Program Verification: Lecture 14

José Meseguer

Computer Science Department
University of Illinois at Urbana-Champaign

Inductive Theorems do not Change the Initial Algebra

Theorem (Lemma Internalization Theorem 1) Let (Σ, E) be an equational theory and G a set of Σ-equations such that $(\Sigma, E) \models_{ind} G$. Then, $\mathbb{T}_{\Sigma/E} = \mathbb{T}_{\Sigma/E \cup G}$.

Proof: Since $\mathbb{T}_{\Sigma/E \cup G} \models E$ we have a unique Σ-homomorphism $h : \mathbb{T}_{\Sigma/E} \to \mathbb{T}_{\Sigma/E \cup G}$. And since $\mathbb{T}_{\Sigma/E} \models E \cup G$, we also have a unique Σ-homomorphism $g : \mathbb{T}_{\Sigma/E \cup G} \to \mathbb{T}_{\Sigma/E}$. But then, the initiality of $\mathbb{T}_{\Sigma/E}$ forces $h; g = id_{\mathbb{T}_{\Sigma/E}}$, and the initiality of $\mathbb{T}_{\Sigma/E \cup G}$ forces $g; h = id_{\mathbb{T}_{\Sigma/E \cup G}}$. Therefore, we have an isomorphism: $\mathbb{T}_{\Sigma/E} \cong \mathbb{T}_{\Sigma/E \cup G}$. We will be done of we prove the following lemma:

Lemma Let E, E' be two sets of Σ-equations such that $\mathbb{T}_{\Sigma/E} \cong \mathbb{T}_{\Sigma/E'}$. Then, $\mathbb{T}_{\Sigma/E} = \mathbb{T}_{\Sigma/E'}$.

Inductive Theorems do not Change the Initial Algebra (II)

Proof of the Lemma: $\mathbb{T}_{\Sigma/E}$ and $\mathbb{T}_{\Sigma/E'}$ are uniquely determined by the respective ground equality relations $=_E \cap T_\Sigma^2$ and $=_{E'} \cap T_\Sigma^2$. We just need to show $(=_E \cap T_\Sigma^2) = (=_{E'} \cap T_\Sigma^2)$. Since we have a Σ-isomorphism $h : \mathbb{T}_{\Sigma/E} \to \mathbb{T}_{\Sigma/E'}$, and unique Σ-homomorphisms $[\_]_E : \mathbb{T}_\Sigma \to \mathbb{T}_{\Sigma/E}$, and $[\_]_{E'} : \mathbb{T}_\Sigma \to \mathbb{T}_{\Sigma/E}$, the initiality of $\mathbb{T}_\Sigma$ forces $[\_]_E; h = [\_]_{E'}$, i.e., $h_s([t]_E) = [t]_{E'}$ for each $t \in T_{\Sigma,s}$, $s \in S$. Let $t \in T_{\Sigma,s}$ and $t' \in T_{\Sigma,s'}$ with $t =_E t'$. Then $[s] = [s']$ and, by h order-sorted Σ-homomorphism and $[t]_E = [t']_E$, we must have $h_s([t]_E) = h_{s'}([t']_E)$, which forces:

$$h_s([t]_E) = [t]_{E'} = [t']_{E'} = h_{s'}([t']_E)$$

giving us the containment $(=_E \cap T_\Sigma^2) \subseteq (=_{E'} \cap T_\Sigma^2)$. Using the inverse isomorphism h^{-1} we likewise get $(=_{E'} \cap T_\Sigma^2) \subseteq (=_E \cap T_\Sigma^2)$, giving us $(=_E \cap T_\Sigma^2) = (=_{E'} \cap T_\Sigma^2)$, as desired. q.e.d. q.e.d.

Equivalence of Equational Theories

Call two equational theories (Σ, E) and (Σ, E') equivalent, denoted $(\Sigma, E) \equiv (\Sigma, E')$ iff (by definition) $E \vdash E'$ and $E' \vdash E$.

