CS 511, Fall 2024, Lecture Slides 32

Second Order Logic

Assaf Kfoury

12 November 2024

example

- Let $\varphi \stackrel{\text{def}}{=} \exists y \left(P(y) \to \forall x\, P(x) \right)$

  φ is a first-order sentence over the vocabulary/signature $\Sigma = \{P\}$.

  Is φ semantically valid (true in every model) or, equivalently, formally provable?

example

- Let $\varphi \stackrel{\text{def}}{=} \exists y \left(P(y) \to \forall x\, P(x) \right)$

  φ is a first-order sentence over the vocabulary/signature $\Sigma = \{P\}$.

  Is φ semantically valid (true in every model) or, equivalently, formally provable?

- Yes, it is, no matter the interpretation of the predicate symbol P.

  So why not consider instead the formula $\psi \stackrel{\text{def}}{=} \forall P\, \varphi$?

  ψ is no longer first-order, but a second-order sentence.

example

- Let $\varphi \stackrel{\text{def}}{=} \exists y \Big(P(y) \to \forall x\, P(x) \Big)$

  φ is a first-order sentence over the vocabulary/signature $\Sigma = \{P\}$.

  Is φ semantically valid (true in every model) or, equivalently, formally provable?

- Yes, it is, no matter the interpretation of the predicate symbol P.

  So why not consider instead the formula $\psi \stackrel{\text{def}}{=} \forall P\, \varphi$?

  ψ is no longer first-order, but a second-order sentence.

- Do we have a formal semantics for second-order logic?

  Do we have a formal proof theory / deductive system for second-order logic?

  If the answer is **yes** to both questions, do we have a soundness-and-completeness theorem for second-order logic?

from first-order to second-order logic

Given a vocabulary $\Sigma = \mathcal{P} \cup \mathcal{F} \cup \mathcal{C}$ as before –

$\mathcal{P}$ is a collection of relation/predicate symbols,
$\mathcal{F}$ a collection of function symbols,
$\mathcal{C}$ a collection of constant symbols –

we go from the syntax and formation rules of first-order logic to second-order logic by adding:

- **relation/predicate variables**: $X_1, X_2, \ldots$ each with an arity $n \geqslant 1$.
- **function variables**: $F_1, F_2, \ldots$ each with an arity $n \geqslant 1$.

from first-order to second-order logic

Given a vocabulary $\Sigma = \mathcal{P} \cup \mathcal{F} \cup \mathcal{C}$ as before –

$\mathcal{P}$ is a collection of relation/predicate symbols,
$\mathcal{F}$ a collection of function symbols,
$\mathcal{C}$ a collection of constant symbols –

we go from the syntax and formation rules of first-order logic to second-order logic by adding:

- **relation/predicate variables**: $X_1, X_2, \ldots$ each with an arity $n \geqslant 1$.
- **function variables**: $F_1, F_2, \ldots$ each with an arity $n \geqslant 1$.

The definition of a model $\mathcal{M}$ proceeds as in Lecture Slides 19 (or, alternatively, in Appendix B of EML.Appendix.pdf), except that now an **environment** (or **valuation** or **look-up table**) ℓ must assign a meaning to **relation variables** and **function variables**, in addition to **individual variables**.

from first-order to second-order logic

The only new features in the definition of ***satisfaction*** deal with the second-order quantifiers – see Lecture Slides 19:

- let X be a n-ary predicate variable, for some $n \geqslant 1$,

  $$\mathcal{M}, \ell \models \forall X \, \varphi \quad \text{iff } \mathcal{M}, \ell[X \mapsto R] \models \varphi \text{ for every } R \subseteq \underbrace{A \times \cdots \times A}_{n}$$

- let F be a n-ary function variable, for some $n \geqslant 1$,

  $$\mathcal{M}, \ell \models \forall F \, \varphi \quad \text{iff } \mathcal{M}, \ell[F \mapsto f] \models \varphi \text{ for every } f : \underbrace{A \times \cdots \times A}_{n} \to A$$

- And similarly for the existential second-order quantifiers.

semantic entailment, semantic validity, satisfiability

Let φ be a second-order WFF. Similar to first-order logic, we say:

- WFF φ is **satisfiable** iff there are some $\mathcal{M}$ and ℓ such that $\mathcal{M}, \ell \models \varphi$
- WFF φ is **semantically valid** iff for all $\mathcal{M}$ and ℓ it holds that $\mathcal{M}, \ell \models \varphi$
- If φ is a closed second-order WFF, we write $\mathcal{M} \models \varphi$ instead of $\mathcal{M}, \ell \models \varphi$

semantic entailment, semantic validity, satisfiability

Let φ be a second-order WFF. Similar to first-order logic, we say:

