CS 511, Fall 2018, Handout 36

Second Order Logic

Assaf Kfoury

5 December 2018

example

- Let $\varphi \triangleq \exists y \Big(P(y) \to \forall x\, P(x) \Big)$

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

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

example

- Let $\varphi \triangleq \exists y \left(P(y) \to \forall x \, P(x) \right)$

  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 $\psi \triangleq \forall P \, \varphi$?

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

example

- Let $\varphi \triangleq \exists y \Big(P(y) \to \forall x\, P(x) \Big)$

  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 $\psi \triangleq \forall P\, \varphi$?

  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 a have 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 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:

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

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

The definition of a model $\mathcal{M}$ proceeds as in Handout 17, except that now an **environment** (or **look-up table**) ℓ must assign a meaning to **predicate 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 Handout 17:

- 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 $n \geqslant 1$,

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

semantic entailment, semantic validity, satisfiability

Let φ be a second-order WFF. Similar to 1st 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 1st 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 this handout.)
- 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 this handout.)

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 Handout 18, 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 Handout 18, page 8: Can first-order logic specify a well-ordering?
- Second-order logic can express the well-ordering property:

$$\varphi \triangleq \forall X \left(\exists y\, X(y) \to \exists v \,\big(X(v) \wedge \forall w\, (X(w) \to v \leqslant w)\big) \right)$$

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 Handout 18, page 8: Can first-order logic specify a well-ordering?
- Second-order logic can express the well-ordering property:

$$\varphi \triangleq \forall X \Big(\exists y\, X(y) \to \exists v\, \big(X(v) \land \forall w\, (X(w) \to v \leqslant w)\big) \Big)$$

- **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 $\mathcal{N}_1 \triangleq (\mathbb{N}, 0, S, <)$ in Handout 23.
- **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**:[1]

$$\begin{array}{llll} \psi \triangleq \exists P \Big(& \forall x \, \forall y \, \forall z \, \big(P(x,y) \wedge P(y,z) \to P(x,z)\big) & & \text{“}P\text{ is transitive”} \\ & \wedge \quad \forall x \, \big(\neg P(x,x)\big) & & \text{“}P\text{ is not reflexive”} \\ & \wedge \quad \forall x \, \exists y \, P(x,y) \ \Big) & & \text{“every } x \text{ is s.t. } x \xrightarrow{P} y \text{ for some } y\text{”} \end{array}$$

[1] By definition, the universe of $\mathcal{M}$, 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**:[1]

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

- A second-order sentence satisfied by a model $\mathcal{M}$ iff the domain of $\mathcal{M}$ is **finite**:

$\neg\psi$

[1] By definition, the universe of $\mathcal{M}$, 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:

$$\theta_n \triangleq \quad \text{“there are at least } n \text{ distinct elements”}$$

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"

$$\varphi \triangleq \exists P\Big(\text{“}P\text{ is a linear order”} \land \forall x \forall y\,(\text{“}y = x + 1\text{”} \to R(x, y))\Big)$$

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”

$$\varphi \triangleq \exists P\Big(\text{“}P\text{ is a linear order”} \wedge \forall x \forall y\,(\text{“}y = x + 1\text{”} \to R(x, y))\Big)$$

$$\varphi \triangleq \exists P\Big(\;\psi_1(P)\;\wedge\;\forall x \forall y\,(\;\psi_2(P, x, y)\;\to R(x, y))\Big)$$

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"

$$\varphi \triangleq \exists P \Big(\text{“}P \text{ is a linear order”} \wedge \forall x \forall y \left(\text{“}y = x + 1\text{”} \rightarrow R(x, y) \right) \Big)$$

$$\varphi \triangleq \exists P \Big(\; \psi_1(P) \;\; \wedge \;\; \forall x \forall y \left(\; \psi_2(P, x, y) \;\; \rightarrow R(x, y) \right) \Big)$$

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

$$\begin{array}{lll} \psi_1(P) \triangleq & \forall x \, P(x, x) \wedge & \text{reflexivity} \\ & \forall x \forall y \forall z \, \big(P(x, y) \wedge P(y, z) \rightarrow P(x, z) \big) \wedge & \text{transitivity} \\ & \forall x \forall y \, \big(P(x, y) \wedge P(y, x) \rightarrow x \doteq y \big) \wedge & \text{anti-symmetry} \\ & \forall x \forall y \, \big(P(x, y) \vee P(y, x) \big) & \text{totality} \end{array}$$

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"

$$\varphi \triangleq \exists P\Big(\text{“}P \text{ is a linear order”} \wedge \forall x \forall y\, (\text{“}y = x + 1\text{”} \to R(x, y))\Big)$$

$$\varphi \triangleq \exists P\Big(\ \psi_1(P)\ \wedge\ \forall x \forall y\, (\ \psi_2(P, x, y)\ \to R(x, y))\Big)$$

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

$$\begin{array}{lll} \psi_1(P) \triangleq & \forall x\, P(x, x)\ \wedge & \text{reflexivity} \\ & \forall x \forall y \forall z\, \big(P(x, y) \wedge P(y, z) \to P(x, z)\big)\ \wedge & \text{transitivity} \\ & \forall x \forall y\, \big(P(x, y) \wedge P(y, x) \to x \doteq y\big)\ \wedge & \text{anti-symmetry} \\ & \forall x \forall y\, \big(P(x, y) \vee P(y, x)\big) & \text{totality} \end{array}$$

