Universal Turing machines

Assumptions about TMs

- **Assumption #1:** All languages are over the alphabet $\{0, 1\}$, and all TMs have input alphabet $\{0, 1\}$
- **Theorem:** For every language L over an alphabet Σ, there exists a language $L_{0,1} \subseteq \{0, 1\}^*$ s.t.
  1. L is recognizable iff $L_{0,1}$ is recognizable
  2. L is decidable iff $L_{0,1}$ is decidable
- Proof: the idea is to use a fixed-length binary encoding of the symbols in Σ, i.e. represent each symbol in $x \in \Sigma$ by a string in $\phi(x) \in \{0, 1\}^*$. The encoding function ϕ should be $1 - 1$, and satisfy that $|\phi(x)| = k$, for all $x \in \Sigma$. The details of the proof are omitted, but are very similar to the proof of the next theorem.
- **Definition:** For an "object" Z, we will use $\langle Z \rangle$ to denote some representation of Z over the alphabet $\{0, 1\}$. For several objects $Z_1, \ldots, Z_k$ we will write $\langle Z_1, \ldots, Z_k \rangle$ to denote a representation over $\{0, 1\}$ of all of the objects. For example, if G is a graph we may let $\langle G \rangle$ denote the adjacency matrix of G, i.e. a list of n^2 entries in $\{0, 1\}$ such that the entry $jn + i$ ($i, j \in \{0, \ldots, n - 1\}$) is 1 iff (i, j) is an edge of G.
- **Assumption #2:** All TMs have tape alphabet $\{0, 1, \sqcup\}$.
- **Theorem:** TMs with tape alphabet $\{0, 1, \sqcup\}$ recognize/decide the same class of languages as ordinary TMs (with arbitrary tape alphabets)
- Proof: again, the idea is to use a fixed-length binary encoding of Γ. Let M be a TM with arbitrary tape alphabet Γ. Pick an encoding function $\phi : \Gamma \mapsto \{0, 1\}^*$ with the following properties:
  1. ϕ is $1 - 1$, i.e. $\forall x \neq y \in \Gamma$, $\phi(x) \neq \phi(y)$
  2. For some $k \in \mathbb{N}$, $|\phi(x)| = k$ for all $x \in \Gamma$ (i.e. ϕ is a fixed-length encoding)
  3. $\phi(\sqcup) = \sqcup^k$ (i.e. $\sqcup$ is encoded by a string of k $\sqcup$'s)
- Define a TM M' which does the following on input $w \in \{0, 1\}^*$:
  - Replace each input symbol w_i with $\phi(w_i)$, so that the tape contains $\phi(w_1)\phi(w_2) \cdots \phi(w_n)$
  - Notice that since we defined $\phi(\sqcup) = \sqcup^k$, the entire tape of M' is now an encoding of the entire tape of M
  - Move to the left-most non-blank symbol
  - Initially, remember the state q_0 of M as part of our state

  - To simulate a step of M, read k symbols to the right starting from the current symbol and remember them in the state. If the symbols are $x_1, \ldots, x_k$, and q is the state of M that is currently remembered, and $y = \phi^{-1}(x_1 \cdots x_k)$, and e.g. $\delta_M(q, y) = (q', z, D)$, then write $\phi(z)$ over the symbols $x_1, \ldots, x_k$ that were just read and change the remembered state to q'. Finally, move to the cell that is k spaces in direction D from the cell we were scanning at the beginning of this step. Accept if $q' = q_{accept}$, reject if $q' = q_{reject}$, otherwise continue the simulation.

Representing TMs

- **Proposition:** A Turing machine M can be represented by a string over $\{0, 1\}$
- There are many ways to do this. Here is one way. Let $M = (Q, \{0,1\}, \{0,1,\sqcup\}, q_0, q_{accept}, q_{reject}, \delta)$ be a TM. Assign a unique number $\#(q)$ to each state $q \in Q$, such that
  - $\#(q_0) = 1$
  - $\#(q_{accept}) = 2$
  - $\#(q_{reject}) = 3$
- Similarly, we will assign a number to each alphabet symbol:
  - $\#(0) = 1$
  - $\#(1) = 2$
  - $\#(\sqcup) = 3$
- Finally, assign a number to each direction:
  - $\#(L) = 1$
  - $\#(R) = 2$
- Now we will represent a transition $\delta(q, x) = (q', y, D)$ by the string

$$0^{\#(q)}10^{\#(x)}10^{\#(q')}10^{\#(y)}10^{\#(D)}$$

- And we will represent the TM M by a string consisting of the representation of each transition (in any order), separated by the string 11 and with 11 preceding the first transition and following the last transition.

Example

STATE	SYMBOL	STATE	SYMBOL	DIRECTION
q_0	0	q_1	1	R
q_0	1	q_0	1	L
q_0	$\sqcup$	q_{accept}	$\sqcup$	R
q_1	0	q_{reject}	0	R
q_1	1	q_{reject}	1	R
q_1	$\sqcup$	q_0	0	L

- We need to choose $\#(q_1)$. It doesn't matter as long as it's not $1, 2$, or 3, so choose $\#(q_1) = 4$.
- The first transition, $\delta(q_0, 0) = (q_1, 1, R)$, is represented by the string

  01010000100100

- Similarly, the representations of the other transitions (in order), are:

  01001010010
  010001001000100
  000010100010100
  00001001000100100
  00001000101010

- So, the representation $\langle M \rangle$ of M is:

  1101010000100100110100101001011010001001000100011000
  0101000101001100001001000100100110000100010101011

A Universal Turing machine

- **Theorem:** There exists a "universal" Turing machine U that, on input $\langle M, w \rangle$ where M is a TM and $w \in \{0, 1\}^*$, simulates the computation of M on input w. Specifically:
  1. U accepts $\langle M, w \rangle$ iff M accepts w
  2. U rejects $\langle M, w \rangle$ iff M rejects w
- Proof: Define a TM U with three tapes.
  - The first tape will hold a copy of $\langle M \rangle$
  - The second tape is the "simulation" tape; it corresponds exactly to M's tape

- The third tape contains $0^{\#(q)}$, where q is the current state of M

- Initially, copy w to the second tape (and erase it from the first tape), and write a single 0 on the third tape. Reset all tape heads to the left-most non-blank symbol on each tape.

- To simulate a step of M, move through the description of M, and for each transition $0^{\#(q)}10^{\#(x)}10^{\#(q')}10^{\#(y)}10^{\#(D)}$ do the following:
  - Compare $0^{\#(q)}$ with the contents of the first tape
  - If it matches, check if the symbol scanned by the second tape-head is x
  - If so, replace it with y, move the second tape-head one cell in direction D, and write $0^{\#(q')}$ onto the third tape (erasing the previous contents)
  - If the third tape ever contents 00 (remember: $00 = \#(q_{accept})$, then accept; if it ever contains 000 ($= \#(q_{reject})$) then reject.