- WFF φ is **satisfiable** iff there are some $\mathcal{M}$ and ℓ such that $\mathcal{M}, \ell \models \varphi$
- WFF φ is **semantically valid** iff for all $\mathcal{M}$ and ℓ it holds that $\mathcal{M}, \ell \models \varphi$
- If φ is a closed second-order WFF, we write $\mathcal{M} \models \varphi$ instead of $\mathcal{M}, \ell \models \varphi$

Let Γ be a set of second-order WFF's:

- Γ is **satisfiable** iff there are some $\mathcal{M}$ and ℓ s.t. $\mathcal{M}, \ell \models \varphi$ for every $\varphi \in \Gamma$
- **semantic entailment**: $\Gamma \models \psi$ iff for every $\mathcal{M}$ and every ℓ, it holds that $\mathcal{M}, \ell \models \Gamma$ implies $\mathcal{M}, \ell \models \psi$

soundness and completeness for second-order logic ???

- There are several deductive systems for second-order logic, but none can be **complete** w.r.t. second-order semantics.
  (Not shown in these lecture slides.)
- At a minimum, each of these deductive systems is **sound**, i.e., any second-order WFF which is formally derivable is semantically valid.
  (Not shown in these lecture slides.)

examples (modeling in second-order logic)

- "A **well-ordering** is an ordering $\leqslant$ such that every non-empty set has a least element w.r.t. $\leqslant$"
- From Lecture Slides 20, page 8: Can first-order logic specify a well-ordering?

examples (modeling in second-order logic)

- "A **well-ordering** is an ordering $\leqslant$ such that every non-empty set has a least element w.r.t. $\leqslant$"
- From Lecture Slides 20, page 8: Can first-order logic specify a well-ordering?
- Second-order logic can express the well-ordering property:

$$\varphi \stackrel{\text{def}}{=} \forall X \left(\exists y\, X(y) \to \exists v \left(X(v) \wedge \forall w \left(X(w) \to v \leqslant w\right)\right) \right)$$

examples (modeling in second-order logic)

- "A **well-ordering** is an ordering $\leqslant$ such that every non-empty set has a least element w.r.t. $\leqslant$"
- From Lecture Slides 20, page 8: Can first-order logic specify a well-ordering?
- Second-order logic can express the well-ordering property:

$$\varphi \stackrel{\text{def}}{=} \forall X \left(\exists y\, X(y) \to \exists v \left(X(v) \land \forall w \left(X(w) \to v \leqslant w \right)\right) \right)$$

- **Fact** (not proved here): The set of sentences

$$\{\varphi\} \cup \mathsf{Th}(\mathcal{N}_1)$$

  defines $\mathcal{N}_1$ (and every structure which is an expansion of $\mathcal{N}_1$) **up to isomorphism**, where $\mathcal{N}_1 \stackrel{\text{def}}{=} (\mathbb{N}, 0, S, <)$ in Lecture Slides 21.
- **Fact** (not proved here): First-order logic cannot specify the well-ordering property, because there are non-isomorphic models of $\mathsf{Th}(\mathcal{N}_1)$, some of which are well-ordered and some are not well-ordered.

examples (modeling in second-order logic)

- A second-order sentence satisfied by a structure $\mathcal{M}$ iff the domain/universe of $\mathcal{M}$ is **infinite**, where X is a binary predicate variable:[1]

$$\Psi_{\text{infinite}} \stackrel{\text{def}}{=} \exists X \Big(\forall x\, \forall y\, \forall z\, \big(X(x,y) \wedge X(y,z) \to X(x,z)\big) \qquad \text{“}X \text{ is transitive”}$$

$$\wedge \quad \forall x\, \big(\neg X(x,x)\big) \qquad \text{“}X \text{ is not reflexive”}$$

$$\wedge \quad \forall x\, \exists y\, X(x,y) \Big) \qquad \text{“every } x \text{ is s.t. } x \xrightarrow{X} y \text{ for some } y\text{”}$$

[1] By definition, the universe of $\mathcal{M}$ of a structure/model, is a non-empty set. Hence, ψ cannot be vacuously true, because all models of ψ have non-empty universes.

examples (modeling in second-order logic)

