CLARIFICATIONS AND NOTES

PART 1

Abhijit Das

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

April 24, 2020

Undecidable Problems: A Note

That a problem Π is undecidable does not mean that all instances of Π are undecidable.

- A decider may exist for some instances of Π.
- HP is proved to be undecidable by a diagonalization argument.
- Restrict the instance $M \# w$ for HP to those TMs M which do **not** have transitions of the form $\delta(p,a) = (q,b,L)$.
- The heads of these machines always move right.
- Given such an M, and any input w for M, it is decidable whether M halts on w.
- A limited-time simulation of M on w can decide this.

A Specific Example

Consider a TM K with the specifications:

- $Q = \{s, t, r\}$.
- $\Sigma = \{0, 1\}$.
- $\Gamma = \{\triangleright, 0, 1, \square\}$.
- Transition function:

$$\begin{aligned} \delta(s, \triangleright) &= (s, \triangleright, R), \\ \delta(s, 0) &= (t, 0, R), \\ \delta(s, 1) &= (r, 1, R), \\ \delta(s, \square) &= (r, \square, R). \end{aligned}$$

- $\mathscr{L}(K) = \{w \in \{0, 1\}^* \mid w \text{ starts with } 0\}$.
- K halts on all inputs in two steps.

The General Case

- M is a TM with no transitions with left head movement.
- w is an input for M.
- Run M for $|w| + 1 + |Q|$ steps.
- If M halts by that time, accept.
- If not, reject.
  - After the first $|w| + 1$ steps, the head of M leaves the input.
  - Subsequently, the head can only scan blanks.
  - If M runs for another $|Q|$ iterations, some state must be repeated.
  - M has started looping, and will not halt.

But Then

Can a reduction HP $\leqslant_m \Pi$ map all instances of HP to only easy instances of Π?

- **No.**
- The easy instances of Π have a decider D.
- The reduction followed by running D decides HP.
- HP is already proved to be undecidable by diagonalization.

- Although reductions are many-to-one maps, using them is perfectly legitimate to prove the undecidability of new problems.

A Fact about CFLs

It is undecidable whether the complement of a CFL L is again a CFL.

- Use the reduction that maps $M \# w$ to $L = \overline{\mathrm{VALCOMP}(M, w)}$.
- If M does not halt on w:
  - There are no valid computation histories.
  - $\mathrm{VALCOMP}(M, w) = \emptyset$.
  - $L = \overline{\mathrm{VALCOMP}(M, w)} = \Delta^*$.
  - $\overline{L} = \emptyset$ is a CFL.
- If M halts on w:
  - There are valid computation histories.
  - $\overline{L} = \mathrm{VALCOMP}(M, w) \neq \emptyset$ should be a non-CFL.
- **But, a non-empty VALCOMP(M, w) may be a CFL, even regular.**

More about a Non-Empty VALCOMP(M, w)

- VALCOMP$(M,w) \neq \emptyset$ must be infinite.
- It is infinite for two reasons:
  - We are allowed to repeat the halting configuration (at the end) as many times as we want.
  - In each configuration, we can append as many blank symbols ($\square$ to be more precise) as we want.
- This alone does not prove that VALCOMP(M,w) is not a CFL.

A Regular VALCOMP(M, w)

- Consider the machine K with
  - $Q = \{s, t, r\}$.
  - $\Sigma = \{0, 1\}$, and $\Gamma = \{\triangleright, 0, 1, \square\}$.
  - $\delta(s, \triangleright) = (s, \triangleright, R)$, $\delta(s, 0) = (t, 0, R)$,
    $\delta(s, 1) = (r, 1, R)$, $\delta(s, \square) = (r, \square, R)$.
- Let the input to K be 0.
- Consider the regular expressions:
  - $C_0 = \begin{matrix} \triangleright & 0 \\ s & - \end{matrix} \begin{pmatrix} \square \\ - \end{pmatrix}^*.$
  - $C_1 = \begin{matrix} \triangleright & 0 \\ - & s \end{matrix} \begin{pmatrix} \square \\ - \end{pmatrix}^*.$
  - $C_2 = \begin{matrix} \triangleright & 0 & \square \\ - & - & t \end{matrix} \begin{pmatrix} \square \\ - \end{pmatrix}^*.$
- $\text{VALCOMP}(K, 0) = \mathscr{L}\Big(\# \, C_0 \, \# \, C_1 \, \# \, C_2 \, \# \, (C_2 \, \# \,)^*\Big).$

A Generic Construction

- **Input:** $M \# w$.
- **Output:** $N \# w$.
  - N is a nondeterministic Turing machine.
  - $\mathscr{L}(N) = \mathscr{L}(M)$.
  - M has valid computation histories on $w \iff N$ has valid computation histories on w.
- The conversion:
  - Mark the accept state t and the reject state r of M as non-halting.
  - Add a new accept state t' and a new reject state r'.
  - Add a new tape symbol $\blacksquare$.
  - Add the transitions $\delta(t, \rhd) = \{(t, \rhd, R)\}$, $\delta(r, \rhd) = \{(r, \rhd, R)\}$,

$$\delta(t, *) = \{(t, \blacksquare, R), (t', \blacksquare, R)\} \quad \text{and} \quad \delta(r, *) = \{(r, \blacksquare, R), (r', \blacksquare, R)\}.$$

- VALCOMP(N, w) is **not** a CFL irrespective of VALCOMP(M, w).

A Strong Variant of Ogden's Lemma

- Let L be a CFL.
- There exists a constant k.
- Take any $z \in L$ such that
  - z has d **distinguished** positions,
  - z has e **excluded** positions,
  - $d \geqslant k(e+1)$.
- Then,

  $$z = uvwxy$$

  such that

  1. vwx contains at most d distinguished positions,
  2. vx contains at least one distinguished position,
  3. vx contains **no** excluded positions,
  4. $z_i = uv^i wx^i z \in L$ for all $i \geqslant 0$.

A Non-Empty VALCOMP(N,w) is not Context-Free

- Take a sufficiently long valid computation history

  $\# \, C_0 \, \# \, C_1 \, \# \, C_2 \, \# \ldots \# \, C_L \, \#.$

- No configuration with trailing blanks (unless the head is there).
- Mark the #'s as excluded positions.
- Mark all other positions as distinguished.
- L is chosen such that $d \geqslant k(e+1)$ is satisfied.
- vx does not contain any #.
- v or x, whichever is non-empty, must be inside a single configuration.
- Pump in or pump out vx once.
- (At least) once inconsistency is introduced.
- $z_0, z_2 \notin L$, a contradiction.

$\overline{\text{VALCOMP}(M,w)} \neq \Delta^*$ is not a DCFL

- A string $\alpha = \#C_0\#C_1\#C_2\#\ldots\#C_N\# \in \Delta^*$ may be in $\overline{\text{VALCOMP}(M,w)}$ for many reasons *simultaneously*.
- Syntactic reasons
  - C_0 is not the start configuration.
  - Some configurations contain multiple (or no) states.
  - C_N is not a halting configuration.
- Semantic reasons
  - Multiple inconsistencies among consecutive pairs of configurations.
- α may have multiple parse trees.
- Any grammar for $\overline{\text{VALCOMP}(M,w)} \neq \Delta^*$ is ambiguous.
- **A DCFL is inherently unambiguous.**