma)$ then $t_1 = t_2$ belongs to $\mathsf{Atoms}(\sigma)$.

2. If $t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})} \in \mathsf{Terms}(\sigma)$, and $\mathrm{R} \in \sigma$ is relational implies $\mathrm{R}(t_1, \ldots, t_{\mathsf{ar}(\mathrm{R})}) \in \mathsf{Atoms}(\sigma)$.

- The set $\mathsf{FO}[\sigma]$ of First-Order formulae over σ is the closure of $\mathsf{Atoms}(\sigma)$ under $\land, \lor, \to, \leftrightarrow, \neg, \exists x, \forall x$ (for all variables $x \in \mathrm{Var}$).

Free variables

$\exists x \; (E(x, y) \land \forall z \, (E(z, y) \to x = z))$ $\qquad$ $\exists x \; (E(x, y) \land \exists y \, \neg E(y, x))$

Formally, we define the set of free variables of φ, denoted with $\mathsf{FVar}(\varphi)$, as follows:

- $\mathsf{FVar}(x) = \{x\}, \mathsf{FVar}(\mathsf{c}) = \emptyset$ for all $x \in \mathrm{Var}$ and constant symbols c from σ.
- $\mathsf{FVar}(t_1 = t_2) = \mathsf{FVar}(t_1) \cup \mathsf{FVar}(t_2)$ for all $t_1, t_2 \in \mathsf{Terms}(\sigma)$.
- $\mathsf{FVar}(\neg\varphi) = \mathsf{FVar}(\varphi)$ and $\mathsf{FVar}(\varphi \land \psi) = \mathsf{FVar}(\varphi) \cup \mathsf{FVar}(\psi)$. (and similarly for $\to, \leftrightarrow, \lor, \top, \bot$)
- $\mathsf{FVar}(\exists x \; \varphi) = \mathsf{FVar}(\varphi) \setminus \{x\}$ for all $x \in \mathrm{Var}$.

Notation regarding formulae

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathsf{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathsf{c}^{\mathfrak{A}}$.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathsf{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathsf{c}^{\mathfrak{A}}$.

2. The value of x_i in $t^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)$ is a_i.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathsf{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathsf{c}^{\mathfrak{A}}$.

2. The value of x_i in $t^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)$ is a_i.

Now we define $\models$ for $\varphi(x_1, x_2, \ldots, x_n)$:

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathrm{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathrm{c}^{\mathfrak{A}}$.

2. The value of x_i in $t^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)$ is a_i.

Now we define $\models$ for $\varphi(x_1, x_2, \ldots, x_n)$:

- If $\varphi \equiv t_1 = t_2$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $t_1^{\mathfrak{A}}(\bar{a}) = t_2^{\mathfrak{A}}(\bar{a})$.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$
the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathrm{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathrm{c}^{\mathfrak{A}}$.

2. The value of x_i in $t^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)$ is a_i.

Now we define $\models$ for $\varphi(x_1, x_2, \ldots, x_n)$:

- If $\varphi \equiv t_1 = t_2$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $t_1^{\mathfrak{A}}(\bar{a}) = t_2^{\mathfrak{A}}(\bar{a})$.
- If $\varphi \equiv \mathrm{R}(t_1, t_2, \ldots, t_n)$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $(t_1^{\mathfrak{A}}(\bar{a}), \ldots, t_n^{\mathfrak{A}}(\bar{a}) \in \mathrm{R}^{\mathfrak{A}}$.

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathrm{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathrm{c}^{\mathfrak{A}}$.
2. The value of x_i in $t^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)$ is a_i.

Now we define $\models$ for $\varphi(x_1, x_2, \ldots, x_n)$:

- If $\varphi \equiv t_1 = t_2$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $t_1^{\mathfrak{A}}(\bar{a}) = t_2^{\mathfrak{A}}(\bar{a})$.
- If $\varphi \equiv \mathrm{R}(t_1, t_2, \ldots, t_n)$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $(t_1^{\mathfrak{A}}(\bar{a}), \ldots, t_n^{\mathfrak{A}}(\bar{a}) \in \mathrm{R}^{\mathfrak{A}}$.
- $\mathfrak{A} \models \neg\varphi$ iff not $\mathfrak{A} \models \varphi$;

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathrm{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathrm{c}^{\mathfrak{A}}$.