- A second-order sentence satisfied by a structure $\mathcal{M}$ iff the domain/universe of $\mathcal{M}$ is **infinite**, where X is a binary predicate variable:[1]

$$
\begin{aligned}
\Psi_{\text{infinite}} \stackrel{\text{def}}{=} \exists X \Big(& \forall x \, \forall y \, \forall z \, \big(X(x,y) \wedge X(y,z) \to X(x,z)\big) && \text{“}X\text{ is transitive”} \\
\wedge \quad & \forall x \, \big(\neg X(x,x)\big) && \text{“}X\text{ is not reflexive”} \\
\wedge \quad & \forall x \, \exists y \, X(x,y) \Big) && \text{“every } x \text{ is s.t. } x \xrightarrow{X} y \text{ for some } y\text{”}
\end{aligned}
$$

- Another second-order sentence satisfied by a structure $\mathcal{M}$ iff the domain/universe of $\mathcal{M}$ is **infinite**, where F is a unary function variable:

$$
\begin{aligned}
\Psi'_{\text{infinite}} \stackrel{\text{def}}{=} \exists F \Big(& \forall x \, \forall y \, \forall z \, \big(F(x) \approx z \wedge F(y) \approx z \to x \approx y\big) && \text{“}F\text{ is injective”} \\
\wedge \quad & \exists y \, \forall x \; \neg(F(x) \approx y) \Big) && \text{“}F\text{ is not surjective”}
\end{aligned}
$$

[1] By definition, the universe of $\mathcal{M}$ of a structure/model, is a non-empty set. Hence, ψ cannot be vacuously true, because all models of ψ have non-empty universes.

examples (modeling in second-order logic)

- A second-order sentence satisfied by a structure $\mathcal{M}$ iff the domain/universe of $\mathcal{M}$ is **infinite**, where X is a binary predicate variable:[1]

$$\begin{array}{llr}\Psi_{\text{infinite}} \stackrel{\text{def}}{=} \exists X \Big(\forall x \forall y \forall z \; \big(X(x,y) \wedge X(y,z) \to X(x,z)\big) & & \text{“X is transitive”} \\ \quad \wedge \quad \forall x \; \big(\neg X(x,x)\big) & & \text{“X is not reflexive”} \\ \quad \wedge \quad \forall x \, \exists y \, X(x,y) \Big) & & \text{“every x is s.t. $x \xrightarrow{X} y$ for some y”}\end{array}$$

- Another second-order sentence satisfied by a structure $\mathcal{M}$ iff the domain/universe of $\mathcal{M}$ is **infinite**, where F is a unary function variable:

$$\begin{array}{llr}\Psi'_{\text{infinite}} \stackrel{\text{def}}{=} \exists F \Big(\forall x \forall y \forall z \; \big(F(x) \approx z \wedge F(y) \approx z \to x \approx y\big) & & \text{“F is injective”} \\ \quad \wedge \quad \exists y \, \forall x \; \neg(F(x) \approx y) \Big) & & \text{“F is not surjective”}\end{array}$$

- A second-order sentence Φ_{finite} satisfied by a model $\mathcal{M}$ iff the domain of $\mathcal{M}$ is **finite** is therefore: $\Phi_{\text{finite}} \stackrel{\text{def}}{=} \neg \Psi_{\text{infinite}}$ or $\Phi_{\text{finite}} \stackrel{\text{def}}{=} \neg \Psi'_{\text{infinite}}$.

[1] By definition, the universe of $\mathcal{M}$ of a structure/model, is a non-empty set. Hence, ψ cannot be vacuously true, because all models of ψ have non-empty universes.

compactness and completeness fail for second-order logic

Compactness Theorem for First-Order

Let Γ be a set of first-order sentences.

1. If every finite subset of Γ is **satisfiable**, then so is Γ.
2. If every finite subset of Γ is **consistent**, then so is Γ.

compactness and completeness fail for second-order logic

Compactness Theorem for First-Order

Let Γ be a set of first-order sentences.

1. If every finite subset of Γ is **satisfiable**, then so is Γ.
2. If every finite subset of Γ is **consistent**, then so is Γ.

Counter-Example for Second-Order Compactness

For every $n \geqslant 1$, define the first-order sentence θ_n by:

$$\theta_n \stackrel{\text{def}}{=} \quad \text{"there are at least } n \text{ distinct elements"}$$

Consider the set of sentences:

$$\Delta = \{\neg\psi\} \cup \{\theta_1, \theta_2, \theta_3, ...\}$$

Every finite subset of Δ is **satisfiable**, while Δ is **unsatisfiable**.

compactness and completeness fail for second-order logic

- There are deductive systems (*i.e.*, formal proof theories) for second-order logic, but none can be complete (for the standard semantics).

  In contrast to first-order logic:
  "There are deductive systems for first-order logic which are complete."

- There are sets Γ of second-order sentences which, although consistent (*i.e.*, $\perp$ cannot be formally deduced from Γ), do not have models.

