High-level TM descriptions and undecidable languages

1. Infinities

Recall the following about infinities:

- A set is *countably infinite* if there is a bijection between that set and the natural numbers.
- The set of strings over any alphabet is countably infinite. The easiest way to show that any *other* set is countable is often to show that each element of the set has a finite representation (i.e. in some alphabet, every object in the set has a string that describes it).
- Every set is strictly smaller than its powerset. In particular, the powerset of any countably infinite set is uncountable.

In this section, let Σ be some fixed alphabet.

a. First, explain why technically there are uncountably many TMs with alphabet Σ. However, in an informal sense, many of these TMs are not "truly distinct" - not only do many of them recognize/decide the same language but they do so "in exactly the same way". How would you formalize this notion: in what circumstances should we consider two TMs to be truly distinct, vs in what circumstances are they effectively identical?

b. Why is the number of "truly distinct" TMs countably infinite?

c. Show that there are uncountably many languages over Σ, and as a corollary there must exist undecidable languages and also unrecognizable languages.

2. High-level descriptions of TMs

a. Consider the following TM:

$M =$ "On input $\langle B \rangle$, where B is a DFA:

- If the string is *not* actually the encoding of a DFA, *reject*.[1]
- (Using standard graph algorithms,) check if there is a path in the transition graph from B's start state to any accept state. If there is, *reject*; otherwise, *accept*."

What language does M decide?

[1] We will omit this step in the future; you should assume that any types in the "On input..." line always implicitly include a type-checking step like this one.

b. Which of the following do you think we should allow as part of a high-level description of a Turing Machine? (Assume that all variables were defined before use, and all capital-letter variables are TMs unless otherwise specified.)

i "On input $\langle M, w\rangle$, where M is a TM and w is a string: $\cdots$ "

ii "On input M, where M is a TM: $\cdots$ "

iii "On input $\langle L\rangle$, where L is a language: $\cdots$ "

iv "On input $\langle M\rangle$, where M is a TM: If the first symbol of $\langle M\rangle$ is a 0, $\cdots$ "

v "$\cdots$ If (graph) G has at least 5 cycles, $\cdots$ "

vi "$\cdots$ If M is a decider, $\cdots$ "

vii "$\cdots$ Loop forever. Then, $\cdots$ "

viii "$\cdots$ Run M on input w. If it accepted, $\cdots$; otherwise, $\cdots$ "

ix "$\cdots$ Run M_1 and M_2 in parallel $\cdots$ "

x "$\cdots$ Run T on input $\langle M, \langle M\rangle\rangle$. $\cdots$ " (and does this differ from running on $\langle M, M\rangle$?)

xi "$\cdots$ Run M on input $\langle M\rangle$. $\cdots$ "

3. Undecidable and unrecognizable languages

a Let H be some fixed decider, and consider the following TM D:

$D =$ "On input $\langle M \rangle$, where M is a TM:

1. Run H on input $\langle M, \langle M \rangle \rangle$.
2. Output the opposite of what H outputs. That is, if H accepts, *reject*; and if H rejects, *accept*."

Is D a decider? (In terms of H,) what does D return on input $\langle D \rangle$?

b Prove $A_{TM} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ is undecidable. *(Hint: assume towards contradiction it's decidable, and its decider is H.)*

c Prove A_{TM} is recognizable.

d Prove a language L is decidable if and only if L and $\overline{L}$ are recognizable. (Corollary: $\overline{A_{TM}}$ is unrecognizable.)