2. The value of x_i in $t^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)$ is a_i.

Now we define $\models$ for $\varphi(x_1, x_2, \ldots, x_n)$:

- If $\varphi \equiv t_1 = t_2$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $t_1^{\mathfrak{A}}(\bar{a}) = t_2^{\mathfrak{A}}(\bar{a})$.
- If $\varphi \equiv \mathrm{R}(t_1, t_2, \ldots, t_n)$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $(t_1^{\mathfrak{A}}(\bar{a}), \ldots, t_n^{\mathfrak{A}}(\bar{a}) \in \mathrm{R}^{\mathfrak{A}}$.
- $\mathfrak{A} \models \neg\varphi$ iff not $\mathfrak{A} \models \varphi$; $\quad \mathfrak{A} \models \varphi \wedge \psi$ iff $\mathfrak{A} \models \varphi$ and $\mathfrak{A} \models \psi$ (similarly for other connectives)

Notation regarding formulae

We write $\varphi(x_1, x_2, \ldots, x_k)$ to indicate that the variables $x_1, \ldots, x_k$ are free in φ.

Formula without free-variables is called a sentence.

Formula without occurrences of $\forall, \exists$ is called a quantifier-free.

A set of sentences is called a theory.

Semantics of FO

For a σ-structure $\mathfrak{A}$ we define inductively, for each term $t(x_1, x_2, \ldots, x_n)$

the value of $t^{\mathfrak{A}}(a_1, \ldots, a_n)$, where $(a_1, \ldots, a_n) \in A^n$ as follows:

1. For a constant symbol $\mathrm{c} \in \sigma$, the value of c in $\mathfrak{A}$ is $\mathrm{c}^{\mathfrak{A}}$.

2. The value of x_i in $t^{\mathfrak{A}}(a_1, a_2, \ldots, a_n)$ is a_i.

Now we define $\models$ for $\varphi(x_1, x_2, \ldots, x_n)$:

- If $\varphi \equiv t_1 = t_2$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $t_1^{\mathfrak{A}}(\bar{a}) = t_2^{\mathfrak{A}}(\bar{a})$.
- If $\varphi \equiv \mathrm{R}(t_1, t_2, \ldots, t_n)$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $(t_1^{\mathfrak{A}}(\bar{a}), \ldots, t_n^{\mathfrak{A}}(\bar{a}) \in \mathrm{R}^{\mathfrak{A}}$.
- $\mathfrak{A} \models \neg\varphi$ iff not $\mathfrak{A} \models \varphi$; $\quad \mathfrak{A} \models \varphi \wedge \psi$ iff $\mathfrak{A} \models \varphi$ and $\mathfrak{A} \models \psi$ (similarly for other connectives)
- If $\varphi \equiv \exists x\ \psi(x, \bar{y})$, then $\mathfrak{A} \models \varphi(\bar{a})$ iff $\mathfrak{A} \models \psi(a', \bar{a})$ for some $a' \in A$ (similarly for $\forall$ quantifier)

The last bunch of notations. Proof systems.

The last bunch of notations. Proof systems.

A formula φ is satisfiable

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$).

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ.

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

Löwenheim–Skolem 1922: If a countable $\mathcal{T}$ has a model then $\mathcal{T}$ has a countable one.

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

Löwenheim–Skolem 1922: If a countable $\mathcal{T}$ has a model then $\mathcal{T}$ has a countable one.

FO has dedicated proof systems, e.g. Gentzen's sequents.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)], Q(a) \vdash Q(a)} \quad \dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash P(a), Q(a)}}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vdash Q(a)}[\neg \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vee Q(a) \vdash Q(a)}[\vee \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], P(a) \to Q(a) \vdash Q(a)}[\to\vdash r.w.]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash Q(a)}[\forall \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash \forall x[Q(x)}[\vdash \forall]$$

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

Löwenheim–Skolem 1922: If a countable $\mathcal{T}$ has a model then $\mathcal{T}$ has a countable one.

FO has dedicated proof systems, e.g. Gentzen's sequents. Check Tim Lyon's lectures! [HERE]

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)], Q(a) \vdash Q(a)} \qquad \dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash P(a), Q(a)}}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vdash Q(a)}\,[\neg \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \lor Q(a) \vdash Q(a)}\,[\lor \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], P(a) \to Q(a) \vdash Q(a)}\,[\to\vdash r.w.]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash Q(a)}\,[\forall \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash \forall x[Q(x)}\,[\vdash \forall]
$$

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

Löwenheim–Skolem 1922: If a countable $\mathcal{T}$ has a model then $\mathcal{T}$ has a countable one.

