Math 260A — Mathematical Logic — Scribe Notes
UCSD — Winter Quarter 2012
Instructor: Sam Buss

Notes by: James Aisenberg
April 27th

1 The decision problem for First order logic

Fix a finite language L. Define the set:

$$\text{Valid}_L = \{\phi : \ \phi \text{ is a valid } L\text{-sentence}\}.$$

But maybe to be more precise we should fix a finite alphabet Σ that is capable of coding L-formulas, in which case we have

$$\text{Valid}_L = \{\omega \in \Sigma^* : \ \omega \text{ codes a valid } L\text{-sentence}\}.$$

Example 1. We will start with a situation where Valid_L is decidable. A word of caution: this is not the general behavior. We take as our example L,the language of pure equality. That is, $L = \{=\}$.

Claim 1. *The language of pure equality is decidable.*

Let $n \in \mathbb{N}$. Define $GE_n := \exists x_1 \cdots \exists x_n \bigwedge_{i<j} x_i \neq x_j$. Let $\Gamma = \{GE_n : n \geq 2\}$. Γ is ω-categorical. Also κ-categorical $\forall \kappa \geq \omega$. Thus Γ is complete and consistent.

Consider the model $M_\infty = \{0, 1, 2, \dots\}$. $M_\infty \models \Gamma$.

Theorem 1. *We claim that* $\{\Psi : \Gamma \models \Psi, \ \Psi$ *an* L*-sentence*$\}$ *is decidable.*

Proof. Given Ψ, by completeness, either $\Gamma \vdash \Psi$ or $\Gamma \vdash \neg\Psi$. Consider the following algorithm.

- Input Ψ.
- Loop over $i = 1, 2, 3, \dots$
  - Enumerate all possible proofs in the L-language with fewer than i symbols.
  - If one of them is a proof of Ψ, then accept.
  - If one of them is a proof of $\neg\Psi$, then reject.

This algorithm will always terminate, since there is either a proof of Ψ or there is a proof $\neg\Psi$, and such a proof will eventually appear in the enumeration of all L-language proofs. $\square$

Lemma 1. *If $M_\infty \models \Psi$, then $\exists n$ s.t. $GE_n \models \Psi$ (so $GE_n \vdash \Psi$).*

Proof. Since $M_\infty \models \Psi$, it follows that $\Gamma \models \Psi$ by completeness of Γ. By compactness, there are finitely $n_1, \ldots, n_k$ such that $GE_{n_1}, \ldots, GE_{n_k} \models \Psi$. Let $n = \max\{n_1, \ldots, n_k\}$. Thus $GE_n \models \Psi$. $\square$

Claim 2. *If $M_\infty \not\models \Psi$, there is a finite n such that $M_n \not\models \Psi$, where $M_n = \{0, 1, 2, \ldots, n-1\}$.*

Proof. By the lemma above, there exists an n s.t. $GE_n \models \neg\Psi$. $\square$

Corollary 1. *If $\not\models \Psi$, then there is an n such that $M_n \not\models \Psi$.*

Theorem 2. *$Valid_L$ is decidable.*

Proof. Here is the algorithm:

- Input Ψ
- Loop for $i = 1, 2, 3, \ldots$.
  - (A) If $M_i \not\models \Psi$, then reject
  - (B) Enumerate all L-proofs of fewer than i symbols, if we find an L-proof of Ψ, then accept

  End loop.

End algorithm.

It is worth hand waving that (A) and (B) are recursive properties. Also, from what we have build up we know the algorithm always terminates. Why? Well either Ψ has an L-proof, or there is some finite M_n that witnesses the invalidity of Ψ. $\square$

2 Undecidability of First order validity

The idea is that first order logic is sufficiently expressive to describe Turing machine behavior.

Fix a Turing machine M with $\Gamma = \{b, 0\}$. M has an infinite tape in one direction with states $q_0, \ldots, q_l$. Now let L be a language consisting

of $\{0, S, \tau, \sigma, q\}$. The meaning of these symbols are as follows: 0 is zero, S is to mean the successor function, τ, σ and q are binary relations. For convenience, let $S^t(0) = S(S(\cdots S(0)\ldots))$, t many applications of the successor function to itself. The meaning of $q(S^t(0), S^l(0))$ is that the Turing machine M is in state l at time t. The meaning of $\tau(S^t(0), S^p(0))$ is that Turing machine M has tape head at position p at time t. The meaning of $\sigma(S^t(0), S^p(0))$ is that the Turing machine M has a zero at position p at time t.

