CS4860 Lecture 19 Löwenheim Skolem Theorem, the Skolem "Paradox"

Thur Nov 1, 2012

Outline

1. Discussion of mistakes on HW4 and suggestions for HW5, now due Monday Nov 5, 5pm:
   (a) examples of mistakes
   (b) a rule set that helps avoid mistakes in managing constants for individual domain elements.

2. Discussion of Löwenheim's Theorem (Smullyan p.61):
   (a) Smullyan's proof – direct from completeness
   (b) The Skolem Paradox – how do we justify a countable model of the real number axioms when we know that the real numbers are uncountable (even constructively). How do we justify a finite axiomatization of set theory in FOL?

3. Reflections on Judith Underwood's "almost constructive" proof of the completeness theorem (using the Fan theorem and Markov's Principle):
   (a) Recall Judith's theorem from Lecture 18 and note that "Russian constructivists" accept Markov's Principle.
   (b) If interested, see the supplemental material by Berardi and Valentini on how to define countable constructive models of FOL and how to prove completeness for classical minimal logic constructively.

CS 4860 Lecture 19 continued

The constructive completeness proof for FOL was discovered by Krivine. Here is what Berardi and Valentini say about it:

> It is useful to stress that the whole proof that $\Gamma \vdash A$ is provable whenever it is true in all minimal models is intutionistic. Hence, it defines, through realizability interpretation, an algorithm turning a model-theoretical proof of $\Gamma \Vdash_{\min} A$ into a classical proof of $\Gamma \vdash A$. Krivine proposed to call this algorithm a "decompiler", because it recovers a first order formal proof out of an informal proof which uses set theory; here we think of informal proofs as "compilations", in set theory, of formal proofs. It is a bit puzzling that, even if Krivine's algorithm was recently implemented by Raffalli, no explicit description of it is currently available.

---

1. Mistakes on HW4. We must be careful about managing constants. In the notes from Prof. Kreitz, we do not use "declarations" of individual variables in the rules. In the version of the rules used to prove consistency we used them and discussed in lecture the relationship between the two sets of rules and how they help clarify Smullyan's restrictions on when variables must be <u>new</u>. Consider these examples.

	format 1	format 2	
$\vdash \forall x (Px \supset C) \supset \exists x.Px \supset C$	$\supset R$ new f	$\lambda f.$ ___	
$f: \forall x.(Px \supset C) \vdash \exists x.Px \supset C$	$\supset R$ new e	$\lambda e.$ ___	
$f: \forall x.(Px \supset C), e: \exists x.Px \vdash C$	$\exists L$	spread(e; $x_0, p.$ ___)	
" " ", $x_0 : D,\ p : Px_0 \vdash C$	$\forall L\ x_0$	apseq(f; x_0; $v.$)	$v = f(x_0)$
$v : Px_0 \supset C,\ x_0 : D,\ p : Px_0 \vdash C$	$\supset L$	apseq(v; p; $u.$ ___)	
$\vdash Px_0$	hyp(p)		
$u : C \vdash C$	hyp(u)	$u = v(p)$	

Notice how we keep track of the constants using $x_0 : D$.

CS 4860 Lecture 19 continued

Here is a more subtle example.

	format 1	format 2
$\vdash \forall x.(P_x \supset C) \supset \exists x.(P_x \supset C)$	$\supset R$	$\lambda f.$ ___
$f : \forall x.(P_x \supset C) \vdash \exists x.(P_x \supset C)$		

constructively we are stuck, classically we assume that every domain D is non empty with individual $d_0 \in D$. We can use this d_0 a la Smullyan and get

$f : \forall x.(P_x \supset C \vdash P d_0 \supset C$ but this leaves no record of this choice.

* It is good practice in the case where we use the classical assumption to make it explicit and see if the rest of the proof can be done constructively. Here is that proof.

$\vdash (\exists x.\mathrm{True} \wedge \forall x.(P_x \supset C)) \supset \exists x.(P_x \supset C)$	$\supset R$	$\lambda h.$
$h : \exists x.\mathrm{True} \wedge \forall x.(P_x \supset C) \vdash \exists x.(P_x \supset C)$	$\& L$	spread(h; e, f. ___)
$e : \exists x.\mathrm{True},\ f : \forall x.(P_x \supset C) \vdash \exists x.(P_x \supset C)$	$\exists L$	spread(e; x_0, t. ___)
$x_0 : D,\ t : \mathrm{True},\ f : \forall x(P_x \supset C) \vdash \exists x(P_x \supset C)$	$\exists R, x_0$	$\langle x_0,$ ___ $\rangle$
$\vdash P_{x_0} \supset C$	$\forall L, x_0$	apseq(f; x_0; v. ___) $v = f(x_0)$
$,\ ,\ v : P_{x_0} \supset C \vdash P_{x_0} \supset C$	by v	

CS 4860 Lecture 19 continued

2. Discussion of Löwenheim's Theorem

The completeness proof shows how to build a countable domain D to satisfy any first order formula. This is true even if we start with axioms about sets. There is a theory of sets called Bernays/Gödel (BG) or sometimes von Neumann Bernays, Gödel (NBG or vBG) which uses <u>sets</u> and <u>classes</u> which can be axiomatized in FOL using two sorts, Set(x) and Class(x), and only finitely many first-order axioms. All theorems provable in this theory can be satisfied in a countable model. This is true even for set theoretical statements that prove that a set, like the reals ($\mathbb{R}$), is uncountable. This is sometimes called <u>Skolem's Paradox</u>. Even vastly huge sets such as $\mathbb{R}^{\mathbb{R}}$ can be "modeled" in a countable model.

What kind of constructive sense could we make out of this? Are these results "constructively true"? Yes, they are. Do they have computational meaning? Yes, they do. What is it? That's a fascinating question.