FO has dedicated proof systems, e.g. Gentzen's sequents. Check Tim Lyon's lectures! [HERE]

$\mathcal{T} \vdash \varphi$ means φ is provable from $\mathcal{T}$ with sequents.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)], Q(a) \vdash Q(a)} \qquad \dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash P(a), Q(a)}}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vdash Q(a)}[\neg \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \lor Q(a) \vdash Q(a)}[\lor \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], P(a) \to Q(a) \vdash Q(a)}[\to\vdash r.w.]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash Q(a)}[\forall \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash \forall x[Q(x)}[\vdash \forall]$$

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

Löwenheim–Skolem 1922: If a countable $\mathcal{T}$ has a model then $\mathcal{T}$ has a countable one.

FO has dedicated proof systems, e.g. Gentzen's sequents. Check Tim Lyon's lectures! [HERE]

$\mathcal{T} \vdash \varphi$ means φ is provable from $\mathcal{T}$ with sequents.

(we treat $\mathcal{T}$ as extra axioms, note that proofs are finite)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)], Q(a) \vdash Q(a)} \qquad \dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash P(a), Q(a)}}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vdash Q(a)}[\neg \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \lor Q(a) \vdash Q(a)}[\lor \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], P(a) \to Q(a) \vdash Q(a)}[\to\vdash r.w.]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash Q(a)}[\forall \vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash \forall x[Q(x)}[\vdash \forall]$$

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

Löwenheim–Skolem 1922: If a countable $\mathcal{T}$ has a model then $\mathcal{T}$ has a countable one.

FO has dedicated proof systems, e.g. Gentzen's sequents. Check Tim Lyon's lectures! [HERE]

$\mathcal{T} \vdash \varphi$ means φ is provable from $\mathcal{T}$ with sequents.

(we treat $\mathcal{T}$ as extra axioms, note that proofs are finite)

Gödel 1929: $\mathcal{T} \models \varphi$ iff $\mathcal{T} \vdash \varphi$

$$\frac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)], Q(a) \vdash Q(a)} \qquad \dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash P(a), Q(a)}}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vdash Q(a)}\,[\neg \vdash]$$

$$\frac{\cdots}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \lor Q(a) \vdash Q(a)}\,[\lor \vdash]$$

$$\frac{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \lor Q(a) \vdash Q(a)}{\forall x[P(x)], \forall x[P(x) \to Q(x)], P(a) \to Q(a) \vdash Q(a)}\,[\to\vdash\ r.w.]$$

$$\frac{\forall x[P(x)], \forall x[P(x) \to Q(x)], P(a) \to Q(a) \vdash Q(a)}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash Q(a)}\,[\forall \vdash]$$

$$\frac{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash Q(a)}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash \forall x[Q(x)}\,[\vdash \forall]$$

The last bunch of notations. Proof systems.

A formula φ is satisfiable if it has a model (there is a structure $\mathfrak{A}$ s.t. $\mathfrak{A} \models \varphi$).

For a theory $\mathcal{T}$ (set of sentences) we write $\mathfrak{A} \models \mathcal{T}$ instead of $\mathfrak{A} \models \bigwedge_{\varphi \in \mathcal{T}} \varphi$.

φ is a tautology iff every structure satisfies φ (written: $\models \varphi$). Note: φ is a tautology iff $\neg\varphi$ is unsatisfiable.

We write $\mathcal{T} \models \varphi$ to say that every model of $\mathcal{T}$ is a model of φ. Note: $\mathcal{T} \models \bot$ iff $\mathcal{T}$ is unSAT.

Warning! Models can be of any size: finite, countably-infinite and larger!

Löwenheim–Skolem 1922: If a countable $\mathcal{T}$ has a model then $\mathcal{T}$ has a countable one.

FO has dedicated proof systems, e.g. Gentzen's sequents. Check Tim Lyon's lectures! [HERE]

$\mathcal{T} \vdash \varphi$ means φ is provable from $\mathcal{T}$ with sequents.

