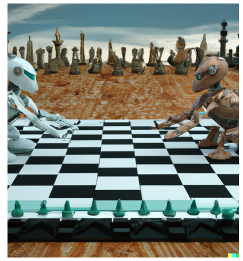
Lecture 14
February 5, 2024

CS21 Decidability and Tractability



1

Outline

- undecidable problems
  - computation histories
  - surprising contrasts between decidable/undecidable
- Rice's Theorem
- Post Correspondence Problem (skip)
- Beyond RE and co-RE
- Recursion Theorem

2

Dec. and undec. problems

- two problems regarding Context-Free Grammars:
  - does a CFG generate all strings:
    $ALL_{CFG} = \{\langle G\rangle : G \text{ is a CFG and } L(G) = \Sigma^*\}$
  - CFG emptiness:
    $E_{CFG} = \{\langle G\rangle : G \text{ is a CFG and } L(G) = \emptyset\}$
- Both decidable? both undecidable? one decidable?

3

Dec. and undec. problems

Theorem: E_{CFG} is decidable.

Proof:

- observation: for each nonterminal A, the set
  $S_A = \{w : A \Rightarrow^* w\}$
  is non-empty iff there is some rule:
  $A \rightarrow \mathbf{x}$
  and for all non-terminals B in string x, $S_B \neq \emptyset$

4

Dec. and undec. problems

Proof:

- on input <G>
- mark all terminals in G
- repeat until no new non-terminals get marked:
  - if there is a production $A \rightarrow x_1x_2x_3\ldots x_k$
  - and each symbol $x_1, x_2, \ldots, x_k$ has been marked
  - then mark A
- if S marked, reject ($G \notin E_{CFG}$), else accept ($G \in E_{CFG}$).
- terminates? correct?

5

Dec. and undec. problems

Theorem: ALL_{CFG} is undecidable.

Proof:

- reduce from co-A_{TM} (i.e. show co-$A_{TM} \leq_m ALL_{CFG}$)
- what should f(<M, w>) produce?
- Idea:
  - produce CFG G that generates all strings that are ***not*** accepting computation histories of M on w

6

Dec. and undec. problems

Proof:

- build a NPDA, then convert to CFG
- want to accept strings not of this form,

  $\#C_1\#C_2\#C_3\#...\#C_k\#$

  plus strings of this form but where
  - C_1 is not the start config. of M on input w, or
  - C_k is not an accept. config. of M on input w, or
  - C_i does not yield in one step C_{i+1} for some i

7

Dec. and undec. problems

Proof:

- our NPDA nondeterministically checks one of:
  - C_1 is not the start config. of M on input w, or
  - C_k is not an accept. config. of M on input w, or
  - C_i does not yield in one step C_{i+1} for some i
  - input has fewer than two #'s
- details of first two?
- to check third condition:
  - nondeterministically guess C_i starting position
  - how to check that C_i doesn't yield in 1 step C_{i+1} ?

8

Dec. and undec. problems

Proof:

- checking:
  - C_i does not yield in one step C_{i+1} for some i
- push C_i onto stack
- at #, start popping C_i and compare to C_{i+1}
  - accept if mismatch away from head location, or
  - symbols around head changed in a way inconsistent with M's transition function.
- is everything described possible with NPDA?

9

Dec. and undec. problems

Proof:

- Problem: cannot compare C_i to C_{i+1}
- could prove in same way that proved

  $\{ww: w \in \Sigma^*\}$ not context-free
- recall that

  $\{ww^R: w \in \Sigma^*\}$ is context-free
- free to tweak construction of G in the reduction
- solution: write computation history:

  $\#C_1\#C_2^R\#C_3\#C_4^R...\#C_k\#$

10

Dec. and undec. problems

Proof:

- f(<M, w>) = <G> equiv. to NPDA below:

on input x, accept if not of form: $\#C_1\#C_2^R\#C_3\#C_4^R...\#C_k\#$
• accept if C_1 is the not the start configuration for M on input w
• accept if check that C_i does not yield in one step C_{i+1}
• accept if C_k is not an accepting configuration for M

- is f computable?
- YES maps to YES?

  <M, w> $\in$ co-A_{TM} $\Rightarrow$ f(M, w) $\in$ ALL_{CFG}
- NO maps to NO?

  <M, w> $\notin$ co-A_{TM} $\Rightarrow$ f(M, w) $\notin$ ALL_{CFG}

11

Rice's Theorem

- We have seen that the following properties of TM's are undecidable:
  - TM accepts string w
  - TM halts on input w
  - TM accepts the empty language
  - TM accepts a regular language
- Can we describe a single generic reduction for all these proofs?
- Yes. *Every* property of TMs undecidable!

12

Rice's Theorem

- A TM property is a language P for which
  - if $L(M_1) = L(M_2)$ then $<M_1> \in P$ iff $<M_2> \in P$
- TM property P is nontrivial if
  - there exists a TM M_1 for which $<M_1> \in P$, and
  - there exists a TM M_2 for which $<M_2> \notin P$.

Rice's Theorem: Every nontrivial TM property is undecidable.

13

Rice's Theorem

- The setup:
  - let $T_\varnothing$ be a TM for which $L(T_\varnothing) = \varnothing$
    - technicality: if $<T_\varnothing> \in P$ then work with property co-P instead of P.
    - conclude co-P undecidable; therefore P undec. due to closure under complement
  - so, WLOG, assume $<T_\varnothing> \notin P$
  - non-triviality ensures existence of TM M_1 such that $<M_1> \in P$

14

Rice's Theorem

Proof:
- reduce from A_{TM} (i.e. show $A_{TM} \leq_m P$)
- what should f(<M, w>) produce?
- f(<M, w>) = <M'> described below:

> on input x,
>
> • accept iff M accepts w and M_1 accepts x
>
> (intersection of two RE languages)

- f computable?
- YES maps to YES?

  $<M, w> \in A_{TM} \Rightarrow$
  $L(f(M, w)) = L(M_1) \Rightarrow$
  $f(M, w) \in P$

15

Rice's Theorem

Proof:
- reduce from A_{TM} (i.e. show $A_{TM} \leq_m P$)
- what should f(<M, w>) produce?
- f(<M, w>) = <M'> described below:

> on input x,
>
> • accept iff M accepts w and M_1 accepts x
>
> (intersection of two RE languages)

- NO maps to NO?

  $<M, w> \notin A_{TM} \Rightarrow$
  $L(f(M, w)) = L(T_\varnothing) \Rightarrow$
  $f(M, w) \notin P$

16