Automated Reasoning

Georg Moser

Institute of Computer Science @ UIBK

Winter 2013

Summary Last Lecture

Definition

- intuitionistic logic is a restriction of classical logic, where certain formulas are no longer derivable
- for example $A \vee \neg A$ is no longer valid

Definition (Curry-Howard)

the Curry-Howard correspondence (aka Curry-Howard isomorphism) consists of the following parts:

1. formulas = types
2. proof = programs
3. normalisation = computation

Example

$$\dfrac{\dfrac{\begin{matrix}\Pi_1 \\ \vdots \\ \Gamma, x : \sigma \Rightarrow M : \tau\end{matrix}}{\Gamma \Rightarrow \lambda x.M : \sigma \to \tau} \quad \begin{matrix}\Pi_2 \\ \vdots \\ \Gamma \Rightarrow N : \tau\end{matrix}}{\Gamma \Rightarrow (\lambda x.M)N : \tau} \quad \Longrightarrow \quad \begin{matrix}\Pi_1[x\backslash\Pi_2] \\ \vdots \\ \Gamma \Rightarrow M[x := N]\end{matrix}$$

the proof $\Pi_1[x\backslash\Pi_2]$ represents the proof that is obtained from Π_1 by replacing assumptions corresponding to the variable x by Π_2

Remark

the Curry-Howard correspondence extends to many systems:

- intuitionistic logic and λ-calculus
- Hilbert axioms and combinatory logic
- ...

Outline of the Lecture

Propositional Logic

short reminder of propositional logic, soundness and completeness theorem, natural deduction, propositional resolution

First Order Logic

introduction, syntax, semantics, undecidability of first-order, Löwenheim-Skolem, compactness, model existence theorem, natural deduction, completeness, sequent calculus, normalisation

Properties of First Order Logic

Craig's Interpolation Theorem, Robinson's Joint Consistency Theorem, Herbrand's Theorem

Limits and Extensions of First Order Logic

Intuitionistic Logic, Curry-Howard Isomorphism, Limits, Second-Order Logic

Outline of the Lecture

Propositional Logic

short reminder of propositional logic, soundness and completeness theorem, natural deduction, propositional resolution

First Order Logic

introduction, syntax, semantics, undecidability of first-order, Löwenheim-Skolem, compactness, model existence theorem, natural deduction, completeness, sequent calculus, normalisation

Properties of First Order Logic

Craig's Interpolation Theorem, Robinson's Joint Consistency Theorem, Herbrand's Theorem

Limits and Extensions of First Order Logic

Intuitionistic Logic, Curry-Howard Isomorphism, Limits, Second-Order Logic

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Proof Sketch.

1. let A be a formula that expresses that node t is reachable from s

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Proof Sketch.

1. let A be a formula that expresses that node t is reachable from s
2. let B_n express that $\neg\exists$ path of length n between s and t

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Proof Sketch.

1. let A be a formula that expresses that node t is reachable from s
2. let B_n express that $\neg\exists$ path of length n between s and t
3. $\mathcal{C} := A \cup \{B_n \mid n \geqslant 1\}$ is unsatisfiable

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Proof Sketch.

1. let A be a formula that expresses that node t is reachable from s
2. let B_n express that $\neg\exists$ path of length n between s and t
3. $\mathcal{C} := A \cup \{B_n \mid n \geqslant 1\}$ is unsatisfiable
4. $\forall$ finite $\mathcal{C}_0 \subset \mathcal{C}$, $\mathcal{C}_0$ is satisfiable

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Proof Sketch.

1. let A be a formula that expresses that node t is reachable from s
2. let B_n express that $\neg\exists$ path of length n between s and t
3. $\mathcal{C} := A \cup \{B_n \mid n \geqslant 1\}$ is unsatisfiable
4. $\forall$ finite $\mathcal{C}_0 \subset \mathcal{C}$, $\mathcal{C}_0$ is satisfiable
5. contradiction to compactness ■

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Proof Sketch.

1. let A be a formula that expresses that node t is reachable from s
2. let B_n express that $\neg\exists$ path of length n between s and t
3. $\mathcal{C} := A \cup \{B_n \mid n \geqslant 1\}$ is unsatisfiable
4. $\forall$ finite $\mathcal{C}_0 \subset \mathcal{C}$, $\mathcal{C}_0$ is satisfiable
5. contradiction to compactness ∎

Corollary

reachability is not expressible in first-order logic

Limits of First-Order Logic

Lemma

given a directed graph $\mathcal{G}$, we can not express the following: let s and t be nodes in $\mathcal{G}$, then there exists a path from s to t

Proof Sketch.

1. let A be a formula that expresses that node t is reachable from s
2. let B_n express that $\neg\exists$ path of length n between s and t
3. $\mathcal{C} := A \cup \{B_n \mid n \geqslant 1\}$ is unsatisfiable
4. $\forall$ finite $\mathcal{C}_0 \subset \mathcal{C}$, $\mathcal{C}_0$ is satisfiable
5. contradiction to compactness ∎

Corollary

reachability is not expressible in first-order logic; i.e., there is no formula $F(x, y)$ such that F holds iff $\exists$ path in graph $\mathcal{G}$ from $\ell(x)$ to $\ell(y)$

Question

what about an infinite set of formulas?

Question

what about an infinite set of formulas?

Definition

let $\mathcal{H}$ be a set of sentences (of $\mathcal{L}$) and let

$$\mathrm{Mod}(\mathcal{H}) = \{\mathcal{A} \mid \mathcal{A} \text{ is a structure (of } \mathcal{L}\text{) and } \mathcal{A} \models \mathcal{H}\}$$

Question

what about an infinite set of formulas?

Definition

let $\mathcal{H}$ be a set of sentences (of $\mathcal{L}$) and let

$$\mathsf{Mod}(\mathcal{H}) = \{\mathcal{A} \mid \mathcal{A} \text{ is a structure (of } \mathcal{L}) \text{ and } \mathcal{A} \models \mathcal{H}\}$$

let $\mathcal{K}$ be a collection of structures

Question

what about an infinite set of formulas?

Definition

let $\mathcal{H}$ be a set of sentences (of $\mathcal{L}$) and let

$$\text{Mod}(\mathcal{H}) = \{\mathcal{A} \mid \mathcal{A} \text{ is a structure (of } \mathcal{L}\text{) and } \mathcal{A} \models \mathcal{H}\}$$

let $\mathcal{K}$ be a collection of structures

- $\mathcal{K}$ is elementary if $\exists$ sentence F and $\mathcal{K} = \text{Mod}(F)$

Question

what about an infinite set of formulas?