Ex.14.1 Prove that:

$$(\Sigma, E) \equiv (\Sigma, E') \;\Leftrightarrow\; (=_E) = (=_{E'}) \;\Leftrightarrow\; \mathbf{Alg}_{(\Sigma,E)} = \mathbf{Alg}_{(\Sigma,E')}.$$

For example, the sets of equations $E = \{x \cdot (y \cdot z) = (x \cdot y) \cdot z, x \cdot 1 = x = 1 \cdot x, x \cdot x^{-1} = 1, 1 = x^{-1} \cdot x\}$, and $E' = \{(x \cdot y) \cdot z = x \cdot (y \cdot z), 1 \cdot x = x, x \cdot 1 = x, x \cdot x^{-1} = 1, x^{-1} \cdot x = 1, 1^{-1} = 1, (x^{-1})^{-1} = x, (x \cdot y)^{-1} = y^{-1} \cdot x^{-1}, x \cdot (x^{-1} \cdot y) = y, x^{-1} \cdot (x \cdot y) = y\}$ define equivalent theories $(\Sigma, E) \equiv (\Sigma, E')$ for the theory of groups. But E' is much better, because $\vec{E'}$ is confluent and terminating. Therefore, by the Church-Rosser Theorem we can decide whether any Σ-equality $u = v$ is a theorem of group theory by checking whether $u!_{\vec{E'}} = v!_{\vec{E'}}$.

Inductive Equivalence of Equational Theories

Call two equational theories (Σ, E) and (Σ, E') inductively equivalent, denoted $(\Sigma, E) \equiv_{ind} (\Sigma, E')$ iff (by definition) $(\Sigma, E) \models_{ind} E'$ and $(\Sigma, E') \models_{ind} E$.

Ex.14.2 Prove that:

$$(\Sigma, E) \equiv_{ind} (\Sigma, E') \;\Leftrightarrow\; (=_E \cap T_\Sigma^2) = (=_{E'} \cap T_\Sigma^2) \;\Leftrightarrow\; \mathbb{T}_{\Sigma/E} = \mathbb{T}_{\Sigma/E'}.$$

Ex.14.1 and **Ex**.14.2 give us $(\Sigma, E) \equiv (\Sigma, E') \Rightarrow (\Sigma, E) \equiv_{ind} (\Sigma, E')$. But in general $(\Sigma, E) \equiv_{ind} (\Sigma, E')$ does not imply $(\Sigma, E) \equiv (\Sigma, E')$.

For example, as explained in Lecture 13, For $\Sigma = \{0, s, \_+\_\}$ and $E = \{x + 0 = x, x + s(y) = s(x + y)\}$, $\mathbb{T}_{\Sigma/E} \models x + y = y + x$. Thus, by the Lemma Internalization Theorem 1 and **Ex**.14.2 we have $(\Sigma, E) \equiv_{ind} (\Sigma, E \cup \{x + y = y + x\})$. But we saw in Lecture 13 that $E \not\vdash x + y = y + x$, and therefore $(\Sigma, E) \not\equiv (\Sigma, E \cup \{x + y = y + x\})$.

Semantic Equivalence of Equational Programs

In Program Verification a fundamental question is:

When are two different programs semantically equivalent?

The most obvious answer for admissible equational programs `fmod` (Σ, E) `endfm` and `fmod` (Σ, E') `endfm` is:

When they compute the same recursive functions,

which mathematically just means: when $\mathbb{C}_{\Sigma/\vec{E}} = \mathbb{C}_{\Sigma/\vec{E}'}$.

For example, we shall prove that for $\Sigma = \{0, s, \_+\_\}$,
$E = \{x + 0 = x, x + s(y) = s(x + y)\}$ and
$E' = \{0 + x = x, s(x) + y = s(x + y)\}$, `fmod` (Σ, E) `endfm` and `fmod` (Σ, E') `endfm` are equivalent equational programs: both compute the standard addition function on natural numbers $+_{\mathbb{N}}$.

Let us give a more precise (and more general) definition.

Admissible and Comparable programs

Call `fmod` $(\Sigma, E \cup B)$ `endfm` admissible iff (i) Σ is B-preregular, with non-empty sorts, (ii) $\vec{E}$ is sort-decreasing, and ground confluent and terminating modulo B, and (iii) it is sufficiently complete w.r.t. a constructor subsignature Ω.