(we treat $\mathcal{T}$ as extra axioms, note that proofs are finite)

Gödel 1929: $\mathcal{T} \models \varphi$ iff $\mathcal{T} \vdash \varphi$

SAT for FO is Recursively Enumerable

$$\dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)], Q(a) \vdash Q(a)} \qquad \dfrac{\dfrac{Ax}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash P(a), Q(a)}}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vdash Q(a)}\,[\neg\vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vee Q(a) \vdash Q(a)}\,[\vee\vdash]$$

$$\dfrac{\dfrac{\dfrac{\forall x[P(x)], \forall x[P(x) \to Q(x)], \neg P(a) \vee Q(a) \vdash Q(a)}{\forall x[P(x)], \forall x[P(x) \to Q(x)], P(a) \to Q(a) \vdash Q(a)}\,[\to\vdash\ r.w.]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash Q(a)}\,[\forall\vdash]}{\forall x[P(x)], \forall x[P(x) \to Q(x)] \vdash \forall x[Q(x)}\,[\vdash\forall]$$

The Gödel's Compactness Theorem

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case:

Showing

inexpressivity

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case:

Showing

inexpressivity

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case:
Showing
inexpressivity

1st excursion: Proving (1)

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

1st excursion: Proving (1)

"$\models \; = \; \vdash$"

Assume $\mathcal{T} \models \varphi$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models \; = \; \vdash$”

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models \; = \; \vdash$”

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$.

So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case:
Showing
inexpressivity

"$\models \; = \; \vdash$"

Proofs are finite

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$.

So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

"$\models \; = \; \vdash$"

Proofs are finite

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models \; = \; \vdash$”

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

"$\models \; = \; \vdash$"

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models \; = \; \vdash$”

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$.

Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!).

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models \; = \; \vdash$”

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$.

Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about “$\models \; = \; \vdash$” again we are done.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models \; = \; \vdash$”

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about “$\models \; = \; \vdash$” again we are done.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case:
Showing
inexpressivity

“$\models\ =\ \vdash$”

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about “$\models\ =\ \vdash$” again we are done.

2nd excursion: Proving (2)

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

"$\models\ =\ \vdash$"

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about "$\models\ =\ \vdash$" again we are done.

Ad absurdum

2nd excursion: Proving (2)

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

"$\models \; = \; \vdash$"

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about "$\models \; = \; \vdash$" again we are done.

Ad absurdum

2nd excursion: Proving (2)

Towards a contradiction suppose $\mathcal{T}$ is unsatisfiable.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case:
Showing
inexpressivity

"$\models \; = \; \vdash$"

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about "$\models \; = \; \vdash$" again we are done.

Ad absurdum

$\mathcal{T}$ unSAT iff $\mathcal{T} \models \bot$

2nd excursion: Proving (2)

Towards a contradiction suppose $\mathcal{T}$ is unsatisfiable.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models = \vdash$”

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$.

Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about “$\models = \vdash$” again we are done.

Ad absurdum

$\mathcal{T}$ unSAT iff $\mathcal{T} \models \bot$

2nd excursion: Proving (2)

Towards a contradiction suppose $\mathcal{T}$ is unsatisfiable. So $\mathcal{T} \models \bot$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

"$\models \;=\; \vdash$"

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about "$\models \;=\; \vdash$" again we are done.

Ad absurdum

$\mathcal{T}$ unSAT iff $\mathcal{T} \models \bot$

Employ (1)

2nd excursion: Proving (2)

Towards a contradiction suppose $\mathcal{T}$ is unsatisfiable. So $\mathcal{T} \models \bot$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.
2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

“$\models\ =\ \vdash$”

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about “$\models\ =\ \vdash$” again we are done.

Ad absurdum

$\mathcal{T}$ unSAT iff $\mathcal{T} \models \bot$

Employ (1)

2nd excursion: Proving (2)

Towards a contradiction suppose $\mathcal{T}$ is unsatisfiable. So $\mathcal{T} \models \bot$. By (1) there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ s.t. $\mathcal{T}_0 \models \bot$.

The Gödel's Compactness Theorem

Let $\mathcal{T}$ be an FO-theory and let φ be an FO sentence.

1. If $\mathcal{T} \models \varphi$ then there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ such that $\mathcal{T}_0 \models \varphi$.