Definition

let $\mathcal{H}$ be a set of sentences (of $\mathcal{L}$) and let

$$\text{Mod}(\mathcal{H}) = \{\mathcal{A} \mid \mathcal{A} \text{ is a structure (of } \mathcal{L}\text{) and } \mathcal{A} \models \mathcal{H}\}$$

let $\mathcal{K}$ be a collection of structures

- $\mathcal{K}$ is elementary if $\exists$ sentence F and $\mathcal{K} = \text{Mod}(F)$
- $\mathcal{K}$ is Δ-elementary if $\exists$ set of sentences $\mathcal{H}$ and $\mathcal{K} = \text{Mod}(\mathcal{H})$

Question

what about an infinite set of formulas?

Definition

let $\mathcal{H}$ be a set of sentences (of $\mathcal{L}$) and let

$$\mathsf{Mod}(\mathcal{H}) = \{\mathcal{A} \mid \mathcal{A} \text{ is a structure (of } \mathcal{L}\text{) and } \mathcal{A} \models \mathcal{H}\}$$

let $\mathcal{K}$ be a collection of structures

- $\mathcal{K}$ is elementary if $\exists$ sentence F and $\mathcal{K} = \mathsf{Mod}(F)$
- $\mathcal{K}$ is Δ-elementary if $\exists$ set of sentences $\mathcal{H}$ and $\mathcal{K} = \mathsf{Mod}(\mathcal{H})$

Fact

- *each elementary class is Δ-elementary*
- *every Δ-elementary class is the intersection of elementary classes:*

$$\mathsf{Mod}(\mathcal{F}) = \bigcap_{F \in \mathcal{F}} \mathsf{Mod}(F)$$

Example

reachability is not expressible in first-order logic; that is, the class $\mathcal{K}_1$ of connected graphs is not Δ-elementary

Example

reachability is not expressible in first-order logic; that is, the class $\mathcal{K}_1$ of connected graphs is not Δ-elementary

Proof.

1. suppose $\mathcal{K}_1 = \mathsf{Mod}(\mathcal{H})$ for set of sentences $\mathcal{H}$

Example

reachability is not expressible in first-order logic; that is, the class $\mathcal{K}_1$ of connected graphs is not Δ-elementary

Proof.

1. suppose $\mathcal{K}_1 = \mathrm{Mod}(\mathcal{H})$ for set of sentences $\mathcal{H}$
2. set B_n, $n \geqslant 2$ as $x = y \vee \exists x_1 \cdots \exists x_{n-2}\ R(x, x_1) \wedge \cdots \wedge R(x_{n-2}, y)$

Example

reachability is not expressible in first-order logic; that is, the class $\mathcal{K}_1$ of connected graphs is not Δ-elementary

Proof.

1. suppose $\mathcal{K}_1 = \mathrm{Mod}(\mathcal{H})$ for set of sentences $\mathcal{H}$
2. set B_n, $n \geqslant 2$ as $x = y \vee \exists x_1 \cdots \exists x_{n-2}\ R(x, x_1) \wedge \cdots \wedge R(x_{n-2}, y)$
3. $\forall\ m$, $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n \leqslant m\}$ has a model, but $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n\}$ is unsatisfiable

Example

reachability is not expressible in first-order logic; that is, the class $\mathcal{K}_1$ of connected graphs is not Δ-elementary

Proof.

1. suppose $\mathcal{K}_1 = \mathrm{Mod}(\mathcal{H})$ for set of sentences $\mathcal{H}$
2. set B_n, $n \geqslant 2$ as $x = y \vee \exists x_1 \cdots \exists x_{n-2}\ R(x, x_1) \wedge \cdots \wedge R(x_{n-2}, y)$
3. $\forall\ m$, $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n \leqslant m\}$ has a model, but $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n\}$ is unsatisfiable
4. contradiction to compactness ∎

Example

reachability is not expressible in first-order logic; that is, the class $\mathcal{K}_1$ of connected graphs is not Δ-elementary

Proof.

1. suppose $\mathcal{K}_1 = \mathrm{Mod}(\mathcal{H})$ for set of sentences $\mathcal{H}$
2. set B_n, $n \geqslant 2$ as $x = y \vee \exists x_1 \cdots \exists x_{n-2}\ R(x, x_1) \wedge \cdots \wedge R(x_{n-2}, y)$
3. $\forall\ m$, $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n \leqslant m\}$ has a model, but $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n\}$ is unsatisfiable
4. contradiction to compactness ∎

Answer

infinite set of formulas are not enough

Example

reachability is not expressible in first-order logic; that is, the class $\mathcal{K}_1$ of connected graphs is not Δ-elementary

Proof.

1. suppose $\mathcal{K}_1 = \mathrm{Mod}(\mathcal{H})$ for set of sentences $\mathcal{H}$
2. set B_n, $n \geqslant 2$ as $x = y \vee \exists x_1 \cdots \exists x_{n-2}\ R(x, x_1) \wedge \cdots \wedge R(x_{n-2}, y)$
3. $\forall\ m$, $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n \leqslant m\}$ has a model, but $\mathcal{H} \cup \{\neg B_n \mid 2 \leqslant n\}$ is unsatisfiable
4. contradiction to compactness ∎

Answer

infinite set of formulas are not enough

Example

finiteness is not expressible in first-order logic

The Language of Second-Order Logic

a second-order language extends a first-order language as follows

The Language of Second-Order Logic

a second-order language extends a first-order language as follows

Definition

1. first-order variables — individual variables

The Language of Second-Order Logic

a second-order language extends a first-order language as follows

Definition

1. first-order variables — individual variables
2. relation (or predicate) variables
   $V_0^i, V_1^i, \dots, V_j^i, \dots$ — denoted X, Y, Z, etc.

The Language of Second-Order Logic

a second-order language extends a first-order language as follows

Definition

1. first-order variables — individual variables
2. relation (or predicate) variables
   $V^i_0, V^i_1, \dots, V^i_j, \dots$ — denoted X, Y, Z, etc.
3. function variables
   $u^i_0, u^i_1, \dots, u^i_j, \dots$ — denoted u, v, w, etc.

The Language of Second-Order Logic

a second-order language extends a first-order language as follows

Definition

1. first-order variables — individual variables
2. relation (or predicate) variables
   $V^i_0, V^i_1, \ldots, V^i_j, \ldots$ — denoted X, Y, Z, etc.
3. function variables
   $u^i_0, u^i_1, \ldots, u^i_j, \ldots$ — denoted u, v, w, etc.

