Encoding Turing machines

Encoding of arbitrary TM $M = (Q, \Sigma, \Gamma, \delta, q_0, h_A, h_R)$ as a string over {0,1,#}.

- Encode each $q \in Q$ by a unique binary string $\langle q \rangle$ of length $\lceil \log_2 |Q| \rceil$.
- Encode $Q = \{q_1, q_2, \ldots, q_n\}$ as the string $\langle Q \rangle = \langle q_1 \rangle \# \langle q_2 \rangle \# \ldots \# \langle q_n \rangle$ with the convention that
  - $\langle q_1 \rangle$ is the encoding of the initial state
  - $\langle q_2 \rangle$ is the encoding of the accept state
  - $\langle q_3 \rangle$ is the encoding of the reject state.

Encoding Turing machines

- Encode each $a \in \Gamma$ by a unique binary string $\langle a \rangle$ of length $\lceil \log_2 |\Gamma| \rceil$.
- Encode $\Sigma = \{a_1, a, \dots, a_m\}$ as the string $\langle \Sigma \rangle = \langle a_1 \rangle \# \langle a_2 \rangle \# \dots \# \langle a_m \rangle$.
- Encode Γ similarly with the convention that the encoding of ⊔ is listed first; the encoding of Γ is denoted $\langle \Gamma \rangle$.
- Encode string $x = a_1 a_2 \dots a_k \in \Gamma^*$as the string $\langle x \rangle = \langle a_1 \rangle \langle a_2 \rangle \dots \langle a_k \rangle$.
  (Since we have fixed-length codes for the symbols in Γ we don't need separators.)

Encoding Turing machines

- Encode the state transition $\delta(q, a) = (p, b, D)$ as the string $\langle q\rangle\langle a\rangle\langle p\rangle\langle b\rangle\langle D\rangle$, where $\langle D\rangle = 0$ if $D = L$ and $\langle D\rangle = 1$ if $D = R$.
- Encode the entire state transition function as $\langle\delta\rangle = \langle\mathrm{trans}_1\rangle\langle\mathrm{trans}_2\rangle \ldots \langle\mathrm{trans}_\ell\rangle$.
- Encode the entire TM M as
$$\langle M\rangle = \langle Q\rangle\#\#\langle\Sigma\rangle\#\#\langle\Gamma\rangle\#\#\langle\delta\rangle$$

Encoding Turing machines

<u>Conventions</u>:

Strings in $\{0,1,\#\}^*$ that do not encode a TM according to the above conventions are assumed to encode a fixed TM, say one that rejects all strings in zero moves (i.e., initial state = reject state).

Strings in $\{0,1,\#\}^*$ that do not encode a string in Γ^* according to the above conventions are assumed to encode a fixed string, say ε (empty string).

Encoding mathematical objects

Fix any alphabet sufficient to encode Turing machines (e.g., $\{0,1,\#\}$ or even just $\{0,1\}$).

- For any finite mathematical object O (natural number, set, sequence, graph, ...) the encoding of O is denoted $\langle O \rangle$.
- The encoding of a sequence $O_1, O_2, \ldots, O_k$ of finite mathematical objects is denoted $\langle O_1, O_2, \ldots, O_k \rangle$. E.g., the encoding of the pair consisting of a Turing machine M and an input x is denoted $\langle M, x \rangle$.

Encoding mathematical objects

- The details of how we encode different (finite) mathematical objects are not important, but we assume that:
  - Given $\langle O_1 \rangle, \langle O_2 \rangle, \ldots, \langle O_k \rangle$, some TM constructs $\langle O_1, O_2, \ldots, O_k \rangle$.
  - Given $\langle O_1, O_2, \ldots, O_k \rangle$, for each $i \in 1..k$, some TM constructs $\langle O_i \rangle$.

  That is, encodings and decodings are computable functions.

Universal Turing machine

A Turing machine M_U that, given $\langle M, x\rangle$, simulates the computation of machine M on input x.

$$\mathcal{L}(M_U) = \{\langle M, x\rangle : \text{Turing machine } M \text{ accepts input } x\}$$

- Tape 1: Input tape --- always contains input $\langle M, x\rangle$
- Tape 2: Contains (encoding of) M's tape; initially blank
- Tape 3: Contains (encoding of) M's state; initially blank

Universal Turing machine

1. From tape 1 copy $\langle q_0 \rangle$ (found in $\langle M \rangle$) to tape 3 and copy $\langle x \rangle$ to tape 2. Move the head of tape 2 to the leftmost cell.
2. If the (encoded) state of M on tape 3 is $\langle h_A \rangle$, accept; if it is $\langle h_R \rangle$, reject.
3. Determine the encoding $\langle a \rangle$ of the symbol of M that starts at the present position of the head on tape 2, and the encoding $\langle q \rangle$ of the current state of M on tape 3. On tape 1 find the encoding $\langle q \rangle \langle a \rangle \langle p \rangle \langle b \rangle \langle D \rangle$ of the state transition that starts with $\langle q \rangle \langle a \rangle$.

Universal Turing machine

4. Simulate the execution of this transition:
   - on tape 3 change the encoding of M's state from $\langle q \rangle$ to $\langle p \rangle$
   - on tape 2 change the encoding of M's symbol that starts at the present position of the tape 2 head from $\langle a \rangle$ to $\langle b \rangle$
   - position the head of tape 2 to the beginning of the encoding of the next or previous symbol of M, depending on $\langle D \rangle$

5. Go to step 2

Universal Turing machine

1. From tape 1 copy $\langle q_0 \rangle$ (found in $\langle M \rangle$) to tape 3 and copy $\langle x \rangle$ to tape 2. Position the head of tape 2 on the leftmost cell.
2. If the (encoded) state of M on tape 3 is $\langle h_A \rangle$, accept; if it is $\langle h_R \rangle$, reject.
3. Determine the encoding $\langle a \rangle$ of the symbol of M that starts at the present position of the head on tape 2, and the encoding $\langle q \rangle$ of the current state of M on tape 3. On tape 1 find the encoding $\langle q \rangle \langle a \rangle \langle p \rangle \langle b \rangle \langle D \rangle$ of the state transition that starts with $\langle q \rangle \langle a \rangle$.