2. If every *finite* $\mathcal{T}_0 \subseteq \mathcal{T}$ is satisfiable then $\mathcal{T}$ is satisfiable.

Use case: Showing inexpressivity

"$\models \; = \; \vdash$"

Proofs are finite

Craft $\mathcal{T}_0$

1st excursion: Proving (1)

Assume $\mathcal{T} \models \varphi$. Then by Gödel's completeness theorem $\mathcal{T} \vdash \varphi$. So there is a formal proof $\mathcal{P}$ of $\mathcal{T} \vdash \varphi$. Since proofs are finite the proof $\mathcal{P}$ uses only finitely many axioms of $\mathcal{T}$. Call them $\mathcal{T}_0$. Thus $\mathcal{T}_0 \vdash \varphi$ holds (use the same proof as before!). After asking Gödel about "$\models \; = \; \vdash$" again we are done.

Ad absurdum

$\mathcal{T}$ unSAT iff $\mathcal{T} \models \bot$

Employ (1)

2nd excursion: Proving (2)

Towards a contradiction suppose $\mathcal{T}$ is unsatisfiable. So $\mathcal{T} \models \bot$. By (1) there is a finite $\mathcal{T}_0 \subseteq \mathcal{T}$ s.t. $\mathcal{T}_0 \models \bot$. A contradiction!

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$.

Employing compactness I: Reachability in $\{E\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable.

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO[$\{\mathrm{E}\}$] formula for connectivity over $\{\mathrm{E}\}$-structures.

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO$[\{\mathrm{E}\}]$ formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO[$\{\mathrm{E}\}$] formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Assume that there is such φ, and let $\mathcal{T}$ be

Employing compactness I: Reachability in $\{E\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO[$\{E\}$] formula for connectivity over $\{E\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{E\}$-path.

Proof:

Assume that there is such φ, and let $\mathcal{T}$ be

Employ reachability!

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO[$\{\mathrm{E}\}$] formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Assume that there is such φ, and let $\mathcal{T}$ be

Employ reachability!

$\varphi_0^{\mathsf{reach(a,b)}} := \mathsf{a} = \mathsf{b}$, $\varphi_1^{\mathsf{reach(a,b)}} := \mathrm{E}(\mathsf{a}, \mathsf{b})$, $\varphi_k^{\mathsf{reach(a,b)}} :=$

$$\exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathsf{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathsf{b})$$

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO$[\{\mathrm{E}\}]$ formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Assume that there is such φ, and let $\mathcal{T}$ be

$$\mathcal{T} := \{\varphi\} \cup \{\neg\varphi_k^{\mathsf{reach(a,b)}} \mid k \geq 0\}.$$

Employ reachability!

$\varphi_0^{\mathsf{reach(a,b)}} := \mathsf{a} = \mathsf{b}$, $\varphi_1^{\mathsf{reach(a,b)}} := \mathrm{E}(\mathsf{a},\mathsf{b})$, $\varphi_k^{\mathsf{reach(a,b)}} :=$

$\exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathsf{a}, x_1) \wedge \bigwedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathsf{b})$

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO[$\{\mathrm{E}\}$] formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Assume that there is such φ, and let $\mathcal{T}$ be

$$\mathcal{T} := \{\varphi\} \cup \{\neg\varphi_k^{\mathrm{reach(a,b)}} \mid k \geq 0\}.$$

Since a and b are disconnected, $\mathcal{T}$ is unSAT.

Employ reachability!

$\varphi_0^{\mathrm{reach(a,b)}} := \mathrm{a} = \mathrm{b}$, $\varphi_1^{\mathrm{reach(a,b)}} := \mathrm{E}(\mathrm{a},\mathrm{b})$, $\varphi_k^{\mathrm{reach(a,b)}} :=$

$\exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathrm{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathrm{b})$

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$ $\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO$[\{\mathrm{E}\}]$ formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Assume that there is such φ, and let $\mathcal{T}$ be

$$\mathcal{T} := \{\varphi\} \cup \{\neg\varphi_k^{\mathsf{reach(a,b)}} \mid k \geq 0\}.$$

Since a and b are disconnected, $\mathcal{T}$ is unSAT.

Let $\mathcal{T}_0$ be any non-empty finite subset of $\mathcal{T}$.

Let N be max such that $\neg\varphi_N^{\mathsf{reach(a,b)}}$ is in $\mathcal{T}_0$. Then:

Employ reachability!

$\varphi_0^{\mathsf{reach(a,b)}} := \mathsf{a} = \mathsf{b}$, $\varphi_1^{\mathsf{reach(a,b)}} := \mathrm{E}(\mathsf{a},\mathsf{b})$, $\varphi_k^{\mathsf{reach(a,b)}} :=$ $\exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathsf{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathsf{b})$

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO[$\{\mathrm{E}\}$] formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Employ reachability!