  In contrast to first-order logic:
  "Every consistent set of first-order sentences has a model."

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- "A **Hamiltonian path** is a path that visits every node exactly once"

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- "A **Hamiltonian path** is a path that visits every node exactly once"

$$\Phi_{\text{Ham}} \stackrel{\text{def}}{=} \exists X \Big(\text{“}X \text{ is a linear order”} \wedge \forall x \forall y \left(\text{“}y = x + 1\text{”} \rightarrow R(x, y) \right) \Big)$$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- "A **Hamiltonian path** is a path that visits every node exactly once"

$$\Phi_{\mathsf{Ham}} \stackrel{\text{def}}{=} \exists X \Big(\text{“}X \text{ is a linear order”} \wedge \forall x \forall y \, (\text{“}y = x + 1\text{”} \rightarrow R(x, y)) \Big)$$

$$\Phi_{\mathsf{Ham}} \stackrel{\text{def}}{=} \exists X \Big(\; \psi_1(X) \; \wedge \; \forall x \forall y \, (\; \psi_2(X, x, y) \; \rightarrow R(x, y)) \Big)$$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- "A **Hamiltonian path** is a path that visits every node exactly once"

$$\Phi_{\text{Ham}} \stackrel{\text{def}}{=} \exists X \Big(\text{“}X \text{ is a linear order”} \wedge \forall x \forall y \, (\text{“}y = x + 1\text{”} \to R(x, y)) \Big)$$

$$\Phi_{\text{Ham}} \stackrel{\text{def}}{=} \exists X \Big(\; \psi_1(X) \; \wedge \; \forall x \forall y \, (\; \psi_2(X, x, y) \; \to R(x, y)) \Big)$$

$\psi_1(X)$ makes predicate-variable X a linear order:

$$\begin{array}{rll} \psi_1(X) \stackrel{\text{def}}{=} & \forall x \, X(x, x) \wedge & \text{reflexivity} \\ & \forall x \forall y \forall z \, \big(X(x, y) \wedge X(y, z) \to X(x, z)\big) \wedge & \text{transitivity} \\ & \forall x \forall y \, \big(X(x, y) \wedge X(y, x) \to x \approx y\big) \wedge & \text{anti-symmetry} \\ & \forall x \forall y \, \big(X(x, y) \vee X(y, x)\big) & \text{totality} \end{array}$$

examples with graphs (A,R)

where A is the set of nodes and R is a binary relation representing edges

- "A **Hamiltonian path** is a path that visits every node exactly once"

$$\Phi_{\text{Ham}} \stackrel{\text{def}}{=} \exists X \Big(\text{"}X \text{ is a linear order"} \wedge \forall x \forall y \,(\text{"}y = x + 1\text{"} \rightarrow R(x,y)) \Big)$$

$$\Phi_{\text{Ham}} \stackrel{\text{def}}{=} \exists X \Big(\; \psi_1(X) \; \wedge \; \forall x \forall y \,(\; \psi_2(X,x,y) \; \rightarrow R(x,y)) \Big)$$

$\psi_1(X)$ makes predicate-variable X a linear order:

$$\begin{array}{lll}
\psi_1(X) \stackrel{\text{def}}{=} & \forall x \, X(x,x) \wedge & \text{reflexivity} \\
& \forall x \forall y \forall z \, \big(X(x,y) \wedge X(y,z) \rightarrow X(x,z)\big) \wedge & \text{transitivity} \\
& \forall x \forall y \, \big(X(x,y) \wedge X(y,x) \rightarrow x \approx y\big) \wedge & \text{anti-symmetry} \\
& \forall x \forall y \, \big(X(x,y) \vee X(y,x)\big) & \text{totality}
\end{array}$$

$\psi_2(X,x,y)$ is a WFF with free predicate-variable X of arity 2 and first-order variables x and y, which makes y the successor of x in the linear order X:

$$\psi_2(X,x,y) \stackrel{\text{def}}{=} \neg(x \approx y) \wedge X(x,y) \wedge \forall z \, \big(X(x,z) \wedge X(z,y) \rightarrow (x \approx z \vee y \approx z)\big)$$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **2-colorability**:
  represent color 1 by unary predicate X, and color 2 by $\neg X$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **2-colorability**:
  represent color 1 by unary predicate X, and color 2 by $\neg X$

$$\Phi_{\text{2-color}} \stackrel{\text{def}}{=} \exists X \forall x \forall y \Big(\neg(x \approx y) \wedge R(x,y) \to (X(x) \leftrightarrow \neg X(y)) \Big)$$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **3-colorability**:
  represent 3 colors by unary predicate variables X_1, X_2, and X_3