Definition

second-order terms are defined like first-order terms together with the following clause

The Language of Second-Order Logic

a second-order language extends a first-order language as follows

Definition

1. first-order variables — individual variables
2. relation (or predicate) variables
   $V^i_0, V^i_1, \dots, V^i_j, \dots$ — denoted X, Y, Z, etc.
3. function variables
   $u^i_0, u^i_1, \dots, u^i_j, \dots$ — denoted u, v, w, etc.

Definition

second-order terms are defined like first-order terms together with the following clause

4. if t_1, ..., t_n are second-order terms, u an n-ary function variable, then $u(t_1, \dots, t_n)$ is a second-order term

The Language of Second-Order Logic

a second-order language extends a first-order language as follows

Definition

1. first-order variables — individual variables
2. relation (or predicate) variables
   $V^i_0, V^i_1, \dots, V^i_j, \dots$ — denoted X, Y, Z, etc.
3. function variables
   $u^i_0, u^i_1, \dots, u^i_j, \dots$ — denoted u, v, w, etc.

Definition

second-order terms are defined like first-order terms together with the following clause

4. if t_1, ..., t_n are second-order terms, u an n-ary function variable, then $u(t_1, \dots, t_n)$ is a second-order term

a second-order terms without function variables is first-order

Definition

second-order formulas are defined as follows

Definition

second-order formulas are defined as follows

1. first-order formulas are second-order formula

Definition

second-order formulas are defined as follows

1. first-order formulas are second-order formula
2. if $t_1, \ldots, t_n$ are second-order terms, X an n-ary predicate variable, then $X(t_1, \ldots, t_n)$ is a second-order formula

Definition

second-order formulas are defined as follows

1. first-order formulas are second-order formula
2. if $t_1, \dots, t_n$ are second-order terms, X an n-ary predicate variable, then $X(t_1, \dots, t_n)$ is a second-order formula
3. If $A(f)$ is a second-order formula, f a function constant, u a function variable, then

$$\forall u\ A(u) \qquad \exists u\ A(u)$$

are second-order formulas

Definition

second-order formulas are defined as follows

1. first-order formulas are second-order formula
2. if $t_1, \dots, t_n$ are second-order terms, X an n-ary predicate variable, then $X(t_1, \dots, t_n)$ is a second-order formula
3. If $A(f)$ is a second-order formula, f a function constant, u a function variable, then
$$\forall u \ A(u) \qquad \exists u \ A(u)$$
are second-order formulas
4. if $A(P)$ a formula, P a predicate constant, X a predicate variable, then
$$\forall X \ A(X) \qquad \exists X \ A(X)$$
are second-order formulas

Definition

second-order formulas are defined as follows

1. first-order formulas are second-order formula
2. if $t_1, \dots, t_n$ are second-order terms, X an n-ary predicate variable, then $X(t_1, \dots, t_n)$ is a second-order formula
3. If $A(f)$ is a second-order formula, f a function constant, u a function variable, then
$$\forall u\ A(u) \qquad \exists u\ A(u)$$
are second-order formulas
4. if $A(P)$ a formula, P a predicate constant, X a predicate variable, then
$$\forall X\ A(X) \qquad \exists X\ A(X)$$
are second-order formulas
5. a second-order formula without predicate and function variables is first-order

Example

let u denote a function variable, X a predicate variable

$$\forall x\ f(x) = x \qquad \exists u \forall x\ u(x) = x$$

Example

let u denote a function variable, X a predicate variable

$$\forall x \; f(x) = x \qquad \exists u \forall x \; u(x) = x$$

1. the first formulas expresses a property of the identity function

Example

let u denote a function variable, X a predicate variable

$$\forall x\ f(x) = x \qquad \exists u \forall x\ u(x) = x$$

1. the first formulas expresses a property of the identity function
2. the 2nd asserts existence of an identity function

Example

let u denote a function variable, X a predicate variable

$$\forall x \; f(x) = x \qquad \exists u \forall x \; u(x) = x$$

1. the first formulas expresses a property of the identity function
2. the 2nd asserts existence of an identity function

Example

consider

$$x = y \rightarrow (P(x) \leftrightarrow P(y)) \qquad x = y \leftrightarrow \forall X(X(x) \leftrightarrow X(y))$$

Example

let u denote a function variable, X a predicate variable

$$\forall x \; f(x) = x \qquad \exists u \forall x \; u(x) = x$$

1. the first formulas expresses a property of the identity function
2. the 2nd asserts existence of an identity function

Example

consider

$$x = y \rightarrow (P(x) \leftrightarrow P(y)) \qquad x = y \leftrightarrow \forall X (X(x) \leftrightarrow X(y))$$

1. the first formulas expresses a property of equality

Example

let u denote a function variable, X a predicate variable

$$\forall x\ f(x) = x \qquad \exists u \forall x\ u(x) = x$$

1. the first formulas expresses a property of the identity function
2. the 2nd asserts existence of an identity function

Example

consider

$$x = y \to (P(x) \leftrightarrow P(y)) \qquad x = y \leftrightarrow \forall X(X(x) \leftrightarrow X(y))$$

1. the first formulas expresses a property of equality
2. the 2nd asserts defines equality

Second-Order Interpretation

Definition

a second-order environment ℓ for $\mathcal{A}$ is a mapping

$$\ell \colon \{\{x_n \mid n \in \mathbb{N}\} \to A\} \cup \{\{u_n^i \mid i, n \in \mathbb{N}\} \to (A^i \to A)\} \cup \{\{V_n^i \mid i, n \in \mathbb{N}\} \to A^i\}$$

Second-Order Interpretation

Definition

a second-order environment ℓ for $\mathcal{A}$ is a mapping

$$\ell \colon \{\{x_n \mid n \in \mathbb{N}\} \to A\} \cup \{\{u_n^i \mid i, n \in \mathbb{N}\} \to (A^i \to A)\} \cup \{\{V_n^i \mid i, n \in \mathbb{N}\} \to A^i\}$$

$\ell\{X \mapsto A'\}$ maps X to relation $A' \subseteq A^n$ if X is n-ary; all other maps are unchanged; similarly for function variables

Second-Order Interpretation

Definition

a second-order environment ℓ for $\mathcal{A}$ is a mapping

$$\ell \colon \{\{x_n \mid n \in \mathbb{N}\} \to A\} \cup$$
$$\{\{u_n^i \mid i, n \in \mathbb{N}\} \to (A^i \to A)\} \cup \{\{V_n^i \mid i, n \in \mathbb{N}\} \to A^i\}$$

