Computability

June 27, 2021

Contents

1 Summary

In computability we consider two main classes

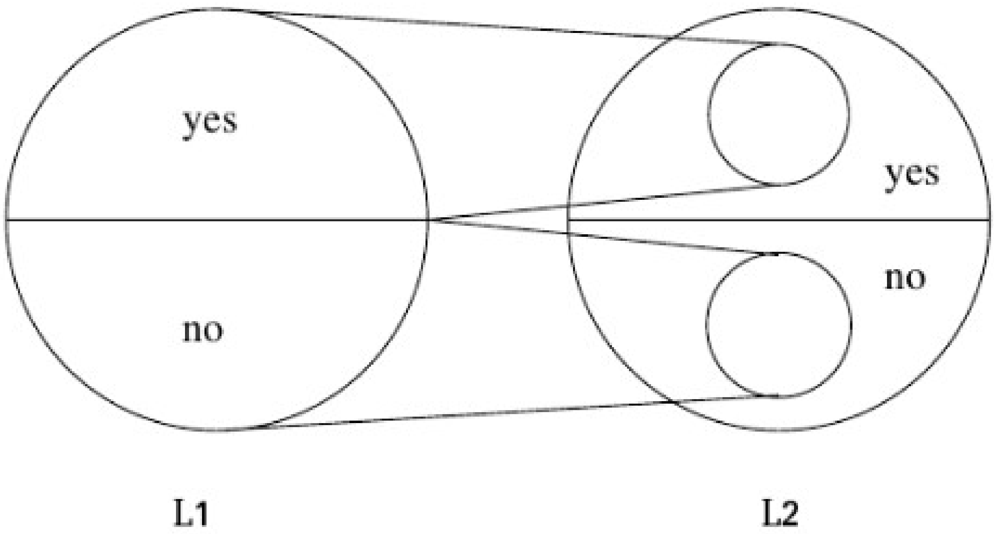
- **RE**, collecting the *recursively enumerable* languages, meaning languages accepted by some TM.
- **R**, collecting all the *recursive* (or *decidable*) languages.

Obviously $R \subseteq RE$. Usually we are interested into proving the decidability of a given language L, and to do so we can rely on few theorems.

- $L \in R \Rightarrow \bar{L} \in R$, where language $\bar{L}$ is the complement of L.
- $L, \bar{L} \in RE \Rightarrow L \in R$, and thus $\bar{L} \in \mathbf{R}$.