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **3-colorability**:
  represent 3 colors by unary predicate variables X_1, X_2, and X_3
- ψ_1 says "each node has exactly one color":

$$\psi_1(X_1, X_2, X_3) \stackrel{\text{def}}{=} \forall x \Big(\big(\boxed{X_1(x)} \wedge \neg X_2(x) \wedge \neg X_3(x)\big) \vee \big(\neg X_1(x) \wedge \boxed{X_2(x)} \wedge \neg X_3(x)\big) \vee \big(\neg X_1(x) \wedge \neg X_2(x) \wedge \boxed{X_3(x)}\big) \Big)$$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **3-colorability**:
  represent 3 colors by unary predicate variables X_1, X_2, and X_3
- ψ_1 says "each node has exactly one color":

$$\psi_1(X_1, X_2, X_3) \stackrel{\text{def}}{=} \forall x \Big(\big(\boxed{X_1(x)} \wedge \neg X_2(x) \wedge \neg X_3(x)\big) \vee \\ \big(\neg X_1(x) \wedge \boxed{X_2(x)} \wedge \neg X_3(x)\big) \vee \\ \big(\neg X_1(x) \wedge \neg X_2(x) \wedge \boxed{X_3(x)}\big) \Big)$$

- ψ_2 says "no two points with the same color are connected":

$$\psi_2(X_1, X_2, X_3) \stackrel{\text{def}}{=} \forall x \forall y \Big(\big(X_1(x) \wedge X_1(y) \to \neg R(x, y)\big) \wedge \\ \big(X_2(x) \wedge X_2(y) \to \neg R(x, y)\big) \wedge \\ \big(X_3(x) \wedge X_3(y) \to \neg R(x, y)\big) \Big)$$

examples with graphs (A,R)

where A is the set of nodes and R is a binary relation representing edges

- **3-colorability**:
  represent 3 colors by unary predicate variables X_1, X_2, and X_3
- ψ_1 says "each node has exactly one color":

$$
\begin{aligned}
\psi_1(X_1,X_2,X_3) \stackrel{\text{def}}{=} \forall x \Big(& \big(\boxed{X_1(x)} \wedge \neg X_2(x) \wedge \neg X_3(x)\big) \vee \\
& \big(\neg X_1(x) \wedge \boxed{X_2(x)} \wedge \neg X_3(x)\big) \vee \\
& \big(\neg X_1(x) \wedge \neg X_2(x) \wedge \boxed{X_3(x)}\big) \Big)
\end{aligned}
$$

- ψ_2 says "no two points with the same color are connected":

$$
\begin{aligned}
\psi_2(X_1,X_2,X_3) \stackrel{\text{def}}{=} \forall x \forall y \Big(& \big(X_1(x) \wedge X_1(y) \to \neg R(x,y)\big) \wedge \\
& \big(X_2(x) \wedge X_2(y) \to \neg R(x,y)\big) \wedge \\
& \big(X_3(x) \wedge X_3(y) \to \neg R(x,y)\big) \Big)
\end{aligned}
$$

- $\boxed{\Phi_{\text{3-color}} \stackrel{\text{def}}{=} \exists X_1 \exists X_2 \exists X_3 \, (\psi_1 \wedge \psi_2)}$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **unconnectedness**

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **unconnectedness**
- ψ_1 says "the set X is non-empty and its complement is nonempty"

$$\psi_1(X) \stackrel{\text{def}}{=} \exists x \exists y \ \big(\ X(x) \wedge \neg X(y)\ \big)$$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **unconnectedness**
- ψ_1 says "the set X is non-empty and its complement is nonempty"

$$\psi_1(X) \stackrel{\text{def}}{=} \exists x \exists y \; \big(\; X(x) \wedge \neg X(y) \;\big)$$

- ψ_2 says "there is no edge between X and its complement"

$$\psi_2(X) \stackrel{\text{def}}{=} \forall x \forall y \; \Big(\; \big(X(x) \wedge \neg X(y)\big) \rightarrow \big(\neg R(x,y) \wedge \neg R(y,x)\big) \;\Big)$$

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **unconnectedness**
- ψ_1 says "the set X is non-empty and its complement is nonempty"

$$\psi_1(X) \stackrel{\text{def}}{=} \exists x \exists y \; \big(\; X(x) \wedge \neg X(y) \; \big)$$

- ψ_2 says "there is no edge between X and its complement"

$$\psi_2(X) \stackrel{\text{def}}{=} \forall x \forall y \; \Big(\; \big(X(x) \wedge \neg X(y)\big) \to \big(\neg R(x,y) \wedge \neg R(y,x)\big) \; \Big)$$