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

$$\psi_2(P, x, y) \triangleq \neg(x \doteq y) \wedge P(x, y) \wedge \forall z\, \big(P(x, z) \wedge P(z, y) \to (x \doteq z \vee y \doteq z)\big)$$

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 P, and color 2 by $\neg P$

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 P, and color 2 by $\neg P$

  $$\varphi \triangleq \exists P \forall x \forall y \Big(\neg(x \doteq y) \wedge R(x,y) \to (P(x) \leftrightarrow \neg P(y)) \Big)$$

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 A_1, A_2, and A_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 A_1, A_2, and A_3
- ψ_1 says "each node has exactly one color":

$$\psi_1(A_1, A_2, A_3) \triangleq \forall x \Big(\big(\boxed{A_1(x)} \wedge \neg A_2(x) \wedge \neg A_3(x) \big) \vee \\ \big(\neg A_1(x) \wedge \boxed{A_2(x)} \wedge \neg A_3(x) \big) \vee \\ \big(\neg A_1(x) \wedge \neg A_2(x) \wedge \boxed{A_3(x)} \big) \Big)$$

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 A_1, A_2, and A_3
- ψ_1 says "each node has exactly one color":

$$\begin{aligned}\psi_1(A_1, A_2, A_3) \triangleq \forall x \Big(& \big(\boxed{A_1(x)} \wedge \neg A_2(x) \wedge \neg A_3(x)\big) \vee \\ & \big(\neg A_1(x) \wedge \boxed{A_2(x)} \wedge \neg A_3(x)\big) \vee \\ & \big(\neg A_1(x) \wedge \neg A_2(x) \wedge \boxed{A_3(x)}\big) \Big)\end{aligned}$$

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

$$\begin{aligned}\psi_2(A_1, A_2, A_3) \triangleq \forall x \forall y \Big(& \big(A_1(x) \wedge A_1(y) \to \neg R(x, y)\big) \wedge \\ & \big(A_2(x) \wedge A_2(y) \to \neg R(x, y)\big) \wedge \\ & \big(A_3(x) \wedge A_3(y) \to \neg R(x, y)\big) \Big)\end{aligned}$$

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 A_1, A_2, and A_3
- ψ_1 says "each node has exactly one color":

$$\begin{aligned}\psi_1(A_1, A_2, A_3) \triangleq \forall x \Big(& \big(\boxed{A_1(x)} \wedge \neg A_2(x) \wedge \neg A_3(x) \big) \vee \\ & \big(\neg A_1(x) \wedge \boxed{A_2(x)} \wedge \neg A_3(x) \big) \vee \\ & \big(\neg A_1(x) \wedge \neg A_2(x) \wedge \boxed{A_3(x)} \big) \Big)\end{aligned}$$

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

$$\begin{aligned}\psi_2(A_1, A_2, A_3) \triangleq \forall x \forall y \Big(& \big(A_1(x) \wedge A_1(y) \to \neg R(x, y) \big) \wedge \\ & \big(A_2(x) \wedge A_2(y) \to \neg R(x, y) \big) \wedge \\ & \big(A_3(x) \wedge A_3(y) \to \neg R(x, y) \big) \Big)\end{aligned}$$

- $\varphi \triangleq \exists A_1 \exists A_2 \exists A_3 \, (\psi_1 \wedge \psi_2)$

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 A is non-empty and its complement is nonempty"

$$\psi_1(A) \triangleq \exists x \exists y \left(A(x) \wedge \neg A(y) \right)$$

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 A is non-empty and its complement is nonempty”

$$\psi_1(A) \;\triangleq\; \exists x \exists y\ \big(\ A(x) \land \neg A(y)\ \big)$$

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

$$\psi_2(A) \;\triangleq\; \forall x \forall y\ \Big(\ \big(A(x) \land \neg A(y)\big) \to \big(\neg R(x, y) \land \neg R(y, x)\big)\ \Big)$$

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 A is non-empty and its complement is nonempty"

$$\psi_1(A) \triangleq \exists x \exists y \left(A(x) \wedge \neg A(y) \right)$$

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

$$\psi_2(A) \triangleq \forall x \forall y \Big(\big(A(x) \wedge \neg A(y)\big) \to \big(\neg R(x, y) \wedge \neg R(y, x)\big) \Big)$$

- $\varphi \triangleq \exists A \, (\psi_1 \wedge \psi_2)$
  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 A is non-empty and its complement is nonempty"

$$\psi_1(A) \triangleq \exists x \exists y \left(A(x) \wedge \neg A(y) \right)$$

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

$$\psi_2(A) \triangleq \forall x \forall y \Big(\big(A(x) \wedge \neg A(y)\big) \rightarrow \big(\neg R(x,y) \wedge \neg R(y,x)\big) \Big)$$

- $\varphi \triangleq \exists A\,(\psi_1 \wedge \psi_2)$
  is true iff graph **is not connected**

- $\varphi' \triangleq \neg\varphi \triangleq \forall A\,(\neg\psi_1 \vee \neg\psi_2) \triangleq \forall A\,(\psi_1 \rightarrow \neg\psi_2)$
  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].

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 on slide 23, slide 25, slide 29, and slide 33, is an existential second-order WFF.

- **Example**:
  The φ in each of slide 25, slide 29, and slide 33, but not on slide 23, 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)