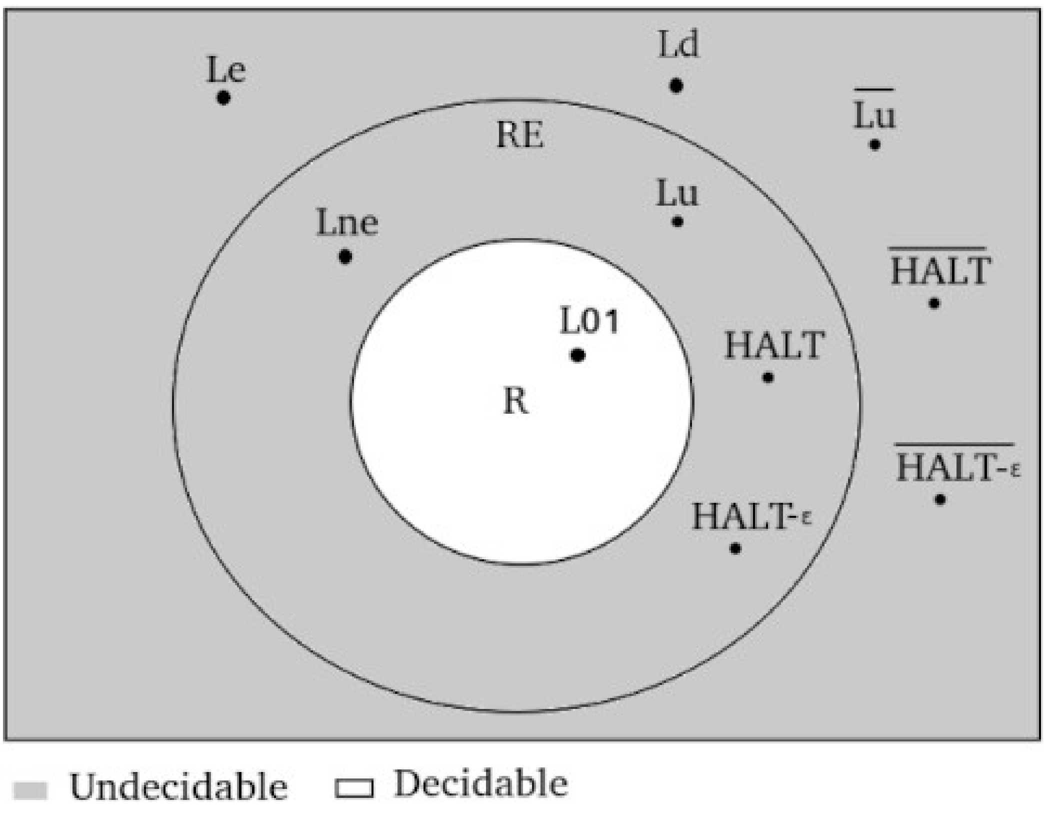
1.1 Reductions

Another tool we can rely on are *many-one reductions*: if $L_1 \leq L_2$ (L_1 reduces to L_2) then we can convert yes-instances of L_1 to yes-instances of L_2 and no-instances of L_1 to no-instances of L_1.



Reductions allow to transfer different properties between languages. If $L1_{\leq}L_2$ and

- $L1 \notin R \Rightarrow L_2 \notin R$, and $L2 \in R \Rightarrow L_1 \in R$.
- $L1 \notin RE \Rightarrow L_2 \notin RE$, and $L2 \in RE \Rightarrow L_1 \in RE$.



A list of some relevant examples of decidable and undecidable languages.

- $L_{01} = \{w | w = 0^n 1^n,\ n \geq 0\}$.
- $L_u = \{(M, w) | M$ accepts $w\}$, the *universal language*.
- $L_d = \{w_i | M_i$ does not accept $w_i\}$, the *diagonal language* (where w_i represents a TM M_i).

- $HALT = \{(M, w)|M$ halts with input $w\}$.
- $HALT - \epsilon = \{M|M$ halts with input $\epsilon\}$.
- $L_e = \{< M > |L(M) = 0\}$.

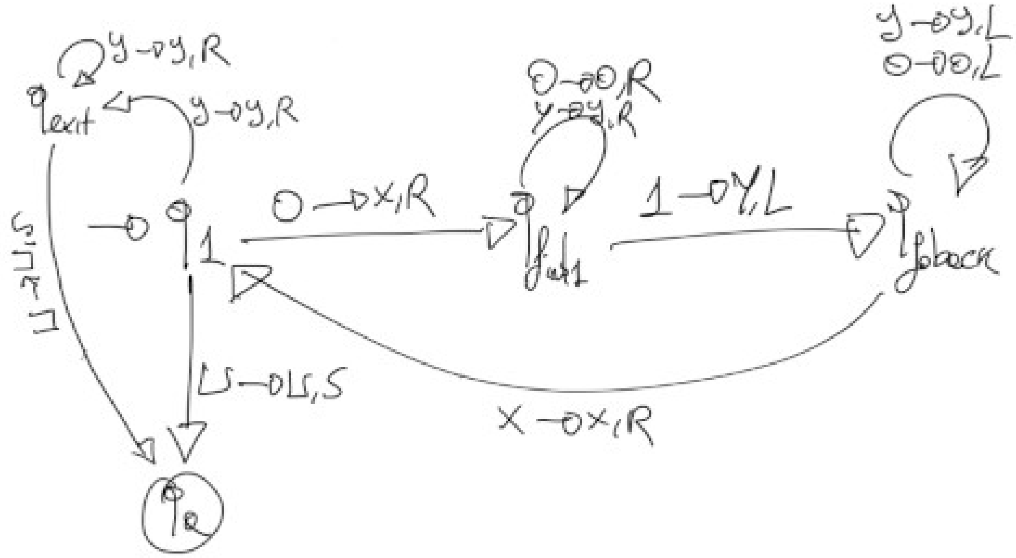
1.2 Rice's theorem

A *property* P is a language that collects all the (encoding of) TMs sharing a common characteristic: the *Rice's Theorem* states that if P is non-trivial and semantic then $P \notin R$.

- P is *non-trivial* if it is neither empty or the set of encodings fo all possible TMs.
- P is *semantic* if, having two TMs M_1, M_2 such that $L(M_1) = L(M_2)$, then either $< M_1 >, < M_2 > \in P$ or $< M_1 >, < M_2 > \notin P$.