Assume that there is such φ, and let $\mathcal{T}$ be

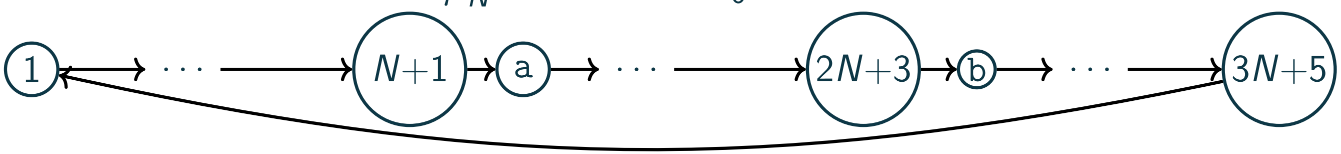
$$\mathcal{T} := \{\varphi\} \cup \{\neg\varphi_k^{\mathrm{reach(a,b)}} \mid k \geq 0\}.$$

$\varphi_0^{\mathrm{reach(a,b)}} := \mathrm{a} = \mathrm{b}$, $\varphi_1^{\mathrm{reach(a,b)}} := \mathrm{E}(\mathrm{a},\mathrm{b})$, $\varphi_k^{\mathrm{reach(a,b)}} :=$ $\exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathrm{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathrm{b})$

Since a and b are disconnected, $\mathcal{T}$ is unSAT.

Let $\mathcal{T}_0$ be any non-empty finite subset of $\mathcal{T}$.

Let N be max such that $\neg\varphi_N^{\mathrm{reach(a,b)}}$ is in $\mathcal{T}_0$. Then:



Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO[$\{\mathrm{E}\}$] formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

Proof:

Employ reachability!

Assume that there is such φ, and let $\mathcal{T}$ be

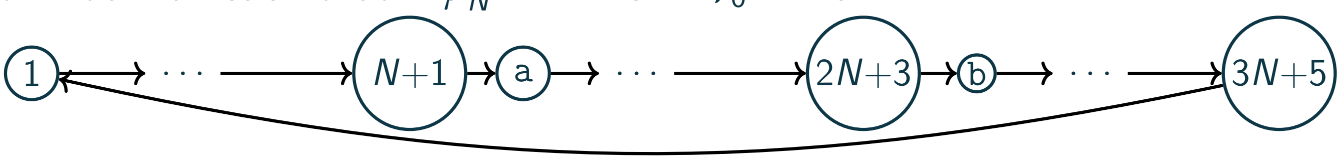
$$\mathcal{T} := \{\varphi\} \cup \{\neg\varphi_k^{\mathrm{reach(a,b)}} \mid k \geq 0\}.$$

Since a and b are disconnected, $\mathcal{T}$ is unSAT.

Let $\mathcal{T}_0$ be any non-empty finite subset of $\mathcal{T}$.

Let N be max such that $\neg\varphi_N^{\mathrm{reach(a,b)}}$ is in $\mathcal{T}_0$. Then:

$\varphi_0^{\mathrm{reach(a,b)}} := \mathrm{a} = \mathrm{b}$, $\varphi_1^{\mathrm{reach(a,b)}} := \mathrm{E(a,b)}$, $\varphi_k^{\mathrm{reach(a,b)}} :=$
$\exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathrm{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathrm{b})$



$\models \mathcal{T}_0$. A contradiction!

Employing compactness I: Reachability in $\{\mathrm{E}\}$-structures

The general proof scheme to show that the property $\mathcal{P}$ is not FO-definable.

Ad absurdum suppose that φ defines $\mathcal{P}$. $\rightsquigarrow$ Manufacture a theory $\mathcal{T}$ containing φ. $\rightsquigarrow$

$\rightsquigarrow$ Prove that $\mathcal{T}$ is unsatisfiable $\rightsquigarrow$ but its every finite subset is satisfiable. $\rightsquigarrow$ Contradict Compactness.

There is no FO$[\{\mathrm{E}\}]$ formula for connectivity over $\{\mathrm{E}\}$-structures.

So there is no formula saying that between any two nodes there is a directed $\{\mathrm{E}\}$-path.

No info about the finite models!

Proof:

Employ reachability!

