CSE491/596, Fri. 9/25/20.   Turing Machines

We saw that DFAs M, nor even NFAs nor GNFAs, cannot recognize simple languages like $\left\{a^m\, b^n : \; m = n\right\}$. How can we augment the DFA *model* to give it the needed capability?

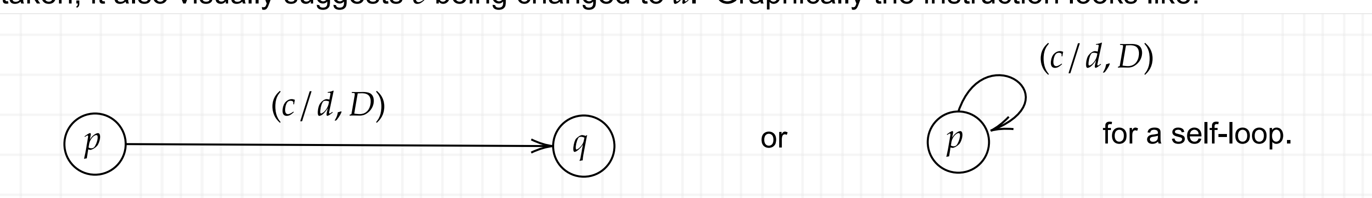
1. Allow M to change a character it reads, storing it on its tape.
2. Allow M to move its scanner left L as well as right R (or keep it stationary S).

Capability 1 by itself changes nothing: the DFA would still have to move R past the changed character. Capability 2 by itself also does not allow recognizing any nonregular languages. The proof, that every "two-way DFA" can be simulated by a simple 1-way DFA, is beyond our scope and involves another "exponential explosion" but we will cite it later to say that the class of regular languages equals "constant space" on a Turing machine.

But if we give both capabilities together, then we can do it---and lots more besides. The capabilities add two components to instructions in δ, making them 5-tuples:

$$(p, c / d, D, q) \quad \text{where } p \text{ and } q \text{ are states, } c \text{ and } d \text{ are chars, and } D \in \{\mathsf{L}, \mathsf{R}, \mathsf{S}\}$$

The meaning is that if M is in state p and scans character c, then it can change it to d, move its scanning head one position left, right, or keep it stationary, and finally transit to state q. The case (p, c, c, R, q) is the same as an ordinary FA instruction (p, c, q) where moving right is automatic. I tend to like to write a slash for the second comma to emphasize that p, c are read and d, D, q are actions taken; it also visually suggests c being changed to d. Graphically the instruction looks like:



We also regard the blank as an explicit character. I will represent it as _ in MathCha but in full LaTeX you can get "\text{\textvisiblespace}" which turns up the corners to look like more than just an underscore. My other notes call the blank B. The blank belongs not to the *input alphabet* Σ but to the work alphabet Γ (capital Gamma) which always includes Σ too. We allow going past the right end of the input string $x \in \Sigma^*$ where successive *tape cells* each initially hold the blank. We *can* also allow moving leftward of the first char of x where there are likewise blanks on a "two-way infinite tape", *or* we can stipulate that x is initially left-justified on a "one-way infinite tape" and consider any left move from the first cell to be a "crash." The *Turing Kit* package shows a two-way infinite tape and this is the default. A compromise is to use a one-way infinite tape but place a special left-endmarker char $\wedge$ in cell 0 with x occupying cells $1, \ldots, n$ where $n = |x|$. If $x = \epsilon$ then the whole tape is initially blank except in the last case it has just $\wedge$ in cell 0. Then $\wedge$, as well as _, belongs to Γ but not to Σ. We will be free to put any other characters we want into Γ, but the blank (and $\wedge$ if used) are required. With all that said, the definition is crisp:

Definition: A k-*tape Turing machine* is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \_, s, F)$ where Q, s, F and Σ are as with a DFA, the *work alphabet* Γ includes Σ and the *blank* _, and

$$\delta \subseteq \left(Q \times \Gamma\right) \times \left(\Gamma \times \{\mathsf{L}, \mathsf{R}, \mathsf{S}\} \times Q\right) .$$

It is *deterministic* (a DTM) if no two instructions share the same first two components. A DTM is "in normal form" if F consists of one state q_{acc} and there is only one other state q_{rej} in which it can halt, so that δ is a function from $(Q \setminus \{q_{acc}, q_{rej}\}) \times \Gamma$ to $(\Gamma \times \{\mathsf{L}, \mathsf{R}, \mathsf{S}\} \times Q)$. The notation then becomes $M = (Q, \Sigma, \Gamma, \delta, \_, s, q_{acc}, q_{rej})$.

To define the language $L(M)$ formally, especially when M is properly nondeterministic (an NTM), requires defining *configurations* (also called *ID*s for *instantaneous descriptions*) and *computations,* but especially with DTMs we can use the informal understanding that $L(M)$ is the set of input strings that cause M to end up in q_{acc}, while seeing some examples first.

$L = \left\{a^m b^n : n = m\right\}$, by default allowances, $\epsilon \in L$ by $n = m = 0$ being allowed.