2 Exercises on TMs

2.1 Exercise 1



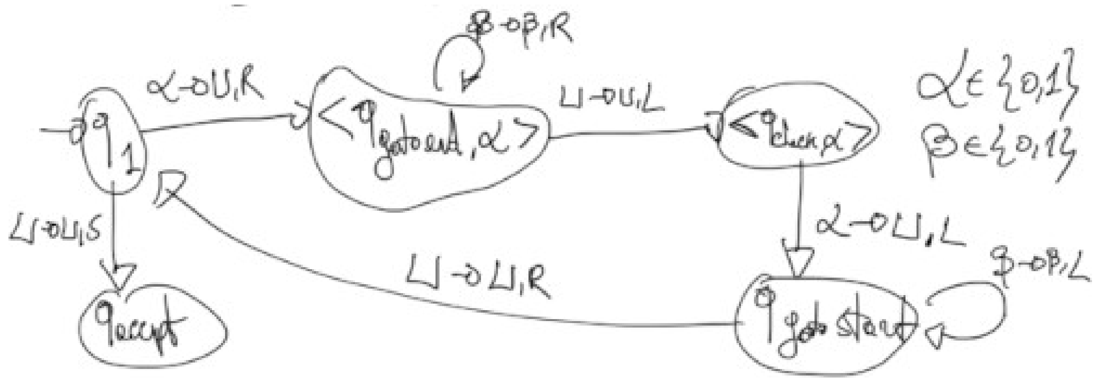
First, we identify what is the *language* decided by the Turing machine represented by the scheme. To undestand the overall behaviour we explore the possible paths to reach q_a. First, we can reach q_a from q_1 in just one step, so the empty string `_` is part of the language. Since `y` can't be part of the input string, the only remaining path is $q_1 \to q_{find1} \to q_{goback} \to q_1 \to q_{exit} \to q_a$. The following part is repeated until whole the string is in the form `xxx...yyy`.

- q_1 replaces the first `0` with an `x`.
- q_{find1} goes to the right and skips any `0` or `y` until a `1` is found, replaces the `1` with a `y`.
- q_{goback} goes to the right and skips any `0` or `y` until an `x` is found

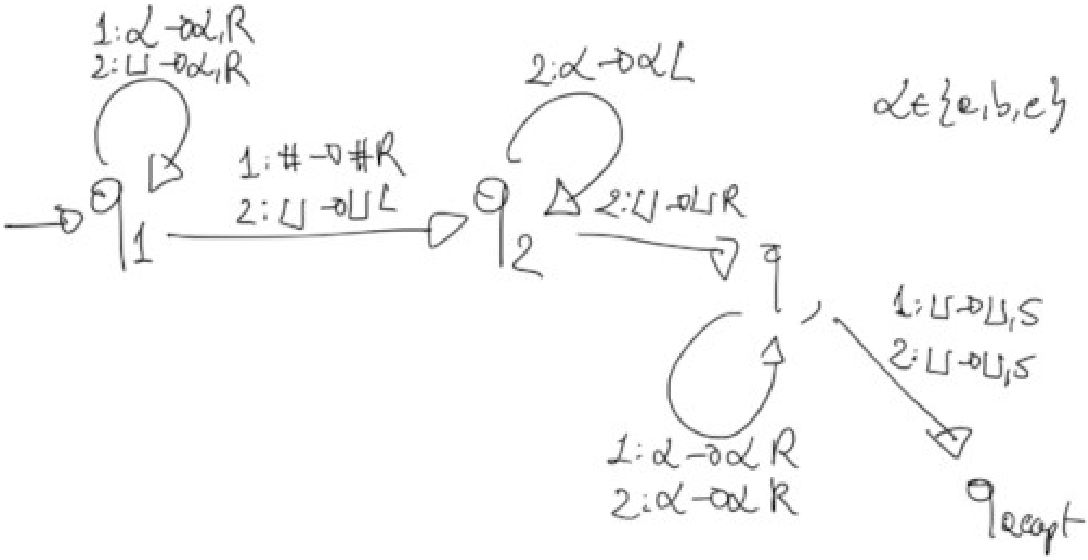
The last part in q_{exit} just moves from the middle of the string to its end checking that only `y` appears and accepts. So the language decided by the TM is $L = \{0^n 1^n | n \geq 0\}$.The TM is

- *Deterministic* since, for any input, there can be only one ending state.
- Works in $O(n^2)$: the worst case is $w \in L$, where each scan requires (forward and back) $|w| = n$ steps for a total of $n/2$ scans, plus $n/2$ for scanning any remaining `y`.

