CS 512, Spring 2016, Handout 16

Second Order Logic (continued)

Assaf Kfoury

22 February 2016   (Last modified: 1 March 2016)
There are several deductive systems for second-order logic, but none can be complete w.r.t. second-order semantics. At a minimum, each of these deductive systems is sound, i.e., any second-order WFF which is formally derivable is semantically valid.
There are several deductive systems for second-order logic, but none can be complete w.r.t. second-order semantics.
There are several deductive systems for second-order logic, but none can be complete w.r.t. second-order semantics.

At a minimum, each of these deductive systems is sound, i.e., any second-order WFF which is formally derivable is semantically valid.
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

Exercise:
When can we omit 
\(\neg (x = y)\) in 
\(\psi_2(P, x, y)\)? Explain.
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- “A \textbf{Hamiltonian path} is a path that visits every node exactly once”
examples about 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”
  
  \[ \varphi \triangleq \exists P \left[ \text{“} P \text{ is a linear order”} \land \forall x \forall y \left( y = x + 1 \implies R(x, y) \right) \right] \]
examples about 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[\text{“}P\text{ is a linear order”} \land \forall x \forall y (y = x + 1 \rightarrow R(x, y))] \\
\varphi \triangleq \exists P[\psi_1(P) \land \forall x \forall y (\psi_2(P, x, y) \rightarrow R(x, y))] \\
\[
\psi_1(P) \text{ makes predicate-variable } P \text{ a linear order:} \\
\psi_1(P) \triangleq [\forall x P(x, x) \land \text{reflexivity}] \\
\land [\forall x \forall y \forall z (P(x, y) \land P(y, z) \rightarrow P(x, z)) \land \text{transitivity}] \\
\land [\forall x \forall y (P(x, y) \lor P(y, x)) \land \text{totality}] \\
\psi_2(P, x, y) \text{ is a WFF with free predicate-variable } P \text{ of arity } 2 \text{ and first-order variables } x \text{ and } y, \text{ which makes } y \text{ the successor of } x \text{ in the linear order } P:
\psi_2(P, x, y) \triangleq \neg (x = y) \land P(x, y) \land \forall z ([P(x, z) \land P(z, y) \rightarrow (x = z \lor y = z)]) \\
\]

Exercise: When can we omit “\(\neg (x = y)\)” in \(\psi_2(P, x, y)\)? Explain.
examples about 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”

\[
\varphi \triangleq \exists P[\text{“}P\text{ is a linear order”} \land \forall x \forall y (\text{“}y = x + 1\text{”} \to R(x, y))] \\
\varphi \triangleq \exists P[\psi_1(P) \land \forall x \forall y (\psi_2(P, x, y) \to R(x, y))] \\
\psi_1(P) \text{ makes predicate-variable } P \text{ a linear order:}
\]

\[
\psi_1(P) \triangleq [\forall x P(x, x)] \land \\
[\forall x \forall y \forall z (P(x, y) \land P(y, z) \to P(x, z))] \land \\
[\forall x \forall y (P(x, y) \land P(y, x) \to x = y)] \land \\
[\forall x \forall y (P(x, y) \lor P(y, x))]
\]

reflexivity

transitivity

anti-symmetry

totality
examples about 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

\[
\phi \triangleq \exists P[ \text{"P is a linear order"} \land \forall x \forall y (\text{"}y = x + 1\text{"} \rightarrow R(x, y))] \\
\phi \triangleq \exists P[ \psi_1(P) \land \forall x \forall y (\psi_2(P, x, y) \rightarrow R(x, y))] \\
\]

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

\[
\psi_1(P) \triangleq [\forall x P(x, x)] \land \text{reflexivity} \\
[\forall x \forall y \forall z (P(x, y) \land P(y, z) \rightarrow P(x, z))] \land \text{transitivity} \\
[\forall x \forall y (P(x, y) \land P(y, x) \rightarrow x = y)] \land \text{anti-symmetry} \\
[\forall x \forall y (P(x, y) \lor P(y, x))] \land \text{totality}
\]

\(\psi_2(P, x, y)\) 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 \(P\):

\[
\psi_2(P, x, y) \triangleq \neg(x = y) \land P(x, y) \land \forall z [P(x, z) \land P(z, y) \rightarrow (x = z \lor y = z)]
\]

**Exercise:** When can we omit \(\neg(x = y)\) in \(\psi_2(P, x, y)\)? Explain.
examples about graphs \((A, R)\) where \(A\) is the set of nodes and \(R\) is a binary relation representing edges
examples about 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 about 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 \(\neg P\)

\[
\varphi \triangleq \exists P \forall x \forall y [ \neg (x = y) \land R(x, y) \rightarrow (P(x) \leftrightarrow \neg P(y)) ]
\]

**Exercise:** When can we omit “\(\neg (x = y)\)” in \(\varphi\)? Explain.
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

\[\psi_1(A_1, A_2, A_3) \equiv \forall x [ (A_1(x) \land \neg A_2(x) \land \neg A_3(x)) \lor (\neg A_1(x) \land A_2(x) \land \neg A_3(x)) \lor (\neg A_1(x) \land \neg A_2(x) \land A_3(x))]\]

\[\psi_2(A_1, A_2, A_3) \equiv \forall x \forall y [ (A_1(x) \land A_1(y) \rightarrow \neg R(x,y)) \land (A_2(x) \land A_2(y) \rightarrow \neg R(x,y)) \land (A_3(x) \land A_3(y) \rightarrow \neg R(x,y))]\]

\[\phi \equiv \exists A_1 \exists A_2 \exists A_3 (\psi_1 \land \psi_2)\]
examples about 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 \(A_1, A_2, \text{ and } A_3\)
examples about 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 \(A_3\)