Call $(\Sigma, E \cup B)$ satisfying (i)–(ii) ground convergent modulo B.

Given a constructor subsignature $\Omega \subseteq \Sigma$, Ω^+ denotes the signature that extends Ω by adding all non-constructor operator typings that are subsort-overloaded with some operator in Ω. Call two admissible equational programs `fmod` $(\Sigma, E \cup B)$ `endfm` and `fmod` $(\Sigma, E' \cup B')$ `endfm` comparable iff: (i) $E = E_0 \uplus E_{\Omega^+}$ and $E' = E'_0 \uplus E'_{\Omega^+}$, with $E_{\Omega^+} \cup E'_{\Omega^+}$ Ω-equations, and each rule in $\vec{E}_0 \cup \vec{E}'_0$ of the form $f(u_1, \ldots, u_n) \to v$, with f in $\Sigma \setminus \Omega^+$, and (ii) $B = B_0 \uplus B_{\Omega^+}$ and $B' = B'_0 \uplus B_{\Omega^+}$, with B_{Ω^+} $A \vee C \vee U$ Ω^+-axioms, and $B_0 \cup B'_0$ $A \vee C$ $(\Sigma \setminus \Omega^+)$-axioms.

Semantic Equivalence of Equational Programs (II)

Admissible and comprable programs `fmod` $(\Sigma, E \cup B)$ `endfm` and `fmod` $(\Sigma, E' \cup B')$ `endfm` are called semantically equivalent, denoted `fmod` $(\Sigma, E \cup B)$ `endfm` $\equiv_{sem}$ `fmod` $(\Sigma, E' \cup B')$ `endfm` iff $\mathbb{C}_{\Sigma/\vec{E},B} = \mathbb{C}_{\Sigma/\vec{E}',B'}$.

Since the axioms in $B_0 \cup B_0'$ are $A \vee C$ $(\Sigma \setminus \Omega^+)$-axioms, for any $u, v \in T_{\Omega^+}$, $u =_B v$ (resp. $u =_{B'} v$) forces $u =_{B_{\Omega^+}} v$. Therefore, the unique Σ-homomorphisms $[\_!_{\vec{E}/B}]_B : \mathbb{T}_\Sigma \to \mathbb{C}_{\Sigma/\vec{E},B}$ and $[\_!_{\vec{E}'/B'}]_{B'} : \mathbb{T}_\Sigma \to \mathbb{C}_{\Sigma/\vec{E}',B'}$ can more precisely be described as $[\_!_{\vec{E}/B}]_{B_{\Omega^+}} : \mathbb{T}_\Sigma \to \mathbb{C}_{\Sigma/\vec{E},B}$ and $[\_!_{\vec{E}'/B'}]_{B_{\Omega^+}} : \mathbb{T}_\Sigma \to \mathbb{C}_{\Sigma/\vec{E}',B'}$.

Ex.14.3. Prove that for admissible and comparable `fmod` $(\Sigma, E \cup B)$ `endfm` and `fmod` $(\Sigma, E' \cup B')$ `endfm`, `fmod` $(\Sigma, E \cup B)$ `endfm` $\equiv_{sem}$ `fmod` $(\Sigma, E' \cup B')$ `endfm` iff $\forall t \in T_\Sigma$, $t!_{\vec{E}/B} =_{B_{\Omega^+}} t!_{\vec{E}'/B'}$. I.e., if Maude's `red` command gives the same result for both modulo B_{Ω^+}.

Semantic Equivalence of Equational Programs (III)

Note that $\mathbb{C}_{\Sigma/\vec{E},B} = \mathbb{C}_{\Sigma/\vec{E}',B'}$ and the Lemma in pg. 2 force $\mathbb{T}_{\Sigma/E\cup B} = \mathbb{T}_{\Sigma/E'\cup B'}$. Therefore, by **Ex**.14.2, `fmod` $(\Sigma, E \cup B)$ `endfm` $\equiv_{sem}$ `fmod` $(\Sigma, E' \cup B')$ `endfm` implies $(\Sigma, E \cup B) \equiv_{ind} (\Sigma, E' \cup B')$. But the converse implication does not hold in general.

