Logic and Proof Hilary 2016

Undecidability of Validity and Satisfiability

James Worrell

In this lecture we consider the computational problems of determining validity and satisfiability of first-order formulas. We show that both validity and satisfiability are undecidable. On a positive note, we show that validity is *semi-decidable* (or *recursively enumerable*). Intuitively this is because a valid formula F has a finite witness of its validity—namely a finite resolution refutation of $\neg F$. Satisfiability is not semi-decidable however. Intuitively, there need not be finite witness that a given formula is satisfiable. In particular, there are satisfiable formulas that have no finite models.

1 Semi-Decidability of Validity

Theorem 1. Validity of first-order formulas is semi-decidable.

Proof. Recall that a semi-decision procedure for validity should halt and return "valid" when given a valid formula as input, but otherwise may compute forever. Such a procedure is as follows. (Note that there is no loss of generality in restricting to closed formulas since F is valid iff $\forall x F$ is valid.)

Semi-Decision Procedure for Validity
Input: Closed formula F
Output: Either that F is valid or compute forever

Compute a Skolem-form formula G equisatisfiable with $\neg F$
Let $G_1, G_2, \ldots$ be an enumeration of the Herbrand expansion $E(G)$
for $n = 1$ to ∞ **do**
begin
  if $\square \in \text{Res}^*(G_1 \cup \ldots \cup G_n)$ **then** stop and output "F is valid"
end

The procedure relies on the fact that F is valid if and only if $\neg F$ is unsatisfiable. To show unsatisfiability of $\neg F$ we transform it into an equisatisfiable formula G in Skolem form. Then, by the refutation completeness of ground resolution, G is unsatisfiable iff there is a ground resolution refutation of G. If such a refutation exists it will eventually be discovered by the procedure. Note that for each n the set of clauses $\text{Res}^*(G_1 \cup \ldots \cup G_n)$ that can be derived by resolution from $G_1 \cup \ldots \cup G_n$ is computable in a finite amount of time. (Here we regard each G_i as a set of clauses.) $\square$

In the above proof it was convenient to invoke the refutation completness of ground resolution. However ulitmately the result relies on Herbrand's Theorem and the Compactness Theorem for propositional logic, which together guarantee that that F is valid if and only if some finite subset of $E(G)$ is unsatisfiable.

2 Undecidability of Validity and Satisfiability

In this section we recall the definition of *Post's Correspondence Problem* (PCP), and show how to transform a given instance of this problem into a first-order formula such that the instance has a solution if and only if the formula is valid. It follows that the validity problem for first-order logic is undecidable.

An instance of Post's correspondence problem consists of a finite set of tiles. Each tile has a bit-string on the top and a bit-string on the bottom. For example, we could have tiles

$$\left\{ \left[\begin{array}{c} 1 \\ 101 \end{array} \right], \left[\begin{array}{c} 10 \\ 00 \end{array} \right], \left[\begin{array}{c} 011 \\ 11 \end{array} \right] \right\}$$

A solution to the problem is a sequence of tiles, allowing the same tile multiple times, such that the top string equals the bottom string. In the above example a solution is

$$\left[\begin{array}{c} 1 \\ 101 \end{array} \right] \left[\begin{array}{c} 011 \\ 11 \end{array} \right] \left[\begin{array}{c} 10 \\ 00 \end{array} \right] \left[\begin{array}{c} 011 \\ 11 \end{array} \right]$$

In general, an instance of Post's correspondence problem is a finite set of pairs of bit-strings $\mathbf{P} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$, where $x_i, y_i \in \{0,1\}^*$. A solution of $\mathbf{P}$ is a sequence $i_1, i_2, \ldots, i_n$ such that $x_{i_1} x_{i_2} \ldots x_{i_n} = y_{i_1} y_{i_2} \ldots y_{i_n}$. In the above example one solution is the sequence $1, 3, 2, 3$. Clearly for each particular PCP instance, the set of *potential solutions*, i.e., sequences of tiles is infinite. Thus solving an instance of PCP involves searching an infinite set.