- $\Phi_{\text{unconnect}} \stackrel{\text{def}}{=} \exists X \, (\psi_1 \wedge \psi_2)$
  is true iff graph **is not connected**

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **unconnectedness**
- ψ_1 says “the set X is non-empty and its complement is nonempty”

$$\psi_1(X) \stackrel{\text{def}}{=} \exists x \exists y \ \big(\ X(x) \wedge \neg X(y)\ \big)$$

- ψ_2 says “there is no edge between X and its complement”

$$\psi_2(X) \stackrel{\text{def}}{=} \forall x \forall y \ \Big(\ \big(X(x) \wedge \neg X(y)\big) \to \big(\neg R(x,y) \wedge \neg R(y,x)\big)\ \Big)$$

- $\Phi_{\text{unconnect}} \stackrel{\text{def}}{=} \exists X\, (\psi_1 \wedge \psi_2)$
  is true iff graph **is not connected**

- $\Phi_{\text{connect}} \stackrel{\text{def}}{=} \neg \Phi_{\text{unconnect}} \stackrel{\text{def}}{=} \forall X\, (\neg\psi_1 \vee \neg\psi_2) \stackrel{\text{def}}{=} \forall X\, (\psi_1 \to \neg\psi_2)$
  is true iff graph **is connected**

examples with graphs (A, R)

where A is the set of nodes and R is a binary relation representing edges

- **reachability**

  Example 2.27 in [LCS. page 140].

Useful Abbreviations

- When stating facts about sets, it is convenient to use "$\subseteq$" and "$\in$" but these are not part of the official syntax of second-order logic. Nonetheless, they can be viewed as "sugar" or "macros" of longer expressions in the official syntax.

Useful Abbreviations

- When stating facts about sets, it is convenient to use "$\subseteq$" and "$\in$" but these are not part of the official syntax of second-order logic. Nonetheless, they can be viewed as "sugar" or "macros" of longer expressions in the official syntax.

- Given set variables X and Y, *i.e.*, X and Y are also unary predicate-variables:
  - $x \in X$ is sugar for $X(x)$.
  - $X \subseteq Y$ is sugar for $\forall x.\ (x \in X \to x \in Y)$.
  - $\forall x.\ (x \in X \to x \in Y)$ is sugar for $\forall x.\ (X(x) \to Y(x))$.

Useful Abbreviations

- When stating facts about sets, it is convenient to use “$\subseteq$” and “$\in$” but these are not part of the official syntax of second-order logic. Nonetheless, they can be viewed as “sugar” or “macros” of longer expressions in the official syntax.

- Given set variables X and Y, *i.e.*, X and Y are also unary predicate-variables:
  - $x \in X$ is sugar for $X(x)$.
  - $X \subseteq Y$ is sugar for $\forall x.\ (x \in X \to x \in Y)$.
  - $\forall x.\ (x \in X \to x \in Y)$ is sugar for $\forall x.\ (X(x) \to Y(x))$.

- We can also de-sugar ***relativized*** quantifiers as follows:
  - $\forall x \in X.\varphi$ is sugar for $\forall x.(x \in X \to \varphi)$ and
    $\forall X \subseteq Y.\varphi$ is sugar for $\forall X.(X \subseteq Y \to \varphi)$
  - $\exists x \in X.\varphi$ is sugar for $\exists x.(x \in X \wedge \varphi)$ and
    $\exists X \subseteq Y.\varphi$ is sugar for $\exists X.(X \subseteq Y \wedge \varphi)$

More Useful Abbreviations

- The cardinality of set B is larger than, or equal to, the cardinality of set A, which we can write as $A \preccurlyeq B$, iff there is a surjection from B to A.

More Useful Abbreviations

- The cardinality of set B is larger than, or equal to, the cardinality of set A, which we can write as $A \preccurlyeq B$, iff there is a surjection from B to A.

- In second-order logic, we can express $X \preccurlyeq Y$, where X and Y are set variables (aka unary predicate-variables), by making it sugar for the following wff:

  $X \preccurlyeq Y$ is sugar for $\exists G.\ \forall x \in X.\ \exists y \in Y.\ G(y) \approx x$

More Useful Abbreviations

- The cardinality of set B is larger than, or equal to, the cardinality of set A, which we can write as $A \preccurlyeq B$, iff there is a surjection from B to A.