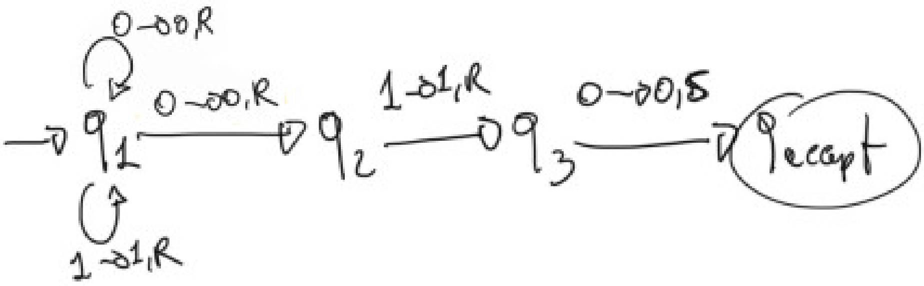
2.2 TODO Exercise 2



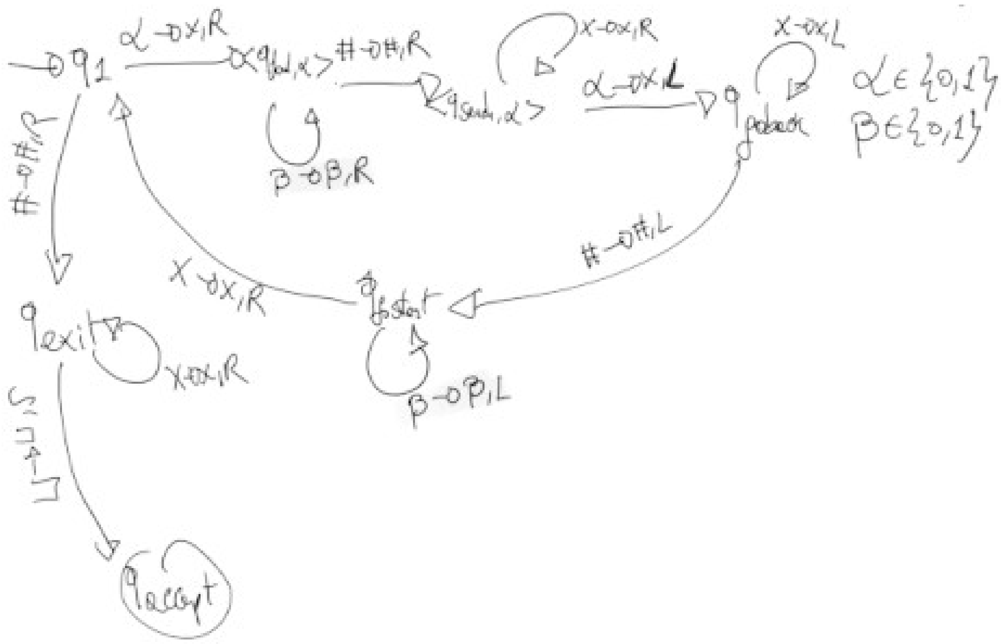
2.3 TODO Exercise 3



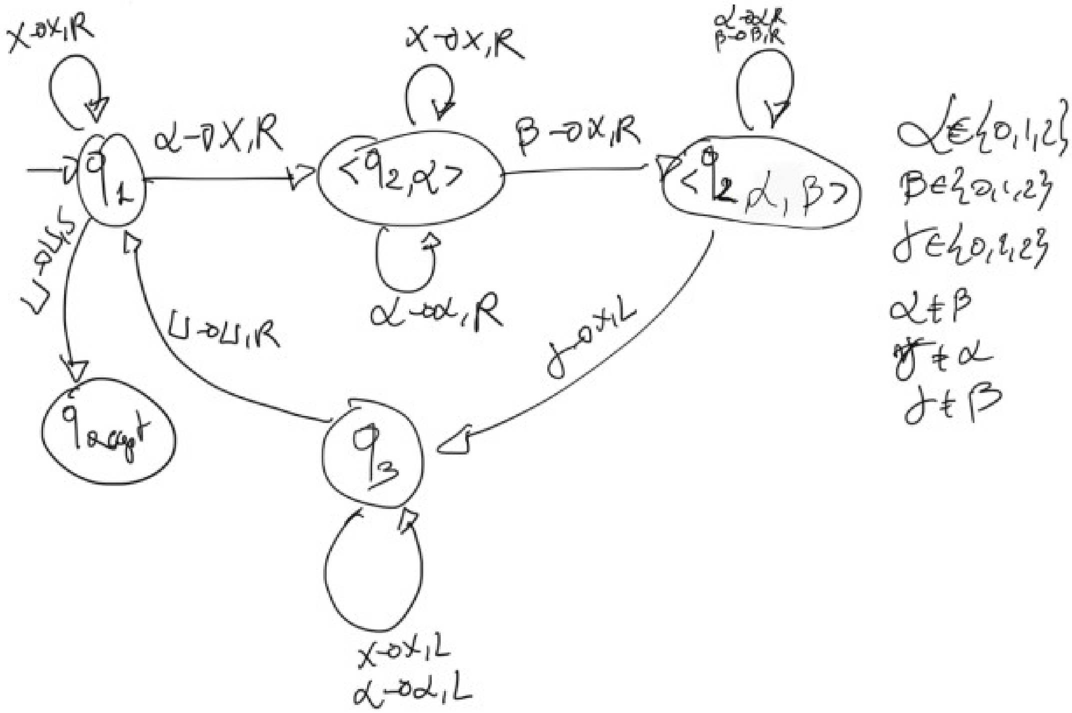
2.4 TODO Exercise 4



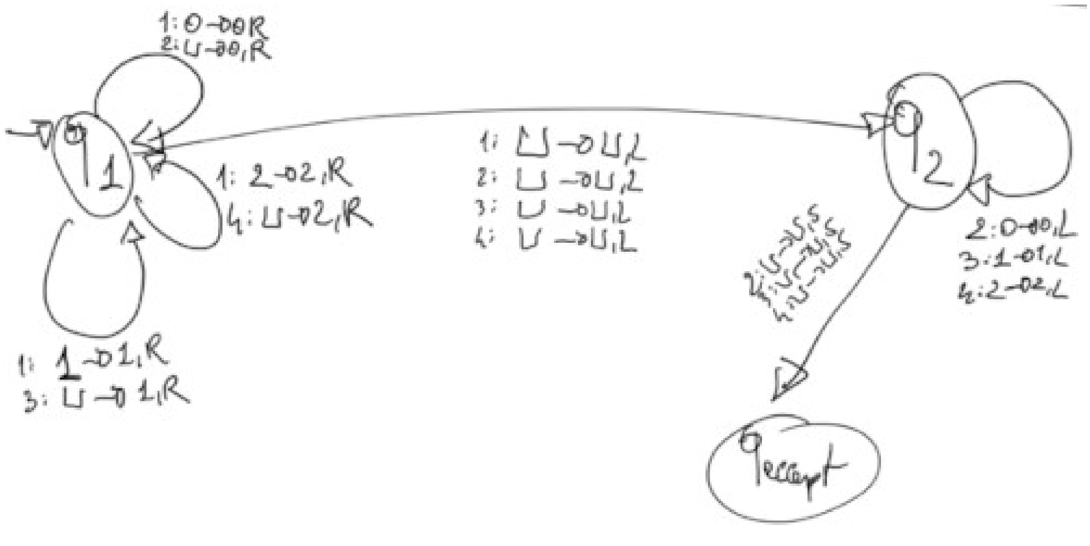
2.5 **TODO** Exercise 5



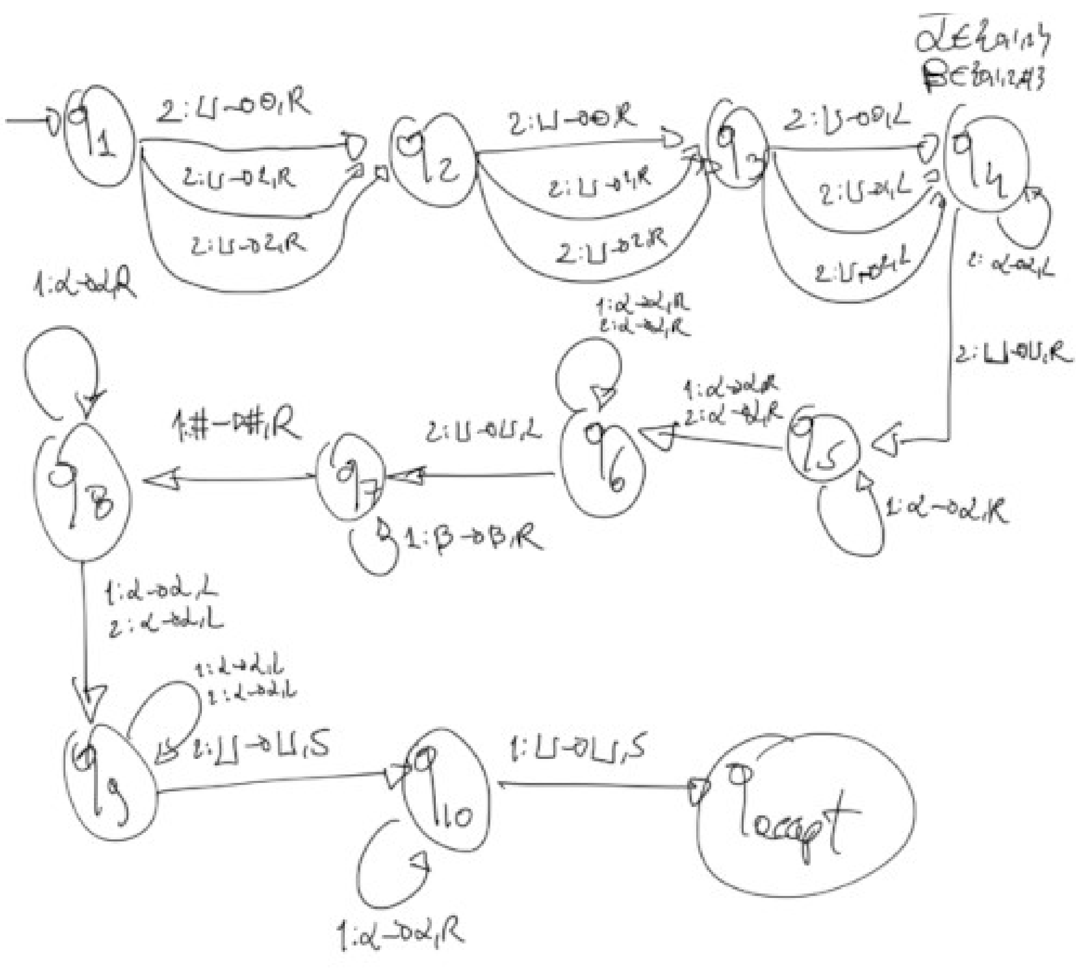
2.6 **TODO** Exercise 6



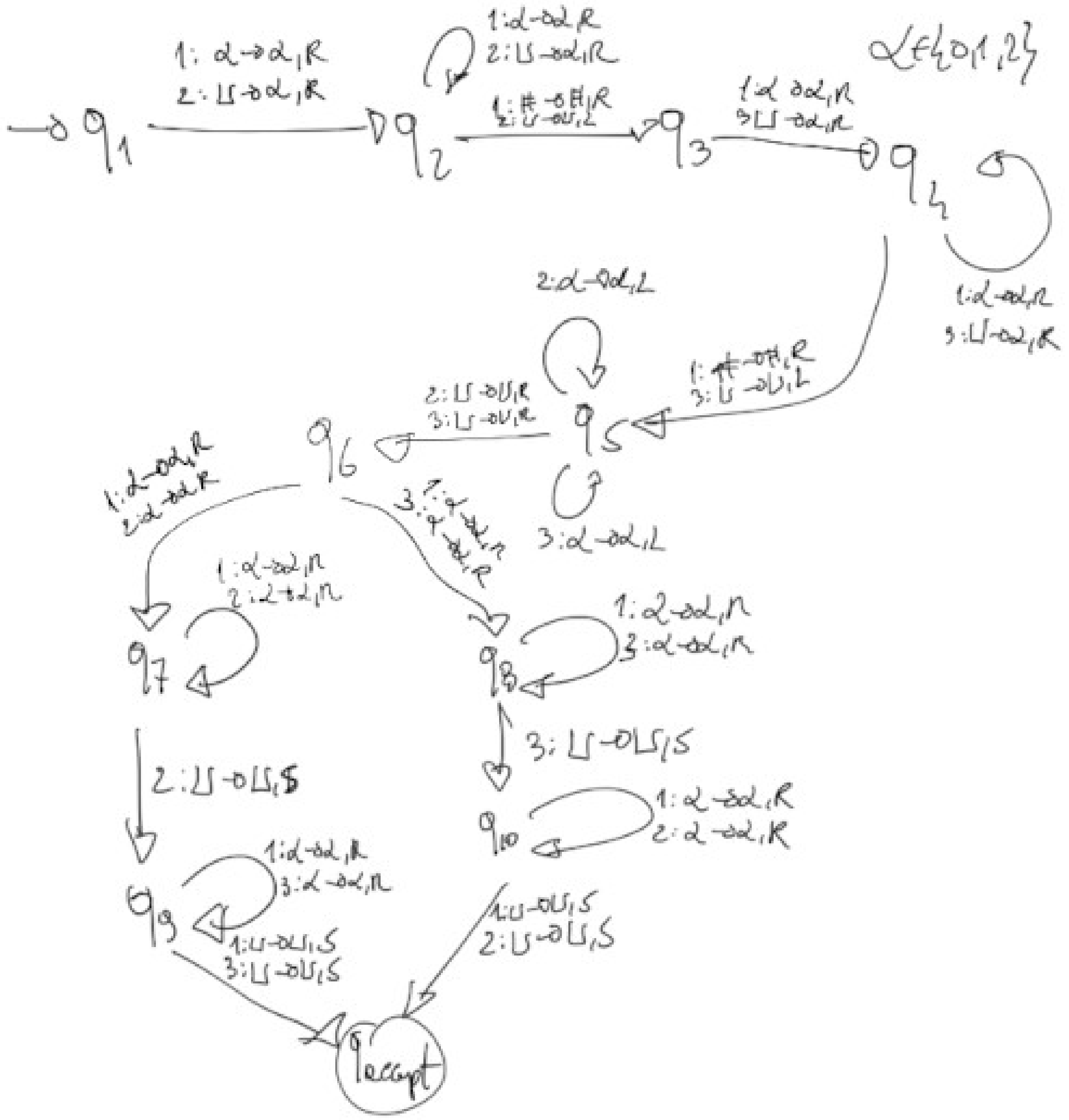
2.7 TODO Exercise 7



2.8 TODO Exercise 8



2.9 TODO Exercise 9



3 Exercises on undecidable properties

Consider the property $P = \{< M > | L(M)$ contains only strings of even length$\}$, is $P \notin R$?