We encode this problem in first-order logic using a signature with constant symbol e, two unary function symbols f_0, f_1 and a binary relation symbol P. The ground terms over this signature can be considered as bit-strings, e.g., the term $f_1(f_1(f_0(e)))$ represents the bit-string 110. In general, for a bit-string $b_1 \ldots b_t \in \{0,1\}^*$ we denote the term $f_{b_1}(\ldots(f_{b_t}(x))\ldots)$ by $f_{b_1 \ldots b_k}(x)$.

Our goal is to transform a given instance $\mathbf{P}$ of Post's correspondence problem into a closed formula F such that $\mathbf{P}$ has a solution if and only if F is valid. We first give the idea of the construction in the above example. Consider the following three formulas:

$$\begin{array}{lcl} F_1 & = & P(f_1(e), f_{101}(e)) \wedge P(f_{10}(e), f_{00}(e)) \wedge P(f_{011}(e), f_{11}(e)) \\ F_2 & = & \forall u \, \forall v \, (P(u,v) \rightarrow P(f_1(u), f_{101}(v)) \wedge P(f_{10}(u), f_{00}(v)) \wedge P(f_{011}(u), f_{11}(v))) \\ F_3 & = & \exists u \, P(u,u) \, . \end{array}$$

We claim that $F_1 \wedge F_2 \rightarrow F_3$ is valid if and only if the PCP instance has a solution.

Given a general instance $\mathbf{P} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ of PCP we have the formulas

$$\begin{array}{lcl} F_1 & = & \displaystyle\bigwedge_{i=1}^{k} P(f_{x_i}(e), f_{y_i}(e)) \\ F_2 & = & \forall u \, \forall v \displaystyle\bigwedge_{i=1}^{k} (P(u,v) \rightarrow P(f_{x_i}(u), f_{y_i}(v))) \\ F_3 & = & \exists u \, P(u,u) \, . \end{array}$$

Proposition 2. $\mathbf{P}$ has a solution if and only if $F_1 \wedge F_2 \rightarrow F_3$ is valid.

Sketch. Suppose that $F_1 \wedge F_2 \to F_3$ is valid. Consider the Herbrand structure $\mathcal{H}$ for which

$$P_{\mathcal{H}} = \{(f_u(e), f_v(e)) : \exists i_1 \ldots \exists i_t \,.\, u = x_{i_1} \ldots x_{i_t} \text{ and } v = y_{i_1} \ldots y_{i_t}\}\,.$$

Clearly $\mathcal{H}$ satisfies $F_1 \wedge F_2$. Thus it must hold that $\mathcal{H}$ satisfies F_3. But this means that **P** has a solution.

Conversely suppose that **P** has a solution. We show that $F_1 \wedge F_2 \to F_3$ is valid. To this end, consider a structure $\mathcal{A}$ that satisfies $F_1 \wedge F_2$. Then we can show by induction on t that for any sequence of tiles $i_1 \ldots i_t$, $\mathcal{A} \models P(f_u(e), f_v(e))$, where $u = x_{i_1} \ldots x_{i_t}$ and $v = y_{i_1} \ldots y_{i_t}$. But since **P** has a solution, $\mathcal{A} \models P(f_u(e), f_u(e))$ for some string u. Thus $\mathcal{A} \models F_3$. □

Corollary 3. The satisfiability and validity problems for first-order logic are undecidable.

Proof. Undecidability of validity follows from undecidability of Post's Correspondence Problem, which was shown in the *Models of Computation* course. Furthermore, since F is valid if and only if $\neg F$ is unsatisfiable, undecidability of satisfiability is immediate from undecidability of validity. □

Look in the notes It follows from Theorem 1 and Corollary 3 that satisfiability is not even semi-decidable. Indeed if satisfiability were semi-decidable then we could decide validity as follows. Given a formula F, either F is valid or $\neg F$ is satisfiable. Thus we could decide validity of F by simultaneously running a semi-decision procedure for validity on F and a semi-decision procedure for satisfiability on $\neg F$.

Corollary 4. Satisfiability of first-order formulas is not semi-decidable.