- In second-order logic, we can express $X \preccurlyeq Y$, where X and Y are set variables (aka unary predicate-variables), by making it sugar for the following wff:

  $X \preccurlyeq Y$ is sugar for $\exists G.\ \forall x \in X.\ \exists y \in Y.\ G(y) \approx x$

- If both $X \preccurlyeq Y$ and $Y \preccurlyeq X$, we also introduce:

  $X \sim Y$ is sugar for $(X \preccurlyeq Y) \wedge (Y \preccurlyeq X)$ which is sugar for

  $(\exists G.\forall x \in X.\exists y \in Y.\ G(y) \approx x) \wedge (\exists F.\forall y \in Y.\exists x \in X.\ F(x) \approx y)$

Exercise: *Define $X \sim Y$ differently in second-order logic by asserting the existence of a unary function F from X to Y which is both injective and surjective.*

Infinite and Countably Infinite

- We can relativize the wff Ψ'_{infinite} defined on slides 14-15-16 to express that the subset X of the universe is infinite (we can just as well relativize Ψ_{infinite} instead of Ψ'_{infinite}):

$$\begin{array}{lll} \Phi_{\text{infty}}(X) \stackrel{\text{def}}{=} \exists F \Big(\forall x \in X.\forall y \in X.\forall z \in X.\big(F(x) \approx z \wedge F(y) \approx z \to x \approx y\big) & & \text{“}F\text{ injective on }X\text{”} \\ \qquad\qquad \wedge \quad \exists y \in X.\forall x \in X.\neg(F(x) \approx y) \Big) & & \text{“}F\text{ not surjective on }X\text{”} \end{array}$$

Infinite and Countably Infinite

- We can relativize the wff Ψ'_{infinite} defined on slides 14-15-16 to express that the subset X of the universe is infinite (we can just as well relativize Ψ_{infinite} instead of Ψ'_{infinite}):

$$\begin{aligned}\Phi_{\text{infty}}(X) \stackrel{\text{def}}{=} \exists F \Big(& \forall x \in X. \forall y \in X. \forall z \in X. \big(F(x) \approx z \wedge F(y) \approx z \to x \approx y\big) && \text{“}F \text{ injective on } X\text{”} \\ \wedge \quad & \exists y \in X. \forall x \in X. \neg (F(x) \approx y) \Big) && \text{“}F \text{ not surjective on } X\text{”}\end{aligned}$$

$\Phi_{\text{infty}}(X)$ above is a relativized version of Ψ'_{infty} on slides 14-15-16, here recalled:

$$\begin{aligned}\Psi'_{\text{infty}} \stackrel{\text{def}}{=} \exists F \Big(& \forall x \, \forall y \, \forall z \; \big(F(x) \approx z \wedge F(y) \approx z \to x \approx y\big) && \text{“}F \text{ is injective”} \\ \wedge \quad & \exists y \, \forall x \; \neg (F(x) \approx y) \Big) && \text{“}F \text{ is not surjective”}\end{aligned}$$

Infinite and Countably Infinite

- We can relativize the wff Ψ'_{infinite} defined on slides 14-15-16 to express that the subset X of the universe is infinite (we can just as well relativize Ψ_{infinite} instead of Ψ'_{infinite}):

$$\Phi_{\text{infty}}(X) \stackrel{\text{def}}{=} \exists F \Big(\forall x \in X. \forall y \in X. \forall z \in X. \big(F(x) \approx z \wedge F(y) \approx z \to x \approx y\big) \quad \text{“}F \text{ injective on } X\text{”}$$

$$\wedge \quad \exists y \in X. \forall x \in X. \neg(F(x) \approx y) \Big) \quad \text{“}F \text{ not surjective on } X\text{”}$$

$\Phi_{\text{infty}}(X)$ above is a relativized version of Ψ'_{infty} on slides 14-15-16, here recalled:

$$\Psi'_{\text{infty}} \stackrel{\text{def}}{=} \exists F \Big(\forall x \, \forall y \, \forall z \, \big(F(x) \approx z \wedge F(y) \approx z \to x \approx y\big) \quad \text{“}F \text{ is injective”}$$

$$\wedge \quad \exists y \, \forall x \, \neg(F(x) \approx y) \Big) \quad \text{“}F \text{ is not surjective”}$$

- Hence, we also have the wff $\Phi_{\text{finite}}(X)$ relativized w.r.t. X to express that the subset X of the universe is finite:

  $\Phi_{\text{finite}}(X) \stackrel{\text{def}}{=} \neg \Phi_{\text{infty}}(X)$

Infinite and Countably Infinite

- **FACT**: **A set Y is countably infinite if Y is infinite and for every infinite subset X of Y there is a bijection from X to Y.**