2.1 Axioms for 0 and S

We first make sure that zero and successor are reasonably well-behaved.

- $(\forall x)(S(x) \neq 0)$
- $(\forall x)(\forall y)(S(x) = S(y) \supset x = y)$
- $(\forall x)(x \neq 0 \supset (\exists y)(S(y) \neq x))$.

We will call the conjunction of these axioms Ψ.

2.2 Initialization axioms

- $q(0, 0)$
- $\tau(0, 0)$
- $(\forall x)(\neg\sigma(0, x))$
- $(\forall x)(\neg\tau(0, S(x))$
- $(\forall x)(\neg q(0, S(x))$.

These initialization axioms assert that the tape head is in position 0 at time zero. The tape is blank, and the machine is in the start state, q_0. We will refer to the conjunction of these axioms as χ_1.

2.3 Uniqueness axioms

- $(\forall x)(\forall y)(\forall z)(y \neq z \supset \neg q(x, y) \vee q(x, z)$.
- $(\forall x)(\forall y)(\forall z)(y \neq z \supset \neg\tau(x, y) \vee \tau(x, z)$.

These axioms assert that a machine is not in two different states at the same time, and also that the tape head isn't at two different positions at the same time. We will refer to them collectively as χ_2.

2.4 "Next configuration" axioms

These axioms will be used to encode the transition function of the Turing machine M. This is the only part of the construction that is specific to the Turing machine M. We will just give some examples of how to build axioms out of transition functions.

- Suppose the transition table has a line $(q_0, 0), (b, N, q_1)$. In other words, when the machine is in state q_0 and has a 0 on the tape head, write a blank, don't move the tape head, and go to the state q_1. This is encoded by the axiom:

$$(\forall t)(\forall p)(q(t,0) \wedge \sigma(t,p) \wedge \tau(t,p) \supset q(S(t), S(0)) \wedge \tau(S(t),p) \wedge \neg\sigma(S(t),p))$$

- Suppose the transition table has a line $(q_1, b), (0, R, q_2)$. In other words, when the machine is in state 1 with a blank on the tape head, write a zero, move to the right and enter state 2. To encode this, we want to use subtraction, but we don't have that in the language. Instead, we just use another quantifier. But we have to ensure that if we are already in the first tape location, and we try to move to the right, we remain in the first tape position. This is encoded by the following axiom:

$$(\forall t)(\forall p)(\forall p')(q(t, S(0)) \wedge \sigma(t,p) \wedge \tau(t,p) \wedge (S(p') = p \vee (p' = 0 \wedge p = 0)) \supset$$

$$q(S(t), S(S(0))) \wedge \tau(S(t), p') \wedge \neg\sigma(S(t), p))$$

There will be lines similar to the above for each line in the transition table. We refer to their conjunction as χ_3.

2.5 No change axioms

$$(\forall t)(\forall p)(\neg\tau(t,p) \supset (\sigma(t,p) \iff \sigma(S(t),p)))$$

This asserts that the tape can only change at a particular position when the tape head is over that position at a particular time. Call this χ_4.

2.6 Halt axiom

$$(\exists x)(q(x, S^h(0)))$$

where q_h is the halt state of M. We refer to this axiom as H. This could get confusing, because we also use H to denote the halting set.

Theorem 3. *M halts iff $\Psi \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \chi_4 \supset H$ is valid.*

Proof. And model of $\Psi \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \chi_4$ has a "standard part" isomorphic to the integers. On this part, σ, τ, and q faithfully reflect M's computation. So if M halts, there exists a standard t such that $q(t, S^h(0))$ holds. Hence $\Psi \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \chi_4 \supset H$ is valid. Now suppose M doesn't halt. Take the standard model of $\Psi \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \chi_4$. In this model H is false. So we're done. □

Corollary 2. *There is a many-one reduction from the Halting set, H to $Valid_L$.*