$\ell\{X \mapsto A'\}$ maps X to relation $A' \subseteq A^n$ if X is n-ary; all other maps are unchanged; similarly for function variables

Definition

a second-order interpretation $\mathcal{I}$ is a pair $(\mathcal{A}, \ell)$ such that

Second-Order Interpretation

Definition

a second-order environment ℓ for $\mathcal{A}$ is a mapping

$$\ell \colon \{\{x_n \mid n \in \mathbb{N}\} \to A\} \cup \{\{u^i_n \mid i, n \in \mathbb{N}\} \to (A^i \to A)\} \cup \{\{V^i_n \mid i, n \in \mathbb{N}\} \to A^i\}$$

$\ell\{X \mapsto A'\}$ maps X to relation $A' \subseteq A^n$ if X is n-ary; all other maps are unchanged; similarly for function variables

Definition

a second-order interpretation $\mathcal{I}$ is a pair $(\mathcal{A}, \ell)$ such that

- $\mathcal{A}$ is a structure

Second-Order Interpretation

Definition

a second-order environment ℓ for $\mathcal{A}$ is a mapping

$$\ell\colon \{\{x_n \mid n \in \mathbb{N}\} \to A\} \cup \{\{u_n^i \mid i, n \in \mathbb{N}\} \to (A^i \to A)\} \cup \{\{V_n^i \mid i, n \in \mathbb{N}\} \to A^i\}$$

$\ell\{X \mapsto A'\}$ maps X to relation $A' \subseteq A^n$ if X is n-ary; all other maps are unchanged; similarly for function variables

Definition

a second-order interpretation $\mathcal{I}$ is a pair $(\mathcal{A}, \ell)$ such that

- $\mathcal{A}$ is a structure
- ℓ is a second-order environment

Example

consider the structure $\mathcal{A}$ with domain $\mathbb{N}$; $\ell(u) = \mathsf{succ}$ and $\ell(x) = 0$ and let $\mathcal{I} = (\mathcal{A}, \ell)$

$$u(x)^{\mathcal{I}} = \mathsf{succ}(0) = 1$$

Example

consider the structure $\mathcal{A}$ with domain $\mathbb{N}$; $\ell(u) = \mathsf{succ}$ and $\ell(x) = 0$ and let $\mathcal{I} = (\mathcal{A}, \ell)$

$$u(x)^{\mathcal{I}} = \mathsf{succ}(0) = 1$$

Definition

the value of a second-order term t:

$$t^{\mathcal{I}} = \begin{cases} \ell(t) & \text{if } t \text{ an individual variable} \\ c^{\mathcal{A}} & \text{if } t = c \\ f^{\mathcal{A}}(t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) & \text{if } t = f(t_1, \dots, t_n),\ f \text{ a function constant} \\ \ell(u)(t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) & \text{if } t = u(t_1, \dots, t_n),\ u \text{ a function variable} \end{cases}$$

Satisfaction relation

Definition

$\mathcal{I} = (\mathcal{A}, \ell)$ an interpretation; F a formula

Satisfaction relation

Definition

$\mathcal{I} = (\mathcal{A}, \ell)$ an interpretation; F a formula

$$
\begin{array}{lcl}
\mathcal{I} \models P(t_1, \dots, t_n) & :\Longleftrightarrow & \text{if } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in P^{\mathcal{A}} \\
\mathcal{I} \models \neg F & :\Longleftrightarrow & \text{if } \mathcal{I} \not\models F \\
\mathcal{I} \models F \vee G & :\Longleftrightarrow & \text{if } \mathcal{I} \models F \text{ or } \mathcal{I} \models G \\
\mathcal{I} \models \forall x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for all } a \in A \\
\mathcal{I} \models \exists x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for some } a \in A
\end{array}
$$

Satisfaction relation

Definition

$\mathcal{I} = (\mathcal{A}, \ell)$ an interpretation; F a formula

$$\begin{array}{lll}
\mathcal{I} \models P(t_1, \dots, t_n) & :\Longleftrightarrow & \text{if } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in P^{\mathcal{A}} \\
\mathcal{I} \models \neg F & :\Longleftrightarrow & \text{if } \mathcal{I} \not\models F \\
\mathcal{I} \models F \vee G & :\Longleftrightarrow & \text{if } \mathcal{I} \models F \text{ or } \mathcal{I} \models G \\
\mathcal{I} \models \forall x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for all } a \in A \\
\mathcal{I} \models \exists x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for some } a \in A \\
\mathcal{I} \models X(t_1, \dots, t_n) & :\Longleftrightarrow & \ell(X) = A' \text{ and } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in A'
\end{array}$$

Satisfaction relation

Definition

$\mathcal{I} = (\mathcal{A}, \ell)$ an interpretation; F a formula

$$
\begin{array}{lcl}
\mathcal{I} \models P(t_1, \dots, t_n) & :\Longleftrightarrow & \text{if } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in P^{\mathcal{A}} \\
\mathcal{I} \models \neg F & :\Longleftrightarrow & \text{if } \mathcal{I} \not\models F \\
\mathcal{I} \models F \vee G & :\Longleftrightarrow & \text{if } \mathcal{I} \models F \text{ or } \mathcal{I} \models G \\
\mathcal{I} \models \forall x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for all } a \in A \\
\mathcal{I} \models \exists x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for some } a \in A \\
\mathcal{I} \models X(t_1, \dots, t_n) & :\Longleftrightarrow & \ell(X) = A' \text{ and } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in A' \\
\mathcal{I} \models \forall X\ F(X) & :\Longleftrightarrow & \text{if } \mathcal{I}\{X \mapsto A'\} \models F(X) \text{ for all } A' \subseteq A^n
\end{array}
$$