For example, for $\Sigma = \{a, b, c\}$, $E = \{a = b\}$, and $E' = \{b = a\}$, of course $(\Sigma, E) \equiv (\Sigma, E')$ and therefore $(\Sigma, E) \equiv_{ind} (\Sigma, E')$; but although $\vec{E}$ and $\vec{E}'$ are both convergent, they have different constructors $\Omega = \{b, c\}$ and $\Omega' = \{a, c\}$, so that $\mathbb{C}_{\Sigma/\vec{E}} \neq \mathbb{C}_{\Sigma/\vec{E}'}$. Therefore, `fmod` $(\Sigma, E \cup B)$ `endfm` $\not\equiv_{sem}$ `fmod` $(\Sigma, E' \cup B')$ `endfm`.

Theorem (Program Equivalence Theorem) For admissible and comparable `fmod` $(\Sigma, E \cup B)$ `endfm` and `fmod` $(\Sigma, E' \cup B')$ `endfm`, `fmod` $(\Sigma, E \cup B)$ `endfm` $\equiv_{sem}$ `fmod` $(\Sigma, E' \cup B')$ `endfm` iff $(\Sigma, E \cup B) \equiv_{ind} (\Sigma, E' \cup B')$.

Semantic Equivalence of Equational Programs (IV)

Proof: The ($\Rightarrow$) implication has already been shown. To prove the ($\Leftarrow$) implication, by **Ex**.14.3. we just need to show that $\forall t \in T_\Sigma$, $t!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E'}/B'}$. But we have ground proofs $t!_{\vec{E}/B} =_{E \cup B} t =_{E' \cup B'} t!_{\vec{E'}/B'}$, and by **Ex**.14.2, a ground proof $t!_{\vec{E}/B} =_{E \cup B} t!_{\vec{E'}/B'}$, which, by $(\Sigma, E \cup B)$ ground convergent modulo B, the ground Church-Rosser Theorem modulo B, and sufficient completeness, forces $t!_{\vec{E}/B} =_{B_{\Omega+}} (t!_{\vec{E'}/B'})!_{\vec{E}/B}$. Note that, by program comparability, both terms are $\vec{E}_{\Omega+}/B$-irreducible. Furthermore, by B_0 $A \vee C$ $(\Sigma \setminus \Omega^+)$-axioms and lefthand sides of rule in $\vec{E}_0$ not Ω^+-terms, if $u \in T_\Omega$, any proof $u =_B v$ must be a proof $u =_{B_{\Omega+}} v$, and therefore with $v \in T_\Omega$. Since the lefthand sides of rules in $\vec{E}_0$ are not Ω^+-terms, this means that $t!_{\vec{E'}/B'}$ is also $\vec{E}_0/B$-irreducible, and therefore $(t!_{\vec{E'}/B'})!_{\vec{E}/B} = t!_{\vec{E'}/B'}$, giving us $t!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E'}/B'}$, as desired. q.e.d

Internalizing Lemmas in Equational Programs

Theorem (Lemma Internalization Theorem 2) Let `fmod` $(\Sigma, E \cup B)$ **`endfm`** be an admissible program with constructors Ω satisfying the extra requirements on E and B allowing it to be comparable to other programs, and let G be a finite set of Σ-equations such that $(\Sigma, E \cup B) \models_{ind} G$. If the equations G can be oriented (left-to right or right to left) as sort-decreasing rules $\vec{G}$ of the form $f(u_1, \ldots, u_n) \to w$ with f in $\Sigma \setminus \Omega^+$ and so that $\vec{E} \cup \vec{G}$ are terminating modulo B, then `fmod` $(\Sigma, E \cup G' \cup B)$ **`endfm`** (with $\vec{G'} = \vec{G}$) is admissible and $(\Sigma, E \cup B)$ **`endfm`** $\equiv_{sem}$ `fmod` $(\Sigma, E \cup G' \cup B)$ **`endfm`**.

