Lecture 24: Halting Problem

CSCI 101
Spring, 2019

Kim Bruce

Diagonalization

- Show the number of functions from N to N is uncountable.
  - Proof by contradiction
    - Suppose countable. List them all: $f_0, f_1, \ldots$
    - Claim list is missing at least one. Define $g(n) = f_n(n) + 1$.
    - g is not included in f_i because for all n, g(n) != f_n(n) so g != f_n
    - Can be no listing of all functions from N to N.
    - Thus N →N is uncountable.

Diagonalization Redux

- Theorem: There are effectively computable total functions that are not included in the primitive recursive functions.
  - The set of primitive recursive functions is can be "effectively enumerated", so list them: $f_0, f_1, \ldots$
  - You showed all total.
  - Define d s.t. $d(n) = f_n(n) + 1$. Function d is total and not in list — therefore not primitive recursive.
  - *What did we need to know about primitive recursive functions for proof to work?*

Undecidability

Computations on Machines

- Look at the following languages:
  - E_{DFA} ={<M> | M is a DFA and L(M)=∅}
    EQ_{DFA} = {⟨M,N⟩ | M and N are DFAs and L(M) = L(N)}
    A_{DFA} = {⟨M,w⟩ | M is a DFA and w ∈L(M)}
- Showed before (informally) that these are decidable.
- Can do the same for PDA's.
  - First & third decidable, second is not!

What about TM's

- We'll see corresponding sets not decidable
  - and perhaps even not semi-decidable.
  - *Recall L semi-decidable means may not halt if answer no*
- Two more sets:
  - H_{TM} = {⟨M,w⟩ | M is a TM which halts on input w}
    $TOTAL_{TM}$ = {<M> | M halts on all inputs}
  - See later that neither is decidable.

Decision Problems with TM's

- Look at following sets:
  - A_{TM} = {⟨M,w⟩ | M is a TM and w ∈ L(M)}
  - H_{TM} = {⟨M,w⟩ | M is a TM which halts on input w}
  - $TOTAL_{TM}$ = {M | M halts on all inputs}
  - E_{TM} ={M | M is a TM and L(M)=∅}
- Halting easy for most algorithms, but:
  - times3(x: positive integer) =
    while x ≠ 1 do:
    If x is even then x = x/2.
    else x = 3x + 1

Universe of discourse

- Easy to determine if have encoding of a TM, so we'll ignore it when take complements, etc., so our universe of discourse will only consider those with valid TM encodings.

Semi-decidable

- A_{TM} and H_{TM} both semi-decidable using UTM.
- Show H_{TM} not decidable.
  - Let E be candidate TM to decide H_{TM}. Show can't be right.
  - From E, create TM D s.t. if input w, create <w,w> and simulate E on it (*i.e, it treats input as if of form <M,w>*)
    - If E rejects then make D accept and if E accepts, D loops forever
  - Now run D on <D> *Diagonal Argument*
    - If $\langle D,D\rangle \in H_{TM}$ then D halts on D, so E rejected <D,D> & $\langle D,D\rangle \notin L(E)$
    - If $\langle D,D\rangle \notin H_{TM}$ then D not halt on D, so E accepted, $\langle D,D\rangle \in L(E)$
    - Either way, $L(E) \neq H_{TM}$ with <D,D> in one but not other.

Decidable and Semi-decidable

- *Earlier:* If L and its complement are both semi-decidable then it is decidable.
  - Corollary: Complement of H_{TM} is not semi-decidable
- Note, if H_{TM} were decidable then every SD language would be decidable.
- Lots of other languages not decidable:
  - L_0 = {<M,w> | M on w eventually writes a 0}
  - ...

Undecidability

- E_{TM} ={<M> | M is a TM and $L(M)=\emptyset$}
  - Spose decidable. For each pair <M,w> define machine M'_w that throws away its input and simulates M on w and if it halts then accept.
  - Then $M'_w \in E_{TM}$ iff not($\langle M,w\rangle \in H_{TM}$).
  - Thus could use solution to E_{TM} to solve H_{TM}.
  - Therefore E_{TM} is not decidable.
- General procedure to show undecidability
  - Reduce halting problem to solving other problem.
  - Proofs are by contradiction

Undecidability

- $TOTAL_{TM}$ ={<M> | M is TM that halts on all inputs}
  - Spose decidable. Use to solve halting!
    - For each pair <M,w> define machine M'_w that throws away its input and simulates M on w, if if it halts.
    - Then $M'_w \in TOTAL_{TM}$ iff $\langle M,w\rangle \in H_{TM}$.
    - Thus could use solution to $TOTAL_{TM}$ to solve H_{TM}.
  - Therefore $TOTAL_{TM}$ is not decidable.

More Undecidability

- L_ε = {<M> | M halts on empty tape}
  - Given M, w, create machine M_w that writes w and then simulates M on that w.
  - Claim <M_w> ∈ L_ε iff <M,w> ∈ H_{TM}.
  - Therefore L_ε not decidable

Another Example

- $L_{\varepsilon 0}$ = {<M> | M on ε eventually writes a 0}
  - Given M, rewrite to replace any occurrences of 0 in transitions by new character φ.
    - If 0 in input alphabet replace all occurrences of 0 on input by φ.
  - Modify again so that if it ever goes into a halt state then writes 0 on tape. Call this machine M'
  - One last modification: Erase input, write w, then run M' on w. Call new machine M'_w.
  - Claim < M'_w > ∈ $L_{\varepsilon 0}$ iff <M,w> ∈ H_{TM}.
  - Therefore can use $L_{\varepsilon 0}$ to solve halting problem, so not decidable.

Back to Hilbert

- Entscheidungsproblem posed by David Hilbert in 1928.
  - Find an algorithm that will take as input a description of a formal language and a mathematical statement in the language and produce as output either "True" or "False" according to whether the statement is true or false.
- If find an algorithm, then no problem, but ...
  - how do you show there is no such algorithm?
  - Turing' solution used undecidability of halting

Entscheidungsproblem

- Turing's solution:
  - First showed universal TM
  - Essentially showed undecidability of halting problem
    - Actually "circle-free" TM's
  - Showed undecidability of determining if ever write 0 on empty input
  - Can encode TM as a number (we did as string).
  - Showed given TM M, can write a logical formula ψ of predicate logic such that ψ is true iff M writes 0 on ε
  - Contradiction! Therefore not decidable

Entscheidungsproblem

- Even more on Turing's solution last step:
  - Given TM M, can write a logical formula ψ of predicate logic such that ψ is true iff M writes 0 on ε input.
    - Let ψ be statement: In some configuration of M starting with ε, some square s contains the symbol 0
    - Let $\phi_1,\ldots,\phi_n$ be axioms for M.
    - Then formula is $\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \psi$
  - Thus M writes 0 on input ε iff $\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \psi$ is provable in predicate calculus.
    - Thus if can decide provability then can decide if M writes 0 on ε
  - Therefore provability undecidable!