  Hence, we also have the wff $\Phi_{\text{countable-infty}}(Y)$ to express that subset Y is countably infinite:

  $$\Phi_{\text{countable-infty}}(Y) \stackrel{\text{def}}{=} \Phi_{\text{infty}}(Y) \ \wedge \ \big(\forall X \subseteq Y.\, \Phi_{\text{infty}}(X) \to (X \sim Y)\big)$$

Infinite and Countably Infinite

- **FACT**: **A set Y is countably infinite if Y is infinite and for every infinite subset X of Y there is a bijection from X to Y.**

  Hence, we also have the wff $\Phi_{\text{countable-infty}}(Y)$ to express that subset Y is countably infinite:

  $$\Phi_{\text{countable-infty}}(Y) \stackrel{\text{def}}{=} \Phi_{\text{infty}}(Y) \;\wedge\; \big(\forall X \subseteq Y.\; \Phi_{\text{infty}}(X) \to (X \sim Y)\big)$$

Exercise:

1. Define a 2nd-order sentence $\Psi_{\text{countable-infty}}$ s.t. $\mathcal{A} \models \Psi_{\text{countable-infty}}$ iff $\mathcal{A}$ is countably infinite.
2. Define a 2nd-order sentence $\Psi_{\text{uncountable}}$ s.t. $\mathcal{A} \models \Psi_{\text{uncountable}}$ iff $\mathcal{A}$ is uncountably infinite.

Note that $\Psi_{\text{countable-infty}}$ and $\Psi_{\text{uncountable}}$ in this exercise are sentences, *i.e.*, closed wff's which do not contain any free variables.

More Useful Abbreviations

- In first-order logic, the equality relation (which is always the interpretation of the symbol $\approx$) is undefinable.
- Is this still the case in second-order logic, *i.e.*, that the equality relation is undefinable?

  Looking back at all the slides in the present set, "$\approx$" appears more than a dozen times – and we never questioned whether or not it must be a primitive relation.

More Useful Abbreviations

- In first-order logic, the equality relation (which is always the interpretation of the symbol $\approx$) is undefinable.
- Is this still the case in second-order logic, *i.e.*, that the equality relation is undefinable?

  Looking back at all the slides in the present set, "$\approx$" appears more than a dozen times – and we never questioned whether or not it must be a primitive relation.
- In fact, it turns out that the equality relation is second-order definable!

  $x \approx y$ is sugar for $\forall X.\ X(x) \leftrightarrow X(y)$

  where $\{x, y\}$ are first-order variables and X is a unary predicate variable. In words,

  "x and y are *identical* iff x and y satisfy the same unary predicates"

More Useful Abbreviations

- **Exercise:**

  1. Put differently, the preceding definition of "$\approx$" says that:
     $x \approx y$ iff "*no unary predicate X can discern x and y*",
     i.e., the English phrase to the right of "iff" is modeled by the second-order wff $(\forall X.\ X(x) \leftrightarrow X(y))$.

     Write a second-order wff $\theta(x, y)$ such that:
     $\theta(x, y)$ iff "*no binary predicate Y can discern x and y*".
     Your task here is to write a wff of second-order logic modeling the English phrase to the right of "iff".

  2. Give a precise (informal) argument that the following second-order sentence is semantically valid:

     $$\forall x.\forall y.\ (\forall X.\ X(x) \leftrightarrow X(y)) \ \rightarrow\ \theta(x, y)$$

     i.e., given arbitrary x and y, if no unary predicate X can discern x and y, then no binary predicate Y can discern x and y.

connections with *descriptive complexity theory*

- Starting point:

  Syntactic classification of second-order WFF's in prenex normal form, over a given signature Σ, according to:

  1. interleaving of universal and existential quantifiers in the prenex, and
  2. arities of predicate and function symbols in Σ.

- **Example**:
  The WFF φ in each of slide 24, slide 26, slide 30, and slide 34, is an existential second-order WFF.

- **Example**:
  The φ in each of slide 26, slide 30, and slide 34, but not on slide 24, is a monadic second-order WFF, because the second-order variables in φ are restricted to be unary-predicate (*i.e.*, set) variables.

- **Example**:
  Monadic second-order logic has been extensively studied in relation to graph properties and their complexities. (Search the WWW with the keyword "monadic second-order logic.")

connections with *descriptive complexity theory*

- Prototypical result of descriptive complexity theory:

  Fagin's theorem: Let $\mathcal{C}$ be the class of all finite undirected graphs (closed under isomorphism). The following are equivalent statements:

  1. $\mathcal{C}$ is in NP.
  2. $\mathcal{C}$ is definable by an existential second-order sentence.

  In fact, every class of objects in NP has an existential second-order characterization with binary predicates and a universal first-order part.

(THIS PAGE INTENTIONALLY LEFT BLANK)