Proof: We first prove that $(\Sigma, E \cup G' \cup B)$ is ground convergent modulo B. Then, `fmod` $(\Sigma, E \cup G' \cup B)$ **`endfm`** will also be admissible and comparable to `fmod` $(\Sigma, E \cup B)$ **`endfm`**. To prove the Theorem, using **Ex**.14.3, we then need to also show that

$t!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E}\cup\vec{G}/B}$ for any $t \in T_\Sigma$.

To prove $(\Sigma, E \cup G' \cup B)$ ground convergent modulo B we only need consider the joinability of all pairs: (i) $u \;{}_{\vec{E}/B}\!\leftarrow t \rightarrow_{\vec{G}/B} v$ and (ii) $u \;{}_{\vec{G}/B}\!\leftarrow t \rightarrow_{\vec{G}/B} v$ with $t \in T_\Sigma$. Since $\rightarrow_{\vec{E}/B} \subseteq \rightarrow_{\vec{E}\cup\vec{G}/B}$, it is enough to show joinability with $\rightarrow_{\vec{E}/B}$. Let us show joinability for case (i); case (ii) is left as an exercise. By the Theorem's hypothesis, the Lemma Internalization 1 Theorem, and **Ex**.14.2, we have $(=_{E\cup B} \cap T_\Sigma^2) = (=_{E\cup G\cup B} \cap T_\Sigma^2)$. Since $u \;{}_{\vec{G}/B}\!\leftarrow t$ is a ground proof $u =_{G\cup B} t$, we then also have a ground proof $u =_{E\cup B} t$, and by $(\Sigma, E \cup B)$ ground convergent modulo B, the ground Church-Rosser Theorem modulo B and sufficient completeness we must have $u!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E}/B}$, showing the pair joinable.

To prove $t!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E}\cup\vec{G}/B}$ for any $t \in T_\Sigma$, note that, using $(=_{E\cup B} \cap T_\Sigma^2) = (=_{E\cup G\cup B} \cap T_\Sigma^2)$ again, we have a ground proof of the form $t!_{\vec{E}/B} =_{E\cup B} t!_{\vec{E}\cup\vec{G}/B}$, which by $(\Sigma, E \cup B)$ ground convergent modulo B, the ground Church-Rosser Theorem modulo B, and sufficient completeness forces $t!_{\vec{E}/B} =_{B_{\Omega+}} (t!_{\vec{E}\cup\vec{G}/B})!_{\vec{E}/B}$. But since $t!_{\vec{E}\cup\vec{G}/B}$ is obviously $\vec{E}/B$-irreducible, we get $(t!_{\vec{E}\cup\vec{G}/B})!_{\vec{E}/B} = t!_{\vec{E}\cup\vec{G}/B}$, and therefore $t!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E}\cup\vec{G}/B}$, as desired. q.e.d.

Internalizing Lemmas in Equational Programs (II)

Theorem (Lemma Internalization Theorem 3) Let `fmod` $(\Sigma, E \cup B)$ `endfm` be an admissible program with constructors Ω satisfying the extra requirements on E and B to be comparable to other programs, and let G be a finite set of $A \vee C$ $\Sigma \setminus \Omega^+$-axioms general enough to declare all subsort-overloaded versions of some binary operators in $\Sigma \setminus \Omega^+$ $A \vee C$ and making Σ $(B \cup G)$-preregular, and such that $(\Sigma, E \cup B) \models_{ind} G$. Then, if the rules $\vec{E}$ can be proved terminating modulo $B \cup G$, `fmod` $(\Sigma, E \cup B \cup G)$ `endfm` is admissible and $(\Sigma, E \cup B)$ `endfm` $\equiv_{sem}$ `fmod` $(\Sigma, E \cup B \cup G)$ `endfm`.