- P is non-trivial, since $L = \{w | w = \{00\}\} \in P$ (P is non-empty) and $L = \{w | w = \{0\}\} \notin P$ (P doesn't contain all the TMs).
- P is semantic, since by definition if $K = L(M_1) = L(M_2)$ and
  - K contains only strings of even length then $< M_1 >, < M_2 > \in P$.
  - K contains some string of even odd length then $< M_1 >, < M_2 > \notin P$.

For Rice's Theorem, we can say that $P \notin R$.

Consider the property $P = \{< M > | L(M)$ is finite$\}$, is $P \notin R$?

- P is non-trivial, since $L = \{w | w = \{1\}\} \in P$ (P is non-empty) and $L = \{w | w = 1^n, n > 0\} \notin P$ (P doesn't contain all the TMs).
- P is semantic, since by definition if $K = L(M_1) = L(M_2)$ and
  - K is finite then $< M_1 >, < M_2 > \in P$.

- K is infinite then $< M_1 >, < M_2 > \notin P$.

For Rice's Theorem, we can say that $P \notin R$.

Consider the property $P = \{< M > | L(M) \text{ is infinite}\}$, is $P \notin R$?

- P is non-trivial, since $L = \{w | w = 1^n, n > 0\} \notin P$ (P is non-empty) and $L = \{w | w = \{1\}\} \in P$ (P doesn't contain all the TMs).
- P is semantic, since by definition if $K = L(M_1) = L(M_2)$ and
  - K is infinite then $< M_1 >, < M_2 > \in P$.
  - K is finite then $< M_1 >, < M_2 > \notin P$.

For Rice's Theorem, we can say that $P \notin R$.

Consider the property $P = \{< M > | L(M) \text{ is accepted only by TMs with 5 states}\}$, is $P \notin R$?

- P is trivial, since $P = \emptyset$: any language accepted by a 5-states machines can be accepted by a 6-states machine that only adds a dummy, unreachable state. So we can't use Rice's Theorem to prove its undecidability.