Decidability and Undecidability

2/17/2016
Pete Manolios
Theory of Computation
Models of Computation

- Regular: finite state machine
- CF: + stack
- Turing machine: + infinite tape
- Decidable (recursive): yes/no
- Recognizable (r.e.): yes
Decidability

• We will use Church-Turing thesis

• What that means is we’ll describe and think about algorithms just like you did in algorithms class

• Because using TMs is really tedious and painful

• Because we “know” that TMs ≡ pseudo code
Decidability

- D is a DFA that accepts w
- N is an NFA that accepts w
- D is a DFA that accepts a non-empty language
- A, B are DFAs and L(A) = L(B)
- C is a CFL that accepts w
- C is a CFL that accepts a non-empty language
- But, A, B are CFLs and L(A) = L(B) **not** decidable
- When the model of comp increases in power
- Your ability to analyze it decreases
Undecidability

- Limits of what can be done with a computer
- Of broad intellectual, philosophical interest
  - Can humans solve problems TMs can’t?
  - Turing test: can machines behave like humans?
  - Can machines have consciousness?

http://imgs.xkcd.com/comics/suspicion.png
Counting Infinities

• f: A → B is *injective* or *one-to-one* if it a≠b ⇒ f(a) ≠f(b)

• f: A → B is *surjective* or *onto* if it \( \cup_{a \in A} \{f(a)\} = B \)

• f is *bijective* or a *correspondence* if it is both injective and surjective

• If f: A → B is bijective then each element of A maps to a unique element of B and conversely

• Given A,B if \( \exists \) a bijection f: A → B then \(|A| = |B|\): they have the same size

• This makes intuitive sense for finite sets, but has non-intuitive consequences for infinite sets

• \(|\{a, b, c, d\}| = |\{1, 21, 3, 2\}| = |\{d, a, f, b, d, a\}| = 4\)

• \(|\mathbb{N}| =? |\mathbb{N} \setminus \{0, 1, 2\}|\)

• \(|\mathbb{N}| = |\{n \in \mathbb{N}: n \text{ is even}\}| = |\mathbb{Z}| = |\mathbb{Q}| = \omega\)

• If \(|A| \leq \omega\) it is *countable*. \(\omega\) is the first infinite *ordinal number*. 

\(\mathbb{N} \setminus \{0, 1, 2\} : 3, 4, 5, 6, 7, \ldots\)

\(\mathbb{Z} : 0, 1, -1, 2, -2, \ldots\)
Theorem: \( \lvert \mathbb{Q} \rvert = \omega \)

Have to enumerate the rationals. Here’s how:

Note: no duplicates

Theorem: A countable union of countable sets is countable!
Theorem: $|\mathbb{R}| > \omega$

- $\mathbb{R}$ is *uncountable*: infinite and no bijection between $\mathbb{R}$ and $\omega$

- Clearly $|\mathbb{R}| \geq \omega$. We show $|\mathbb{R}| \neq \omega$.

- The proof is by contradiction.

- Suppose that there is a bijection, say:

- We derive a contradiction by showing that it can’t include every real number

- Select $r_i$ to differ from digit $i$ of $f(i)$

- Don’t use 0, 9 (because 0.9999... = 1.0000...)

- We showed $|\mathbb{R}| > \omega$ and that $|A| > \omega$ for $A$ is any non-empty interval of reals

- This technique is called *diagonalization* and is due to Cantor (1873)

Existence of the Undecidable

- There exist languages that are not Turing-recognizable (R.E.)
- So they are also not Turing-decidable (R.) either
- And it turns out that *most* languages are not Turing-recognizable!
- Observe: If $|\Sigma| \leq \omega$ then the set of all strings, $\Sigma^*$, is countable
- Observe: The set of all TMs is countable (each is described by a finite string of symbols over a finite alphabet)
- Observe: $B = \{0,1\}^\omega$ is uncountable (binary representation of reals in $[0..1]$)
- Observe: There is a bijection between $\mathcal{L}$, the set of languages, and $B$. Use the characteristic function: given $L \in \mathcal{L}$, $f(L)=<s_1 \in L, s_2 \in L, s_3 \in L, ...>$
- So, $|\mathcal{L}| > \omega$ and most languages are not Turing-recognizable
A_{TM} is R.E.

• $A_{TM} = \{ <M, w>: M \text{ is a TM that accepts } w \}$

• Theorem: $A_{TM}$ is R.E. (Turing recognizable)

• Proof: Consider TM $U$: On input $<M, w>$ it runs $M$ on $w$. If $M$ halts and accepts $w$, accept. If $M$ halts and rejects $w$, reject.