Proof: We first need to show $(\Sigma, E \cup B \cup G)$ ground convergent modulo $B \cup G$, i.e., the joinability of all pairs $u \ {}_{\vec{E}/B \cup G}\!\leftarrow t \rightarrow_{\vec{E}/B \cup G} v$ with $t \in T_{\Sigma}$. Since $\rightarrow_{\vec{E}/B} \subseteq \rightarrow_{\vec{E}/B \cup G}$, it is enough to show joinability with $\rightarrow_{\vec{E}/B}$. But by the Theorem's hypothesis, the Lemma Internalization 1 Theorem, and **Ex**.14.2, we have $(=_{E \cup B} \cap T_{\Sigma}^2) = (=_{E \cup G \cup B} \cap T_{\Sigma}^2)$. Furthermore, the pair $u \ {}_{\vec{E}/B \cup G}\!\leftarrow t \rightarrow_{\vec{E}/B \cup G} v$ gives us a ground proof $u =_{E \cup B \cup G} t =_{E \cup B \cup G} v$, and therefore a ground proof $u =_{E \cup B} t =_{E \cup B} v$. But by $(\Sigma, E \cup B)$ ground convergent modulo B, the ground Church-Rosser Theorem modulo B and sufficient completness we must have $u!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E}/B} =_{B_{\Omega+}} v!_{\vec{E}/B}$, showing the pair joinable.

We will be done if we show that $t!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E}/B \cup G}$. But, using $(=_{E \cup B} \cap T_{\Sigma}^2) = (=_{E \cup G \cup B} \cap T_{\Sigma}^2)$ again, we have a ground proof $t!_{\vec{E}/B} =_{E \cup B} t!_{\vec{E}/B \cup G}$, which by $(\Sigma, E \cup B)$ ground convergent modulo B, the ground Church-Rosser Theorem modulo B and sufficient completeness forces $t!_{\vec{E}/B} =_{B_{\Omega+}} (t!_{\vec{E}/B \cup})!_{\vec{E}/B}$. But since $t!_{\vec{E}/B \cup G}$ is obviously $\vec{E}/B$-irreducible, we get $(t!_{\vec{E}/B \cup G})!_{\vec{E}/B} = t!_{\vec{E}/B \cup G}$, and therefore $t!_{\vec{E}/B} =_{B_{\Omega+}} t!_{\vec{E}/B \cup G}$, as desired. q.e.d.

Formal Verification of Equational Programs

We shall consider two main problems in the formal verification of equational programs:

1. Proofs of Program Equivalence, that is, of equivalences of the form: $\texttt{fmod}\ (\Sigma, E \cup B)\ \texttt{endfm} \equiv_{sem} \texttt{fmod}\ (\Sigma, E' \cup B')\ \texttt{endfm}$ for admissible and comparable programs.

2. Proofs of Program Properties, which in their most general form, for an admissible program $\texttt{fmod}\ (\Sigma, E \cup B)\ \texttt{endfm}$, just means proofs of properties of the form $\mathbb{C}_{\Sigma/\vec{E},B} \models \varphi$ or, equivalently, $\mathbb{T}_{\Sigma/E \cup B} \models \varphi$, for φ a first-order logic (FOL) Σ-formula.

Formal Verification of Equational Programs (II)

Regarding proofs of program equivalence, we have three theorems, namely, the Program Equivalence Theorem, and the Lemma Internalization Theorems 2 and 3, which in essence reduce all such proofs to proofs of inductive consequences of the form $(\Sigma, E \cup B) \models_{ind} G$, for G a finite set of equations.

Regarding proofs of program properties, since equational logic is a sublogic of first-order logic, we can just generalize the $\models_{ind}$ relation to first-order logic Σ-formulas φ by stating that $(\Sigma, E \cup B) \models_{ind} \varphi$ holds by definition iff $\mathbb{T}_{\Sigma/E \cup B} \models \varphi$.

This requires explaining the syntax and semantics of first-order logic, including the satisfaction relation $\mathbb{A} \models \varphi$ between a Σ-algebra $\mathbb{A}$ and a first-order logic Σ-formula φ. The Appendix to this lecture explains these topics in sufficient detail for our present purposes.