Satisfaction relation

Definition

$\mathcal{I} = (\mathcal{A}, \ell)$ an interpretation; F a formula

$$
\begin{array}{lll}
\mathcal{I} \models P(t_1, \dots, t_n) & :\Longleftrightarrow & \text{if } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in P^{\mathcal{A}} \\
\mathcal{I} \models \neg F & :\Longleftrightarrow & \text{if } \mathcal{I} \not\models F \\
\mathcal{I} \models F \vee G & :\Longleftrightarrow & \text{if } \mathcal{I} \models F \text{ or } \mathcal{I} \models G \\
\mathcal{I} \models \forall x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for all } a \in A \\
\mathcal{I} \models \exists x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for some } a \in A \\
\mathcal{I} \models X(t_1, \dots, t_n) & :\Longleftrightarrow & \ell(X) = A' \text{ and } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in A' \\
\mathcal{I} \models \forall X\ F(X) & :\Longleftrightarrow & \text{if } \mathcal{I}\{X \mapsto A'\} \models F(X) \text{ for all } A' \subseteq A^n \\
\mathcal{I} \models \exists X\ F(X) & :\Longleftrightarrow & \text{if } \mathcal{I}\{X \mapsto A'\} \models F(X) \text{ for some } A' \subseteq A^n
\end{array}
$$

Satisfaction relation

Definition

$\mathcal{I} = (\mathcal{A}, \ell)$ an interpretation; F a formula

$$
\begin{array}{lcl}
\mathcal{I} \models P(t_1, \dots, t_n) & :\Longleftrightarrow & \text{if } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in P^{\mathcal{A}} \\
\mathcal{I} \models \neg F & :\Longleftrightarrow & \text{if } \mathcal{I} \not\models F \\
\mathcal{I} \models F \vee G & :\Longleftrightarrow & \text{if } \mathcal{I} \models F \text{ or } \mathcal{I} \models G \\
\mathcal{I} \models \forall x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for all } a \in A \\
\mathcal{I} \models \exists x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for some } a \in A \\
\mathcal{I} \models X(t_1, \dots, t_n) & :\Longleftrightarrow & \ell(X) = A' \text{ and } (t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}}) \in A' \\
\mathcal{I} \models \forall X\ F(X) & :\Longleftrightarrow & \text{if } \mathcal{I}\{X \mapsto A'\} \models F(X) \text{ for all } A' \subseteq A^n \\
\mathcal{I} \models \exists X\ F(X) & :\Longleftrightarrow & \text{if } \mathcal{I}\{X \mapsto A'\} \models F(X) \text{ for some } A' \subseteq A^n \\
\mathcal{I} \models \forall u\ F(u) & :\Longleftrightarrow & \text{if } \mathcal{I}\{u \mapsto f\} \models F(u) \text{ for all } f \colon A^n \to A
\end{array}
$$

Satisfaction relation

Definition

$\mathcal{I} = (\mathcal{A}, \ell)$ an interpretation; F a formula

$$
\begin{array}{lcl}
\mathcal{I} \models P(t_1, \ldots, t_n) & :\Longleftrightarrow & \text{if } (t_1^{\mathcal{I}}, \ldots, t_n^{\mathcal{I}}) \in P^{\mathcal{A}} \\
\mathcal{I} \models \neg F & :\Longleftrightarrow & \text{if } \mathcal{I} \not\models F \\
\mathcal{I} \models F \vee G & :\Longleftrightarrow & \text{if } \mathcal{I} \models F \text{ or } \mathcal{I} \models G \\
\mathcal{I} \models \forall x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for all } a \in A \\
\mathcal{I} \models \exists x\ F(x) & :\Longleftrightarrow & \text{if } \mathcal{I}\{x \mapsto a\} \models F(x) \text{ holds for some } a \in A \\
\mathcal{I} \models X(t_1, \ldots, t_n) & :\Longleftrightarrow & \ell(X) = A' \text{ and } (t_1^{\mathcal{I}}, \ldots, t_n^{\mathcal{I}}) \in A' \\
\mathcal{I} \models \forall X\ F(X) & :\Longleftrightarrow & \text{if } \mathcal{I}\{X \mapsto A'\} \models F(X) \text{ for all } A' \subseteq A^n \\
\mathcal{I} \models \exists X\ F(X) & :\Longleftrightarrow & \text{if } \mathcal{I}\{X \mapsto A'\} \models F(X) \text{ for some } A' \subseteq A^n \\
\mathcal{I} \models \forall u\ F(u) & :\Longleftrightarrow & \text{if } \mathcal{I}\{u \mapsto f\} \models F(u) \text{ for all } f \colon A^n \to A \\
\mathcal{I} \models \exists u\ F(u) & :\Longleftrightarrow & \text{if } \mathcal{I}\{u \mapsto f\} \models F(u) \text{ for some } f \colon A^n \to A
\end{array}
$$

Reachability is Expressible in Second-Order Logic

Example

- let $\mathcal{G}$ be a structure defined over the language $\mathcal{L} = \{R\}$ with the domain G

Reachability is Expressible in Second-Order Logic

Example

- let $\mathcal{G}$ be a structure defined over the language $\mathcal{L} = \{R\}$ with the domain G
- R represents the (directed) edge relation of the graph $\mathcal{G}$

Reachability is Expressible in Second-Order Logic

Example

- let $\mathcal{G}$ be a structure defined over the language $\mathcal{L} = \{R\}$ with the domain G
- R represents the (directed) edge relation of the graph $\mathcal{G}$
- consider the second order formula $F(x, y)$

$$\begin{gathered}\exists P\big(\forall z_1 \forall z_2 \forall z_3 \big(\neg P(z_1, z_1) \land \\ \land (P(z_1, z_2) \land P(z_2, z_3) \to P(z_1, z_3))\big) \land \\ \land \forall z_1 \forall z_2 ((P(z_1, z_2) \land \neg \exists z_3 (P(z_1, z_3) \land P(z_3, z_2)) \to R(z_1, z_2)) \land \\ \land P(x, y)\big)\end{gathered}$$

Reachability is Expressible in Second-Order Logic

Example

- let $\mathcal{G}$ be a structure defined over the language $\mathcal{L} = \{R\}$ with the domain G
- R represents the (directed) edge relation of the graph $\mathcal{G}$
- consider the second order formula $F(x, y)$

$$\begin{gathered}
\exists P\big(\forall z_1 \forall z_2 \forall z_3 \big(\neg P(z_1, z_1) \land \\
\land (P(z_1, z_2) \land P(z_2, z_3) \rightarrow P(z_1, z_3))\big) \land \\
\land \forall z_1 \forall z_2((P(z_1, z_2) \land \neg \exists z_3(P(z_1, z_3) \land P(z_3, z_2)) \rightarrow R(z_1, z_2)) \land \\
\land P(x, y)\big)
\end{gathered}$$

- suppose $\mathcal{I} \models F(x, y)$, then $\exists$ path in $\mathcal{G}$ from $\ell(x)$ to $\ell(y)$