- \(\psi_1\) says “each node has exactly one color”:

  \[
  \psi_1(A_1, A_2, A_3) \triangleq \forall x \left[ (A_1(x) \land \neg A_2(x) \land \neg A_3(x)) \lor \\
  (\neg A_1(x) \land A_2(x) \land \neg A_3(x)) \lor \\
  (\neg A_1(x) \land \neg A_2(x) \land A_3(x)) \right]
  \]
examples about 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 \(A_3\)

- \(\psi_1\) says “each node has exactly one color”:

  \[
  \psi_1(A_1, A_2, A_3) \overset{\Delta}{=} \forall x \left[ (A_1(x) \land \neg A_2(x) \land \neg A_3(x)) \lor \\
  (\neg A_1(x) \land A_2(x) \land \neg A_3(x)) \lor \\
  (\neg A_1(x) \land \neg A_2(x) \land A_3(x)) \right]
  \]

- \(\psi_2\) says “no two points with the same color are connected”:

  \[
  \psi_2(A_1, A_2, A_3) \overset{\Delta}{=} \forall x \forall y \left[ (A_1(x) \land A_1(y) \rightarrow \neg R(x, y)) \land \\
  (A_2(x) \land A_2(y) \rightarrow \neg R(x, y)) \land \\
  (A_3(x) \land A_3(y) \rightarrow \neg R(x, y)) \right]
  \]
examples about 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 \(A_3\)

- \(\psi_1\) says “each node has exactly one color”:

  \[
  \psi_1(A_1, A_2, A_3) \triangleq \forall x \left[ (A_1(x) \land \neg A_2(x) \land \neg A_3(x)) \lor 
  (\neg A_1(x) \land A_2(x) \land \neg A_3(x)) \lor 
  (\neg A_1(x) \land \neg A_2(x) \land A_3(x)) \right]
  \]

- \(\psi_2\) says “no two points with the same color are connected”:

  \[
  \psi_2(A_1, A_2, A_3) \triangleq \forall x \forall y \left[ (A_1(x) \land A_1(y) \rightarrow \neg R(x, y)) \land 
  (A_2(x) \land A_2(y) \rightarrow \neg R(x, y)) \land 
  (A_3(x) \land A_3(y) \rightarrow \neg R(x, y)) \right]
  \]

- \(\varphi \triangleq \exists A_1 \exists A_2 \exists A_3 \left( \psi_1 \land \psi_2 \right)\)
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- **unconnectedness**
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- **unconnectedness**
- \(\psi_1\) says “the set \(A\) is non-empty and its complement is nonempty”

\[
\psi_1(A) \triangleq \exists x \exists y \ [ A(x) \land \neg A(y) ]
\]
examples about graphs \((A, R)\) where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- **unconnectedness**
- \(\psi_1\) says “the set \(A\) is non-empty and its complement is nonempty”
  \[
  \psi_1(A) \triangleq \exists x \exists y \ [ A(x) \land \neg A(y) ]
  \]
- \(\psi_2\) says “there is no edge between \(A\) and its complement”
  \[
  \psi_2(A) \triangleq \forall x \forall y \ [ (A(x) \land \neg A(y)) \rightarrow (\neg R(x, y) \land \neg R(y, x)) ]
  \]
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- **unconnectedness**

- \(\psi_1\) says “the set \(A\) is non-empty and its complement is nonempty”

\[
\psi_1(A) \triangleq \exists x \exists y [ A(x) \land \neg A(y) ]
\]

- \(\psi_2\) says “there is no edge between \(A\) and its complement”

\[
\psi_2(A) \triangleq \forall x \forall y [ (A(x) \land \neg A(y)) \rightarrow (\neg R(x, y) \land \neg R(y, x)) ]
\]

- \(\varphi \triangleq \exists A (\psi_1 \land \psi_2)\)
  is true iff graph is not connected
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- unconnectedness
- \(\psi_1\) says “the set \(A\) is non-empty and its complement is nonempty”
  \[
  \psi_1(A) \triangleq \exists x \exists y \left[ A(x) \land \neg A(y) \right]
  \]
- \(\psi_2\) says “there is no edge between \(A\) and its complement”
  \[
  \psi_2(A) \triangleq \forall x \forall y \left[ (A(x) \land \neg A(y)) \rightarrow (\neg R(x, y) \land \neg R(y, x)) \right]
  \]
- \(\varphi \triangleq \exists A \left( \psi_1 \land \psi_2 \right)\)
  is true iff graph is not connected
- \(\varphi' \triangleq \neg \varphi \triangleq \forall A \left( \neg \psi_1 \lor \neg \psi_2 \right) \triangleq \forall A \left( \psi_1 \rightarrow \neg \psi_2 \right)\)
  is true iff graph is connected
connections with descriptive complexity theory

- The WFF $\varphi$ in each of page 5, page 11, page 14, and page 19 is an **existential second-order WFF**.

- Moreover, the $\varphi$ in each of page 11, page 14, and page 19, but not in page 5, is a **monadic second-order WFF**, because the second-order variables in $\varphi$ are restricted to be unary-predicate (i.e., set) variables.

- Fagin’s theorem: existential second-order logic coincides with the complexity class NP in the sense that a decision problem can be expressed in existential second-order logic if and only if it can be solved by a nondeterministic Turing machine in polynomial time.

- **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.”)