1 Turing Machine Extensions; Nondeterminism

Section 4.3 of the text shows that Turing machines are equivalent in computing power (disregarding time and space considerations) to many extensions, such as multiple tapes, a two-way infinite tape, two dimensional tapes, and multiple read-write heads. Section 4.4 shows that random access Turing machines are also equivalent in power. The fact that all these extensions do not add any power is evidence that Turing machines are a maximally powerful model of computing.

- We know that finite automata are not universal models of computation because push-down automata are plausible computing machines and are more powerful.
- We know that push-down automata are not universal models of computation because Turing machines are plausible models of computing machines and are more powerful than push-down automata.
- But despite many extensions of Turing machines that have been considered, no one has found any plausible computing device that is more powerful than a Turing machine.
- This justifies the belief that a Turing machine is a fully general model of computability.

We will spend some time in class on section 4.5, discussing nondeterministic Turing machines. These are also equivalent in power to the standard Turing machines presented in the text, but nondeterminism in Turing machines is important for other reasons, because of its connection to the P versus NP problem.

1.1 Nondeterminism

Definition 1.1 (Nondeterministic Turing machine) *A* nondeterministic Turing machine *is a quintuple* $(K, \Sigma, \Delta, s, H)$ *where* $K, \Sigma, s,$ *and* H *are as for standard Turing machines and* Δ *is a subset of*

$$((K - H) \times \Sigma) \times (K \times (\Sigma \cup \{\leftarrow, \rightarrow\}))$$

rather than a function from $((K - H) \times \Sigma)$ *to* $(K \times (\Sigma \cup \{\leftarrow, \rightarrow\}))$.

Configurations are defined as for deterministic Turing machines. The relations $\vdash_M$ and $\vdash_M^*$ are defined in the natural way, but now there may be more than one configuration C_2 such that $C_1 \vdash C_2$, or there may be none at all.

The text defines acceptance for nondeterministic Turing machines in a different way than for deterministic Turing machines, unfortunately.

Definition 1.2 (4.5.1, page 222, Nondeterministic Acceptance, Deciding) *Let $M = (K, \Sigma, \Delta, s, H)$ be a nondeterministic Turing machine. Then M* accepts *an input $w \in (\Sigma - \{\triangleright, \sqcup\})^*$ if $(s, \triangleright\underline{\sqcup}w) \vdash_M^* C$ for some halting configuration C. M* semidecides *a language $L \subseteq (\Sigma - \{\triangleright, \sqcup\})^*$ if for all $w \in (\Sigma - \{\triangleright, \sqcup\})^*$, $w \in L$ iff M accepts w.*

Even if M accepts an input, there may be some computations that never halt.

The definition of deciding a language, or computing a function, are based on Turing machines with two halting states, y and n.

Deciding a language:

1. All computations on input $w \in (\Sigma - \{\triangleright, \sqcup\})$ must eventually halt.
2. $w \in L$ if and only if some computation halts in state y. Other computations may halt in state n.

This definition is relevant for the P versus NP problem because it permits computation b the "guess and verify" approach – to test if a graph can be 4 colored, for example, just guess a coloring and see if it is valid.

Computing a function:

1. All computations on input $w \in (\Sigma - \{\triangleright, \sqcup\})$ must eventually halt.
2. All computations on input w must halt with $f(w)$ as output.

Like all the other extensions, nondeterminism does not add any power to Turing machines, though it may make some computations a lot faster.

Theorem 1.1 (4.5.1, Nondeterminism does not add power) *If a nondeterministic Turing machine M semidecides or decides a language or computes a function, there is a standard Turing machine M' that semidecides or decides the same language or computes the same function.*

The details of the simulation of a nondeterministic Turing machine by a deterministic one, are given in the text.