Assume that there is such φ, and let $\mathcal{T}$ be

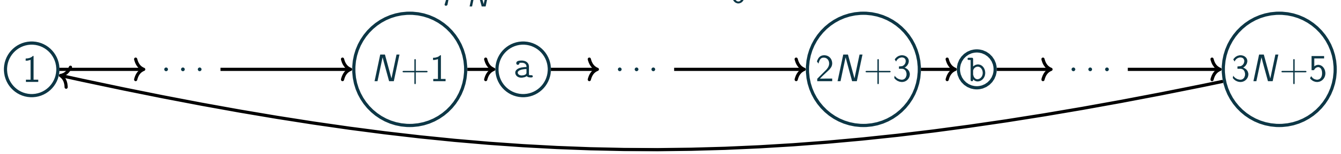
$$\mathcal{T} := \{\varphi\} \cup \{\neg\varphi_k^{\mathrm{reach(a,b)}} \mid k \geq 0\}.$$

Since a and b are disconnected, $\mathcal{T}$ is unSAT.

$\varphi_0^{\mathrm{reach(a,b)}} := \mathrm{a} = \mathrm{b}$, $\varphi_1^{\mathrm{reach(a,b)}} := \mathrm{E}(\mathrm{a},\mathrm{b})$, $\varphi_k^{\mathrm{reach(a,b)}} :=$ $\exists x_1 \ldots \exists x_{k-1}\, \mathrm{E}(\mathrm{a}, x_1) \wedge \wedge_{i=1}^{k-2} \mathrm{E}(x_i, x_{i+1}) \wedge \mathrm{E}(x_{k-1}, \mathrm{b})$

Let $\mathcal{T}_0$ be any non-empty finite subset of $\mathcal{T}$.

Let N be max such that $\neg\varphi_N^{\mathrm{reach(a,b)}}$ is in $\mathcal{T}_0$. Then:



$\models \mathcal{T}_0$. A contradiction!

Employing compactness II: Parity of the domain

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise).

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Exploit ∞!

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}$, $\quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}$.

Let λ_k say "there are $\geq k$ elem.".

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}$, $\quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}$.

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

Exploit ∞!

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

Let λ_k say "there are $\geq k$ elem.".

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Löwenheim–Skolem!

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Thus, by Löwenheim–Skolem, $\mathcal{T}_1, \mathcal{T}_2$ have countably-inf models $\mathfrak{A}$ and $\mathfrak{B}$.

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

Löwenheim–Skolem!

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Löwenheim–Skolem!

Thus, by Löwenheim–Skolem, $\mathcal{T}_1, \mathcal{T}_2$ have countably-inf models $\mathfrak{A}$ and $\mathfrak{B}$.

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$, and $\mathfrak{A} \models \mathcal{T}_2$ we get $\mathfrak{B} \models \neg\varphi$.

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

Exploit ∞!

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

Let λ_k say "there are $\geq k$ elem.".

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Löwenheim–Skolem!

Thus, by Löwenheim–Skolem, $\mathcal{T}_1, \mathcal{T}_2$ have countably-inf models $\mathfrak{A}$ and $\mathfrak{B}$.

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$, and $\mathfrak{A} \models \mathcal{T}_2$ we get $\mathfrak{B} \models \neg\varphi$.

$\emptyset$-structures $=$ sets

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Thus, by Löwenheim–Skolem, $\mathcal{T}_1, \mathcal{T}_2$ have countably-inf models $\mathfrak{A}$ and $\mathfrak{B}$.

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$, and $\mathfrak{A} \models \mathcal{T}_2$ we get $\mathfrak{B} \models \neg\varphi$.

As there is a bijection between any two countably-inf sets, we get $\mathfrak{A} \cong \mathfrak{B}$.

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

Löwenheim–Skolem!

$\emptyset$-structures $=$ sets

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

Exploit ∞!

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

Let λ_k say "there are $\geq k$ elem.".

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Löwenheim–Skolem!

Thus, by Löwenheim–Skolem, $\mathcal{T}_1, \mathcal{T}_2$ have countably-inf models $\mathfrak{A}$ and $\mathfrak{B}$.

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$, and $\mathfrak{A} \models \mathcal{T}_2$ we get $\mathfrak{B} \models \neg\varphi$.

As there is a bijection between any two countably-inf sets, we get $\mathfrak{A} \cong \mathfrak{B}$.

$\emptyset$-structures $=$ sets

Formulae are preserved by isomorphisms, so $\mathfrak{B} \models \neg\varphi$ implies $\mathfrak{A} \models \neg\varphi$:

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Thus, by Löwenheim–Skolem, $\mathcal{T}_1, \mathcal{T}_2$ have countably-inf models $\mathfrak{A}$ and $\mathfrak{B}$.