• Note: $U$ is a universal Turing machine
A\textsubscript{TM} is Undecidable

• Theorem: A\textsubscript{TM} is Undecidable. (A\textsubscript{TM} = \{<M,w>: M is a TM that accepts w\})

• Proof: Suppose there exists a TM H that decides A\textsubscript{TM}. Then, for any input <M,w>, H accepts if M accepts w and rejects otherwise.

• Consider a TM D that takes an input <M>, the description of M, and takes the following steps.
  • Run H on <M,<M>>
  • If H accepts, reject
  • If H rejects, accept

• Since H is a decider, D is also a decider.

• Consider D's output on <D>. If D accepts, then this implies that according to H, D rejects <D>. If D rejects, then this implies that according to H, D accepts <D>. But this is a contradiction.
Diagonalization?

- Another way to see this is that we have essentially proved that the language \{\langle M \rangle : M \text{ accepts } \langle M \rangle \} is undecidable. How did we do this?

- Number the machines \(M_1, M_2, \ldots\). Suppose the above language is decidable by a TM \(E\).

- Define \(D\) to be a machine that on input \(\langle M \rangle\), accepts if \(E\) rejects \(\langle M \rangle\), and rejects if \(E\) accepts \(\langle M \rangle\).

- This is precisely flipping the diagonal entries of the matrix in which the columns list the machines \(M_1, M_2, \ldots\), and the rows list the inputs \(\langle M_1 \rangle, \langle M_2 \rangle, \ldots\).

- If \(D\) is on this list, then we obtain a contradiction.
L and \( \neg L \) are RE then \( L \) is R.

- \( \neg L \) is the complement of \( L \): \( \Sigma^* \setminus L \)

- Theorem: If \( L \) and \( \neg L \) are Turing-recognizable, then \( L \) is decidable.

- Proof: Let \( M_1 \) and \( M_2 \) be TMs that recognize \( L \) and \( \neg L \). Given a string \( w \), exactly one of the following happens
  - \( M_1 \) accepts \( w \) or \( M_2 \) accepts \( w \)
  - TM \( M \) for deciding \( L \) simulates \( M_1 \) and \( M_2 \) in parallel, running one step of each on \( w \).
  - Within a finite number of steps, one of them will halt and accept.
  - If \( M_1 \) accepts, then \( M \) accepts. If \( M_2 \) accepts, then \( M \) rejects.
¬A_{TM} is not RE

• Corollary: ¬A_{TM} is unrecognizable (not RE)

• What is ¬A_{TM}?

• \{<M, w>: M is not a TM or M does not accept w\}

• Proof: A_{TM} is not decidable, so by previous theorem either A_{TM} or ¬A_{TM} is not RE, but A_{TM} is RE, so ¬A_{TM} is not.
Halting Problem

- \( \text{HALT}_{TM} = \{<M, w>: M \text{ halts on } w\} \)

- Theorem: \( \text{HALT}_{TM} \) is undecidable.

- Proof: We show that if \( \text{HALT}_{TM} \) is decidable, then so is \( A_{TM} \).

- Preview of reduction: We reduce from \( A_{TM} \) to \( \text{HALT}_{TM} \) (\( A_{TM} \leq \text{HALT}_{TM} \)).

- Suppose \( H \) is the decider for \( \text{HALT}_{TM} \). Then the decider \( A \) for \( A_{TM} \) is as follows. On input \( <M, w> \), \( A \) calls \( H \) on input \( <M, w> \). If \( H \) accepts, then \( A \) runs \( M \) on \( w \) and accepts if \( M \) accepts \( w \), rejecting otherwise. If \( H \) rejects, then \( A \) rejects.

- Consider \( <M,w> \) in \( A_{TM} \). Since \( M \) accepts \( w \), \( M \) halts on \( w \). So \( H \) accepts \( <M, w> \). Since \( M \) accepts and halts on \( w \), \( A \)'s call of \( M \) on \( w \) terminates in an accept state.

- Consider \( <M,w> \) not in \( A_{TM} \). There are two cases. The first is when \( M \) halts on \( w \) and rejects \( w \). So \( H \) accepts \( <M, w> \). \( A \)'s call of \( M \) on \( w \) terminates in a reject state. The second case is when \( M \) does not halt on \( w \). So \( H \) rejects \( <M, w> \), and so does \( A \).