First Order Logic is Undecidable

Lee Danilek, Priyanka Krishnamurthi, Andrew Malta, ...

March 25, 2016

(1) The decision problem for a property is solvable if there is a mechanical test which, applied to *any* object of the appropriate sort, eventually (after a finite number of steps) classifies that object correctly as a positive or negative instance of that property.

In first order logic the decision problem deals with validity and satisfiability of sentences.

We will prove that there does not exist a mechanical negative test for deciding if a sentence in first order logic is valid.

In this presentation, we will demonstrate that if there existed such a routine, it would imply that we could mechanically determine whether a Turing machine will eventually halt. Since we know this is not possible, there can be no mechanical routine to determine validity.

- Δ is a finite set of sentences that describe the operation of the Turing machine.
- n is a number, the input to the Turing machine.
- H is a sentence such that $\Delta \vdash H \iff$ the machine does eventually halt when given input n, when H is interpreted in $\mathscr{I}$.
- $\mathscr{I}$ is an interpretation for H and the sentences in Δ. The variables range over the integers, and it uses the following definitions:
- Q_i for $0 \leq i \leq r$ is a binary predicate function.
  $tQ_ix \iff$ at time t the machine is in state q_i, scanning square number x.
- S_j for $0 \leq j \leq r$ is a binary predicate function.
  $tS_jx \iff$ at time t the machine is scanning the symbol S_j, scanning square number x.
- $<$ is the standard less-than binary predicate.
- 0 is the standard zero function.
- $'$ is the standard successor function.

Thus, if we could solve the decision problem for validity of sentences we could determine whether the machine eventually halts, because $\Delta \vdash H$ if and only if a certain sentence is valid, namely the condition whose antecedent is the conjunction of all sentences in Δ and whose consequent is H: $\Delta_1 \wedge \Delta_2 \wedge \cdots \rightarrow H$.

The squares of the tape are numbered by integers, and moments of time are integers. Each moment in time is a single step in the Turing machine. The machine begins at time $t = 0$ in square $x = 0$ and stops when the machine halts.

If $t < 0$ or $t >$ the halting time, $tQ_ix \mapsto 0$ and $tS_jx \mapsto 0$.

Rule: $q_i \rightarrow S_j : S_k \rightarrow q_m$.

$$\forall t \forall x \forall y \{[tQ_ix \& tS_jx] \rightarrow [t'Q_mx \& t'S_kx \& (y \neq x \rightarrow (tS_0y \rightarrow t'S_0y) \& \ldots \& (tS_ry \rightarrow t'S_ry))]\}$$

Rule: $q_i \rightarrow S_j : R \rightarrow q_m$.

$$\forall t \forall x \forall y \{[tQ_tx \& tS_jx] \rightarrow [t'Q_mx' \& (tS_0y \rightarrow t'S_0y) \& \ldots \& (tS_ry \rightarrow t'S_ry)]\}$$

Rule: $q_i \rightarrow S_j : L \rightarrow q_m$

$$\forall t \forall x \forall y \{[tQ_i x' \& tS_j x'] \rightarrow [t' Q_m x \& (tS_0 y \rightarrow t' S_0 y) \& \ldots \& (tS_r y \rightarrow t' S_r y)]\}$$

Starting condition:

$$\mathbf{o}Q_1\mathbf{o} \ \& \ \mathbf{o}S_1\mathbf{o}' \ \& \ \mathbf{o}S_0\mathbf{o}' \ \& \ldots \& \ \mathbf{o}S_1\mathbf{o}^{(n-1)} \ \& \ \forall y[(y \neq \mathbf{o} \ \& \ y \neq \mathbf{o}' \ \& \ldots \& \ y \neq \mathbf{o}^{(n-1)}) \rightarrow \mathbf{o}S_0 y]$$

One sentence says that each integer is the