More examples

Example

consider Whitehead-Russel definition of equality:

$$x = y \iff \forall X (X(x) \to X(y))$$

More examples

Example

consider Whitehead-Russel definition of equality:

$$x = y \iff \forall X (X(x) \to X(y))$$

Lemma

Leibnitz's equality and Whitehead-Russel's equality are equivalent

More examples

Example

consider Whitehead-Russel definition of equality:

$$x = y \iff \forall X(X(x) \to X(y))$$

Lemma

Leibnitz's equality and Whitehead-Russel's equality are equivalent

Example

consider the following "axiom" of enumerability (Enum)

$$\exists z \exists u \forall X((X(z) \land \forall x(X(x) \to X(u(x)))) \to \forall x X(x))$$

which is true in an interpretation iff its domain is countable

Example

consider the following "axiom" of infinity (Inf)

$$\exists z \exists u (\forall x z \neq u(x) \wedge \forall x \forall y (u(x) = u(y) \rightarrow x = y))$$

which is true in an interpretation iff the domain it infinite

Example

consider the following "axiom" of infinity (Inf)

$$\exists z \exists u (\forall x z \neq u(x) \land \forall x \forall y (u(x) = u(y) \rightarrow x = y))$$

which is true in an interpretation iff the domain it infinite

Lemma

Löwenheim-Skolem fails for second-order logic

Example

consider the following "axiom" of infinity (Inf)

$$\exists z \exists u (\forall x z \neq u(x) \wedge \forall x \forall y (u(x) = u(y) \to x = y))$$

which is true in an interpretation iff the domain it infinite

Lemma

Löwenheim-Skolem fails for second-order logic

Proof.

1. recall that Löwenheim-Skolem asserts that if a set of sentences $\mathcal{G}$ has a model, then $\mathcal{G}$ has a countable model
2. consider $\mathcal{G} = \{\neg \mathsf{Enum}, \mathsf{Inf}\}$
3. then $\mathcal{G}$ is satisfiable, but only with uncountable models
4. contradiction

Example

consider the following "axiom" of infinity (Inf)

$$\exists z \exists u (\forall x z \neq u(x) \land \forall x \forall y (u(x) = u(y) \to x = y))$$

which is true in an interpretation iff the domain it infinite

Lemma

Löwenheim-Skolem fails for second-order logic

Proof.

1. recall that Löwenheim-Skolem asserts that if a set of sentences $\mathcal{G}$ has a model, then $\mathcal{G}$ has a countable model
2. consider $\mathcal{G} = \{\neg \mathrm{Enum}, \mathrm{Inf}\}$
3. then $\mathcal{G}$ is satisfiable, but only with uncountable models
4. contradiction

∎

Definition

consider (the following variant of) Robinson's **Q**

$$
\begin{array}{ll}
N_1: & \mathsf{s}(v_1) = \mathsf{s}(v_2) \rightarrow v_1 = v_2 \\
N_2: & 0 \neq \mathsf{s}(v_1) \\
N_3: & (v_1 + 0) = v_1 \\
N_4: & (v_1 + \mathsf{s}(v_2)) = \mathsf{s}(v_1 + v_2) \\
N_5: & (v_1 \cdot 0) = 0 \\
N_6: & (v_1 \cdot \mathsf{s}(v_2)) = ((v_1 \cdot v_2) + v_1) \\
N_7: & (v_1 \leqslant 0) \iff (v_1 = 0) \\
N_8: & (v_1 \leqslant \mathsf{s}(v_2)) \iff (v_1 \leqslant v_2 \vee v_1 = \mathsf{s}(v_2)) \\
N_9: & (v_1 \leqslant v_2) \vee (v_2 \leqslant v_1)
\end{array}
$$

Definition

consider (the following variant of) Robinson's **Q**

$$
\begin{aligned}
N_1: &\quad \mathsf{s}(v_1) = \mathsf{s}(v_2) \rightarrow v_1 = v_2 \\
N_2: &\quad 0 \neq \mathsf{s}(v_1) \\
N_3: &\quad (v_1 + 0) = v_1 \\
N_4: &\quad (v_1 + \mathsf{s}(v_2)) = \mathsf{s}(v_1 + v_2) \\
N_5: &\quad (v_1 \cdot 0) = 0 \\
N_6: &\quad (v_1 \cdot \mathsf{s}(v_2)) = ((v_1 \cdot v_2) + v_1) \\
N_7: &\quad (v_1 \leqslant 0) \iff (v_1 = 0) \\
N_8: &\quad (v_1 \leqslant \mathsf{s}(v_2)) \iff (v_1 \leqslant v_2 \vee v_1 = \mathsf{s}(v_2)) \\
N_9: &\quad (v_1 \leqslant v_2) \vee (v_2 \leqslant v_1)
\end{aligned}
$$

Fact

Q *is complete for quantifier-free sentences of the language of arithmetic*

Example

let $\mathbf{P}^2$ be the axioms in $\mathbf{Q}$ together with the following axiom of induction

$$\forall X((X(0) \wedge \forall x(X(x) \rightarrow X(\mathsf{s}(x)))) \rightarrow \forall x X(x))$$

then any interpretation of the language of arithmetic is a model of $\mathbf{P}^2$ iff it is isomorphic to the standard interpretation

Example

let $\mathbf{P}^2$ be the axioms in $\mathbf{Q}$ together with the following axiom of induction

$$\forall X((X(0) \land \forall x(X(x) \to X(\mathsf{s}(x)))) \to \forall x X(x))$$

then any interpretation of the language of arithmetic is a model of $\mathbf{P}^2$ iff it is isomorphic to the standard interpretation

Lemma

compactness fails for second-order logic

Example

let $\mathbf{P}^2$ be the axioms in $\mathbf{Q}$ together with the following axiom of induction

$$\forall X((X(0) \wedge \forall x(X(x) \rightarrow X(\mathsf{s}(x)))) \rightarrow \forall x X(x))$$

then any interpretation of the language of arithmetic is a model of $\mathbf{P}^2$ iff it is isomorphic to the standard interpretation

Lemma

compactness fails for second-order logic

Proof.

