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.
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
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”
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”

  \[
  \varphi \triangleq \exists P \left[ \text{“} P \text{ is a linear order” } \land \forall x \forall y \left( \text{“} y = x + 1 \text{” } \Rightarrow \text{ } R(x, y) \right) \right]
  \]
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

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

\[
\varphi \triangleq \exists P \left[ \psi_1(P) \land \forall x \forall y \left( \psi_2(P, x, y) \rightarrow R(x, y) \right) \right]
\]
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”

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

\[
\varphi \triangleq \exists P \left[ \psi_1(P) \wedge \forall x \forall y \left( \psi_2(P, x, y) \rightarrow R(x, y) \right) \right]
\]

\(\psi_1(P)\) is a WFF with free predicate-variable \(P\) of arity 2, which makes \(P\) a linear order:

\[
\psi_1(P) \triangleq [\forall x P(x, x)] \wedge \text{reflexivity}
\]

\[
[\forall x \forall y \forall z \left( P(x, y) \wedge P(y, z) \rightarrow P(x, z) \right)] \wedge \text{transitivity}
\]

\[
[\forall x \forall y \left( P(x, y) \wedge P(y, x) \rightarrow x = y \right)] \wedge \text{anti-symmetry}
\]

\[
[\forall x \forall y \left( P(x, y) \vee P(y, x) \right)] \wedge \text{ totality}
\]
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”

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

\(\psi_1(P)\) is a WFF with free predicate-variable \(P\) of arity 2, which makes \(P\) 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 \(P\):

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

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

Assaf Kfoury, CS 512, Spring 2016, Handout 16
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- **3-colorability:**
  represent the 3 colors by unary predicates \(A_1, A_2,\) 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 the 3 colors by unary predicates \(A_1, A_2,\) and \(A_3\)
- \(\psi_1\) says “each node has exactly one color”:

\[
\psi_1 \triangleq \forall x \left[ (A_1(x) \land \neg A_2(x) \land \neg A_3(x)) \lor \\
(\neg A_1(x) \land A_2(x) \land \neg A_3(x)) \lor \\
(\neg A_1(x) \land \neg A_2(x) \land A_3(x)) \right]
\]
examples about graphs \((A, R)\)
where \(A\) is the set of nodes and \(R\) is a binary relation representing edges

- **3-colorability:**
  represent the 3 colors by unary predicates \(A_1, A_2, \text{ and } A_3\)
- \(\psi_1\) says “each node has exactly one color”:
  
  \[
  \psi_1 \triangleq \forall x \left[ \left( A_1(x) \land \neg A_2(x) \land \neg A_3(x) \right) \lor \left( \neg A_1(x) \land A_2(x) \land \neg A_3(x) \right) \lor \left( \neg A_1(x) \land \neg A_2(x) \land A_3(x) \right) \right]
  \]

- \(\psi_2\) says “no two points with the same color are connected”:
  
  \[
  \psi_2 \triangleq \forall x \forall y \left[ \left( A_1(x) \land A_1(y) \rightarrow \neg R(x, y) \right) \land \left( A_2(x) \land A_2(y) \rightarrow \neg R(x, y) \right) \land \left( A_3(x) \land A_3(y) \rightarrow \neg R(x, y) \right) \right]
  \]
examples about graphs \((A, R)\)

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

- **3-colorability:**
  represent the 3 colors by unary predicates \(A_1, A_2,\) and \(A_3\)

  - \(\psi_1\) says “each node has exactly one color”:
    \[
    \psi_1 \triangleq \forall x \left[ \left( A_1(x) \land \neg A_2(x) \land \neg A_3(x) \right) \lor \left( \neg A_1(x) \land A_2(x) \land \neg A_3(x) \right) \lor \left( \neg A_1(x) \land \neg A_2(x) \land A_3(x) \right) \right]
    \]

  - \(\psi_2\) says “no two points with the same color are connected”:
    \[
    \psi_2 \triangleq \forall x \forall y \left[ \left( A_1(x) \land A_1(y) \rightarrow \neg R(x, y) \right) \land \left( A_2(x) \land A_2(y) \rightarrow \neg R(x, y) \right) \land \left( A_3(x) \land A_3(y) \rightarrow \neg R(x, y) \right) \right]
    \]

  - \(\varphi\) is defined as:
    \[
    \varphi \triangleq \exists A_1 \exists A_2 \exists A_3 \left( \psi_1 \land \psi_2 \right)
    \]
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”

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

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

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

  \[
  \psi_2 \triangleq \forall x \forall y \left[ (A(x) \land \neg A(y)) \rightarrow (\neg R(x, y) \land \neg R(y, x)) \right]
  \]
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”
  \[
  \psi_1 \equiv \exists x \exists y \left[ A(x) \land \neg A(y) \right]
  \]
- \(\psi_2\) says “there is no edge between \(A\) and its complement”
  \[
  \psi_2 \equiv \forall x \forall y \left[ (A(x) \land \neg A(y)) \rightarrow (\neg R(x, y) \land \neg R(y, x)) \right]
  \]
- \(\varphi \equiv \exists A (\psi_1 \land \psi_2)\)
  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”

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

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

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

- \(\varphi \triangleq \exists A \left( \psi_1 \land \psi_2 \right)\)
  
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.”)