The Need for an Inductive Logic

Therefore, the task of equational program verification, both in proving program equivalences and program properties, boils down to proving inductive consequences of the form $(\Sigma, E \cup B) \models_{ind} \varphi$ (in the case of a set of equations $G = \{u_1 = v_1, \ldots, u_n = v_n\}$, $\varphi = (u_1 = v_1 \wedge \ldots \wedge u_n = v_n)$). But, by definition, proving $(\Sigma, E \cup B) \models_{ind} \varphi$ exactly means proving that $\mathbb{T}_{\Sigma/E \cup B} \models \varphi$, which is a semantic relation between the initial algebra $\mathbb{T}_{\Sigma/E \cup B}$ and a FOL formula φ.

For this, we need correct reasoning principles unambiguously embodied in a formal system of inference rules which we can rightly call an inductive logic, denoted $\vdash_{ind}$, allowing us to prove the semantic property $(\Sigma, E \cup B) \models_{ind} \varphi$ by proving $(\Sigma, E \cup B) \vdash_{ind} \varphi$.

The Need for an Inductive Logic (II)

Of course, saying that the inductive logic $\vdash_{ind}$ provides "correct reasoning principles" for this task exactly means that $\vdash_{ind}$ is sound. That is, that for any $(\Sigma, E \cup B)$ and φ we have an implication:

$$(\Sigma, E \cup B) \vdash_{ind} \varphi \quad \Rightarrow \quad (\Sigma, E \cup B) \models_{ind} \varphi$$

Can $\vdash_{ind}$ be complete, so that the reverse implication holds?

The answer is no. To explain why not, we need to observe that the set $PThm_{\vdash_{ind}}(\Sigma, E \cup B)$ of theorems of a theory $(\Sigma, E \cup B)$ provable by an inference system $\vdash_{ind}$ defined by inference rules that syntactically manipulate formulas (where the theory's "axioms" $E \cup B$ are a finite or recursively enumerable set) must be a recursively enumerable set (r.e. set). This is so because we can implement $\vdash_{ind}$ by a computer program that generates the set $PThm_{\vdash_{ind}}(\Sigma, E \cup B)$, so that $PThm_{\vdash_{ind}}(\Sigma, E \cup B)$ must be r.e.

Göedel for Dummies

Let (Σ, E) be the equational theory of the Maude program:

```
fmod NAT+x is sort Nat .
op 0 : -> Nat [ctor] .  op s: Nat -> Nat [ctor] .
ops (_+_) (_*_) : Nat Nat -> Nat .   vars N  M : Nat .
eq N + 0 = N .                eq N * 0
eq N + s(M) = s(N + M) .    eq N * s(M) = N + (N * M) .
```

Theorem (Göedel's Incompleteness of Arithmetic). For the above theory (Σ, E), the set

$$Thm_{\models_{ind}}(\Sigma, E) = \{\varphi \in Form_{FOL}(\Sigma) \mid \mathbb{T}_{\Sigma/E} \models_{ind} \varphi\} = Thm_{FOL}(\mathbb{T}_{\Sigma/E})$$

is not r.e.

Therefore for any sound inductive logic $\vdash_{ind}$ in general we will have a strict containment $PThm_{\vdash_{ind}}(\Sigma, E \cup B) \subset Thm_{\models_{ind}}(\Sigma, E \cup B)$, making $\vdash_{ind}$ necessarily incomplete.

The Inference System $\vdash_{ind}$ of Maude's NuITP

To prove both equational program equivalences and equational program properties we shall use Maude's New Inductive Theorem Prover (NuITP), which mechanizes the inference rules of a sound inductive logic $\vdash_{ind}$.

The formulas that $\vdash_{ind}$, and therefore Maude's NuITP, proves are quantifier-free multiclauses, which, as the Appendix to this lecture on FOL explains, are formulas of the form:

$$(w_1 = w'_1 \wedge \ldots \wedge w_k = w'_k) \Rightarrow ((u^1_1 = v^1_1 \vee \ldots \vee u^1_{m_1} = v^1_{m_1}) \wedge \ldots \wedge (u^k_1 = v^k_1 \vee \ldots \vee u^k_{m_k} = v^k_{m_k}).)$$