1. add a constant c to the language of arithmetic
2. consider $\mathcal{G} = \{\mathbf{P}^2, \mathsf{c} \neq 0, \mathsf{c} \neq 1, \mathsf{c} \neq 2, \dots\}$
3. any finite subset of $\mathcal{G}$ is satisfiable, while $\mathcal{G}$ is not
4. contradiction

Summary of Bad News

Lemma

the set of valid second-order sentences is not recursively enumerable

Summary of Bad News

Lemma

the set of valid second-order sentences is not recursively enumerable

Proof Sketch.

the proof essentially employs that $\mathbf{P}^2$ exactly confirms to number theory, as the later is incomplete $\neg \, \exists$ a calculus complete for second-order ∎

Summary of Bad News

Lemma

the set of valid second-order sentences is not recursively enumerable

Proof Sketch.

the proof essentially employs that $\mathbf{P}^2$ exactly confirms to number theory, as the later is incomplete $\neg\ \exists$ a calculus complete for second-order ■

Theorem

1. *compactness fails for second-order logic*
2. *Löwenheim-Skolem fails for second-order logic*
3. $\neg\ \exists$ *a calculus that is complete for second-order logic*

Summary of Bad News

Lemma

the set of valid second-order sentences is not recursively enumerable

Proof Sketch.

the proof essentially employs that $\mathbf{P}^2$ exactly confirms to number theory, as the later is incomplete $\neg \exists$ a calculus complete for second-order ∎

Theorem

1. *compactness fails for second-order logic*
2. *Löwenheim-Skolem fails for second-order logic*
3. $\neg \exists$ *a calculus that is complete for second-order logic, in particular the set of valid second-order sentences is not recursively enumerable*

Good News

Example

$\exists$ set $\mathcal{H}$ of second-order sentences, such that $\mathsf{Mod}^{\mathsf{fin}}(\mathcal{H}) = \mathsf{NP}$

Good News

Example

finite models

$\exists$ set $\mathcal{H}$ of second-order sentences, such that $\mathsf{Mod}^{\mathsf{fin}}(\mathcal{H}) = \mathsf{NP}$

Good News

Example

finite models

$\exists$ set $\mathcal{H}$ of second-order sentences, such that $\mathrm{Mod}^{\mathrm{fin}}(\mathcal{H}) = \mathrm{NP}$

Definition

- Let $\mathcal{K}$ be a set of finite structures and let F be a (second-order) sentence
- suppose $\mathcal{M}$ is a (second-order) structure in $\mathcal{K}$

then the $F-\mathcal{K}$ problem asks, whether $\mathcal{M} \models F$ holds

Definition (existential second-order formula ($\exists$SO))

we call a second-order formula F existential if F has the following form:

$$\exists X_1 \exists X_2 \cdots \exists X_n \ G$$

Definition (existential second-order formula ($\exists$SO))

we call a second-order formula F existential if F has the following form:

$$\exists X_1 \exists X_2 \cdots \exists X_n \; G$$

where G is essentially a first-order formula that may contain the free second-order variables $X_1, \dots, X_n$

Definition (existential second-order formula (∃SO))

we call a second-order formula F existential if F has the following form:

$$\exists X_1 \exists X_2 \cdots \exists X_n \; G$$

where G is essentially a first-order formula that may contain the free second-order variables $X_1, \dots, X_n$

Lemma ①

if F is ∃SO, then the $F-\mathcal{K}$ problem is in NP

Definition (existential second-order formula (∃SO))

we call a second-order formula F existential if F has the following form:

$$\exists X_1 \exists X_2 \cdots \exists X_n \; G$$

where G is essentially a first-order formula that may contain the free second-order variables $X_1, \dots, X_n$

Lemma ①

if F is ∃SO, then the $F-\mathcal{K}$ problem is in NP

Lemma ②

if $F-\mathcal{K}$ is decidable by a NTM M that runs in polynomial time then F is equivalent to an existential second-order sentence

An Implicit Characterisation of a Complexity Class

Theorem (Fagin's Theorem)

1. *a sentence F is equivalent to a sentence in* $\exists SO$ *iff* $F - \mathcal{K} \in \mathsf{NP}$

An Implicit Characterisation of a Complexity Class

Theorem (Fagin's Theorem)

1. *a sentence F is equivalent to a sentence in $\exists SO$ iff $F - \mathcal{K} \in \mathrm{NP}$*
2. *if $F - \mathcal{K} \in \mathrm{NP}$, then it can be assumed that the first-order part of F is a universal formula*

An Implicit Characterisation of a Complexity Class

Theorem (Fagin's Theorem)

1. *a sentence F is equivalent to a sentence in $\exists SO$ iff $F-\mathcal{K} \in \mathsf{NP}$*
2. *if $F-\mathcal{K} \in \mathsf{NP}$, then it can be assumed that the first-order part of F is a universal formula*

Proof.

1. suppose F is an existential second-order sentence; by Lemma ①, $F-\mathcal{K} \in \mathsf{NP}$

An Implicit Characterisation of a Complexity Class

Theorem (Fagin's Theorem)

1. *a sentence F is equivalent to a sentence in $\exists SO$ iff $F - \mathcal{K} \in \mathsf{NP}$*
2. *if $F - \mathcal{K} \in \mathsf{NP}$, then it can be assumed that the first-order part of F is a universal formula*

Proof.

1. suppose F is an existential second-order sentence; by Lemma ①, $F - \mathcal{K} \in \mathsf{NP}$
2. suppose $F - \mathcal{K} \in \mathsf{NP}$; $\exists$ NTM N that decides $F - \mathcal{K}$

An Implicit Characterisation of a Complexity Class

Theorem (Fagin's Theorem)

1. *a sentence F is equivalent to a sentence in $\exists SO$ iff $F-\mathcal{K} \in \mathsf{NP}$*
2. *if $F-\mathcal{K} \in \mathsf{NP}$, then it can be assumed that the first-order part of F is a universal formula*

Proof.

1. suppose F is an existential second-order sentence; by Lemma ①, $F-\mathcal{K} \in \mathsf{NP}$
2. suppose $F-\mathcal{K} \in \mathsf{NP}$; $\exists$ NTM N that decides $F-\mathcal{K}$
3. by Lemma ②, F is equivalent to an $\exists$SO-formula G

An Implicit Characterisation of a Complexity Class

Theorem (Fagin's Theorem)

1. *a sentence F is equivalent to a sentence in $\exists SO$ iff $F-\mathcal{K} \in \mathsf{NP}$*
2. *if $F-\mathcal{K} \in \mathsf{NP}$, then it can be assumed that the first-order part of F is a universal formula*

Proof.

