Undecidability of FOL

Computability and Logic

Reducing the Halting Problem to the FOL Consequence Problem

- We'll show that FOL is undecidable by showing that we can cast the Halting Problem as a FOL problem.
- That is, we will use FOL statements to describe the configurations and workings of a Turing-machine M when given some kind of input tape T, and thus pose the Halting Problem as a logic problem.
- Thus, we reduce the Halting problem to the FOL Consequence problem.
- So, if we could solve FOL Consequence (the Enscheidungsproblem!) we would thereby solve the Halting Problem.
- Since we (most likely) can't solve the Halting Problem, we therefore can't solve the FOL Consequence problem either: FOL is undecidable!

A Simple Class of Turing-Machines

- To keep things as simple as possible, we'll restrict ourselves to the following kind of Turing-machine:
  - Binary alphabet
  - One-sided infinite tape
  - Starting state is q_1
  - Halting state is q_0
- It can easily be shown that, even as defined over this restricted kind of machine, the Halting Problem is still unsolvable.

Describing Turing-Machines

- We'll describe Turing-machines using the following predicates, functions, and constants, that have the provided standard interpretation I with its domain the set of natural numbers:
  - 0: the number 0
  - s: the successor function
  - S(t,i): After t steps, the machine is in state i
  - C(t,i): After t steps, the machine is looking at cell i
  - M(t,i): After t steps, square i contains a 1 ('mark')

Numbering the squares:

0	1	1	1	1	0	1	0	0
0	1	2	3	…				

Describing Transitions

- All transitions naturally translate in FOL expressions. Example:
  - Transition: <i, 0, 1, j> ("if you are in state i, and see a 0, write a 1, and go to state j")
  - Corresponding FOL statement:

$$\forall t\ \forall x\ [(S(t,i) \wedge C(t,x) \wedge \neg M(t,x)) \rightarrow$$

$$(S(s(t),j) \wedge C(s(t),x) \wedge M(s(t),x) \wedge$$

$$\forall y\ (y \neq x \rightarrow (M(s(t),y) \leftrightarrow M(t,y)))))]$$

Set $D_{M,T}$

- Every transition A gets a corresponding statement D_A. All these statements will be put into set $D_{M,T}$
  - Note: Here is the easiest way to deal with a Left action <i, 0, L, j>:

    $\forall t\ \forall x\ [(S(t,i) \wedge C(t,s(x)) \wedge \neg M(t,s(x))) \rightarrow$

    $(S(s(t),j) \wedge C(s(t),x) \wedge \forall y\ (M(s(t),y) \leftrightarrow M(t,y))]$
  - Also, let's say that when trying to move left of the left-most square, the Turing-machine will stay at the left square. Thus, add:

    $\forall t\ \forall x\ [(S(t,i) \wedge C(t,0) \wedge \neg M(t,0)) \rightarrow$

    $(S(s(t),j) \wedge C(s(t),0) \wedge \forall y\ (M(s(t),y) \leftrightarrow M(t,y))]$

Starting Configuration

- Use the statement S to describe the starting configuration. Assume that squares x_1, … x_k are the non-zero squares:
- $S = S(0,s(0)) \wedge C(0,0) \wedge M(0,x_1) \wedge \ldots \wedge M(0,x_k) \wedge \forall y\ (\neg(y=x_1 \vee \ldots \vee y=x_k) \rightarrow \neg M(0,y))$
- S will also be put into $D_{M,T}$

Halting Statement

- The following statement H captures what it means for the machine to halt:
- $H = \exists t\, S(t,0)$

Description Statement

- The following statement $D_{M,T,t}$ describes the machine and tape configuration we find ourselves in after t steps of operation. Thus, where x_1, … x_k are the non-zero squares at that time, where x_m is the square the machine is at, and q_i the state it is in:
- $D_{M,T,t} = S(t,i) \wedge C(t,x_m) \wedge M(0,x_1) \wedge \ldots \wedge M(0,x_k) \wedge \forall y\, (\neg(y=x_1 \vee \ldots \vee y=x_k) \rightarrow \neg M(0,y))$

The Central Theorem

- We want to claim that given any machine M and input tape T, it is true that for any step t: $D_{M,T} \vDash D_{M,T,t}$
- Proof: By induction on t
- Base: t = 0. $D_{M,T,0}$ = S, and S $\in D_{M,T}$, so $D_{M,T} \vDash D_{M,T,0}$
- Step: Here, we just have to show that for the transition A that applies at time t, the statement $D_{M,T,t+1}$ is implied by the statement D_A that is in $D_{M,T}$, together with statement $D_{M,T,t}$. Thus, the inductive hypothesis that $D_{M,T} \vDash D_{M,T,t}$ implies that $D_{M,T} \vDash D_{M,T,t+1}$ as desired.

Example

- Again, let's consider A = <i, 0, 1, j>.
- That is, suppose that this is the next transition the machine has to take after having taken t steps, that x_1, … x_k are the non-zero squares at that time, and that x_m is the square the machine is at.
- Thus, by inductive hypothesis, we have:
  - $D_{M,T,t} = S(t,i) \wedge C(t,x_m) \wedge M(t,x_1) \wedge \ldots \wedge M(t,x_k) \wedge \forall y\ (\neg(y=x_1 \vee \ldots \vee y=x_k) \rightarrow \neg M(t,y))$
- We also have:
  - $D_A = \forall t\ \forall x\ [(S(t,i) \wedge C(t,x) \wedge \neg M(t,x)) \rightarrow (S(s(t),j) \wedge C(s(t),x) \wedge M(s(t),x) \wedge \forall y\ (y \neq x \rightarrow (M(s(t),y) \leftrightarrow M(t,y))))]$
- Eliminate/instantiate the universals with t,i, and x_m, and you'll find that you can derive:
  - $D_{M,T,t+1} = S(s(t),i) \wedge C(s(t),x_m) \wedge M(s(t),x_1) \wedge \ldots \wedge M(s(t),x_k) \wedge M(s(t),x_m) \wedge \forall y\ (\neg(y=x_1 \vee \ldots \vee y=x_k \vee y=x_m) \rightarrow \neg M(s(t),y))$
  - (actually, this is not quite true: you need that for any two numbers x and y: x = y iff $D_{M,T} \models \mathbf{x = y}$. But this is easily achieved by adding PA1 and PA2 to $D_{M,T}$)

What we Get

- Using the Central Theorem, it is easy to show that machine M with input T halts if and only if $D_{M,T} \models H$:
  - If $D_{M,T} \models H$ then for any interpretation that is a model for $D_{M,T}$ is a model for H. Since the standard interpretation I is a model for $D_{M,T}$, we thus know that $I \models H$, i.e. machine M with input T will halt.
  - If machine with input T halts, then there is some time t_H at which the machine halts. Since by the Central Theorem $D_{M,T} \models D_{M,T,tH}$, where $D_{M,T,tH} \models H$, we thus have that $D_{M,T} \models H$.