Löwenheim–Skolem!

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$, and $\mathfrak{A} \models \mathcal{T}_2$ we get $\mathfrak{B} \models \neg\varphi$.

As there is a bijection between any two countably-inf sets, we get $\mathfrak{A} \cong \mathfrak{B}$.

Formulae are preserved by isomorphisms, so $\mathfrak{B} \models \neg\varphi$ implies $\mathfrak{A} \models \neg\varphi$:

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$.

$\emptyset$-structures $=$ sets

Employing compactness II: Parity of the domain

The previous proof does not give us any information about the finite domain reasoning.

Even worse, Compactness fails in the finite setting (exercise). Can we use it nevertheless?

There is no FO[$\emptyset$] formula expressing the domain is even over $\emptyset$-structures.

Proof:

Exploit ∞!

Let λ_k say "there are $\geq k$ elem.".

Suppose that such a φ exists. Consider two theories $\mathcal{T}_1$ and $\mathcal{T}_2$:

$\mathcal{T}_1 := \{\varphi\} \cup \{\lambda_k \mid k \geq 0\}, \quad \mathcal{T}_2 := \{\neg\varphi\} \cup \{\lambda_k \mid k \geq 0\}.$

It's easy to see that any finite subset of $\mathcal{T}_1$ and $\mathcal{T}_2$ is satisfiable (WHY?).

So by compactness $\mathcal{T}_1$ and $\mathcal{T}_2$ are also satisfiable (∞ models!).

Löwenheim–Skolem!

Thus, by Löwenheim–Skolem, $\mathcal{T}_1, \mathcal{T}_2$ have countably-inf models $\mathfrak{A}$ and $\mathfrak{B}$.

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$, and $\mathfrak{A} \models \mathcal{T}_2$ we get $\mathfrak{B} \models \neg\varphi$.

$\emptyset$-structures $=$ sets

As there is a bijection between any two countably-inf sets, we get $\mathfrak{A} \cong \mathfrak{B}$.

Formulae are preserved by isomorphisms, so $\mathfrak{B} \models \neg\varphi$ implies $\mathfrak{A} \models \neg\varphi$:

By $\mathfrak{A} \models \mathcal{T}_1$ we get $\mathfrak{A} \models \varphi$. A contradiction (with the semantics of $\models$)!

Copyright of used icons and pictures

1. Universities/DeciGUT/ERC logos downloaded from the corresponding institutional webpages.
2. Query icons created by Eucalyp - Flaticon `flaticon.com/free-icons/query`
3. Bug icons created by Freepik - Flaticon `flaticon.com/free-icons/bug`
4. Automation icons created by Eucalyp - Flaticon `flaticon.com/free-icons/automation`
5. Head icons created by Eucalyp - Flaticon `flaticon.com/free-icons/head`
6. A picture of Gödel from `mathshistory.st-andrews.ac.uk/Biographies/Godel/pictdisplay/`
7. Money icons created by Smashicons - Flaticon `flaticon.com/free-icons/money`
8. Book covers by ©Springer. No changes have been made.
9. Pepe frog meme picture from `www.pngegg.com/en/png-bbzsj`. Used for non-commercial use.
10. Codd's picture from Wikipedia
11. Protege/SnomedCT/W3C/Dublin Core and the DL logo from their corresponding pages.
12. Model checking picture by Nicolas Markey from

    `people.irisa.fr/Nicolas.Markey/PDF/Talks/170823-NM-TL4MAS.pdf`
13. Descriptive complexity picture by Immerman from his [webpage]

Copyright of used icons and pictures: II

1. Fagin's picture from Wikipedia
2. Holy grail icons created by Freepik - Flaticon c
3. Structural classes picture Felix Reidl. tcs.rwth-aachen.de/reidl/pictures/SparseClasses.svg
4. Picture of Löwenheim from mathshistory.st-andrews.ac.uk/Biographies/Lowenheim
5. Picture of Skolem from en.wikipedia.org/wiki/Thoralf_Skolem
6. Sequents by Thomas Carroll from tex.stackexchange.com/questions/44582/sequent-calculus.
7. Gear icon created by Vectors Market - Flaticon flaticon.com/free-icons/idea.
8. Warning icon created by Freepik - Flaticon flaticon.com/free-icons/warning.