1. suppose F is an existential second-order sentence; by Lemma ①, $F-\mathcal{K} \in \mathsf{NP}$
2. suppose $F-\mathcal{K} \in \mathsf{NP}$; $\exists$ NTM N that decides $F-\mathcal{K}$
3. by Lemma ②, F is equivalent to an $\exists$SO-formula G
4. the proof of the second lemma even yields that the first-order part of G is universal

An Implicit Characterisation of a Complexity Class

Theorem (Fagin's Theorem)

1. *a sentence F is equivalent to a sentence in $\exists$SO iff $F-\mathcal{K} \in \mathsf{NP}$*
2. *if $F-\mathcal{K} \in \mathsf{NP}$, then it can be assumed that the first-order part of F is a universal formula*

Proof.

1. suppose F is an existential second-order sentence; by Lemma ①, $F-\mathcal{K} \in \mathsf{NP}$
2. suppose $F-\mathcal{K} \in \mathsf{NP}$; $\exists$ NTM N that decides $F-\mathcal{K}$
3. by Lemma ②, F is equivalent to an $\exists$SO-formula G
4. the proof of the second lemma even yields that the first-order part of G is universal ∎

Corollary

SAT *is* NP-*complete (wrt. the polytime reducibility relation)*

Corollary

SAT *is* NP*-complete (wrt. the polytime reducibility relation)*

Proof.

1. SAT $\in$ NP follows from Lemma ①, as SAT can be easily encoded as $\exists$SO-formula

Corollary

SAT *is* NP-*complete (wrt. the polytime reducibility relation)*

Proof.

1. SAT $\in$ NP follows from Lemma ①, as SAT can be easily encoded as $\exists$SO-formula
2. thus let $A \in$ NP

Corollary

SAT *is* NP*-complete (wrt. the polytime reducibility relation)*

Proof.

1. SAT $\in$ NP follows from Lemma ①, as SAT can be easily encoded as $\exists$SO-formula
2. thus let $A \in$ NP
3. by Fagin's theorem, there exists an $\exists$SO-formula F and some finite structures $\mathcal{K}$, such that A is equivalent to the $F-\mathcal{K}$ problem; moreover the first-order part of F is univeral

Corollary

SAT *is* NP-*complete (wrt. the polytime reducibility relation)*

Proof.

1. SAT $\in$ NP follows from Lemma ①, as SAT can be easily encoded as $\exists$SO-formula
2. thus let $A \in$ NP
3. by Fagin's theorem, there exists an $\exists$SO-formula F and some finite structures $\mathcal{K}$, such that A is equivalent to the $F - \mathcal{K}$ problem; moreover the first-order part of F is univeral
4. let $\mathcal{M} \in \mathcal{K}$ be a finite; the universal part of F can be represented as propositional formula B

Corollary

SAT *is* NP*-complete (wrt. the polytime reducibility relation)*

Proof.

1. SAT $\in$ NP follows from Lemma ①, as SAT can be easily encoded as $\exists$SO-formula
2. thus let $A \in$ NP
3. by Fagin's theorem, there exists an $\exists$SO-formula F and some finite structures $\mathcal{K}$, such that A is equivalent to the $F - \mathcal{K}$ problem; moreover the first-order part of F is univeral
4. let $\mathcal{M} \in \mathcal{K}$ be a finite; the universal part of F can be represented as propositional formula B
5. any interpretation of F is conceivable as an assignment of B (and vice versa)

Corollary

SAT *is* NP*-complete (wrt. the polytime reducibility relation)*

Proof.

1. SAT $\in$ NP follows from Lemma ①, as SAT can be easily encoded as $\exists$SO-formula
2. thus let $A \in$ NP
3. by Fagin's theorem, there exists an $\exists$SO-formula F and some finite structures $\mathcal{K}$, such that A is equivalent to the $F - \mathcal{K}$ problem; moreover the first-order part of F is univeral
4. let $\mathcal{M} \in \mathcal{K}$ be a finite; the universal part of F can be represented as propositional formula B
5. any interpretation of F is conceivable as an assignment of B (and vice versa)
6. thus A is reducible to a SAT problem

Corollary

SAT *is* NP*-complete (wrt. the polytime reducibility relation)*

Proof.

1. SAT $\in$ NP follows from Lemma ①, as SAT can be easily encoded as $\exists$SO-formula
2. thus let $A \in$ NP
3. by Fagin's theorem, there exists an $\exists$SO-formula F and some finite structures $\mathcal{K}$, such that A is equivalent to the $F{-}\mathcal{K}$ problem; moreover the first-order part of F is univeral
4. let $\mathcal{M} \in \mathcal{K}$ be a finite; the universal part of F can be represented as propositional formula B
5. any interpretation of F is conceivable as an assignment of B (and vice versa)
6. thus A is reducible to a SAT problem

■

Corollary

the following is equivalent:

- NP $=$ coNP *and*
- $\exists$*SO is equivalent to (full) second-order logic*

Corollary

the following is equivalent:

- NP $=$ coNP *and*
- $\exists$*SO is equivalent to (full) second-order logic*

Proof.

1. any problem in coNP is representable as $\forall$SO formula

Corollary

the following is equivalent:

- NP $=$ coNP *and*
- $\exists SO$ *is equivalent to (full) second-order logic*

Proof.

1. any problem in coNP is representable as $\forall$SO formula
2. thus, if NP $=$ coNP, then $\exists$SO $\equiv$ $\forall$SO

Corollary

the following is equivalent:

- NP $=$ coNP *and*
- $\exists SO$ *is equivalent to (full) second-order logic*

Proof.

1. any problem in coNP is representable as $\forall$SO formula
2. thus, if NP $=$ coNP, then $\exists$SO $\equiv$ $\forall$SO
3. hence, $\exists$SO would be closed under negation and thus equivalent to full second-order logic

Corollary

the following is equivalent:

- NP $=$ coNP *and*
- $\exists SO$ *is equivalent to (full) second-order logic*

Proof.

1. any problem in coNP is representable as $\forall$SO formula
2. thus, if NP $=$ coNP, then $\exists$SO $\equiv \forall$SO
3. hence, $\exists$SO would be closed under negation and thus equivalent to full second-order logic

$\blacksquare$

Corollary

the following is equivalent:

- NP $=$ coNP *and*
- $\exists SO$ *is equivalent to (full) second-order logic*

Proof.

1. any problem in coNP is representable as $\forall$SO formula
2. thus, if NP $=$ coNP, then $\exists$SO $\equiv$ $\forall$SO
3. hence, $\exists$SO would be closed under negation and thus equivalent to full second-order logic

∎

We leave it to the reader to verify and expand upon the claims in this section and to resolve the problems whether P $=$ NP $=$ coNP *(S. Hedman, A First (sic!) Course in Logic)*