5CCS2FC2: Foundations of Computing II

Automata and Turing Machines

Week 1

Dr Christopher Hampson

Department of Informatics

King's College London

Non-deterministic Turing Machines (NDTMs)

Non-deterministic Turing Machines (NDTMs)

- **Non-deterministic Turing Machine**

$$M = \langle \Sigma, Q, q_{\text{init}}, q_{\text{accept}}, q_{\text{reject}}, \delta \rangle$$

  - **Transition function**

$$\delta : (Q \times \Sigma \cup \{\sqcup\}) \to \mathcal{P}(Q \times \Sigma \cup \{\sqcup\} \times \{\leftarrow, \rightarrow\})$$

$$\delta\left(\text{current state}, \text{read symbol}\right) \mapsto \left\{ \begin{array}{c} \text{possible instructions} \\ (\text{state}, \text{write}, \text{move}) \end{array} \right\}$$

Non-deterministic Turing Machines (NDTMs)

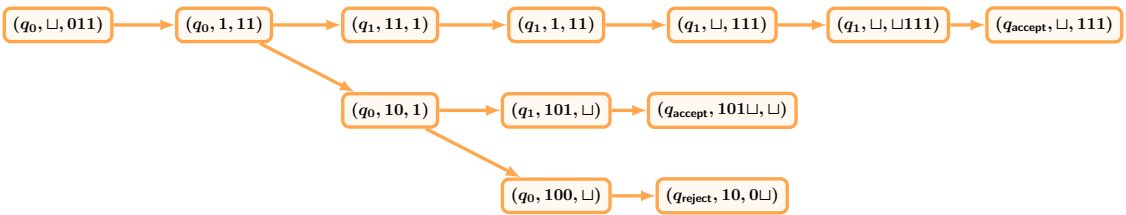
- **Configurations and Computations**
  - The **configurations** of a NDTM M are defined as for deterministic Turing Machines as a complete description of the machine at any given time

$$(\text{current state}, \text{ left substring}, \text{ right substring})$$

Non-deterministic Turing Machines (NDTMs)

- **Configurations and Computations**
  - The **computations** of a NDTM M are all possible sequences of configurations obtained by running M on a given input

- **Configurations and Computations**
  - These can be arranged into a branching **computation tree** with each branch corresponding to a possible computation of M.



- **Accept Criterion for NFAs**

M accepts w $\iff$ The final state of the **ANY** computation starting $(q_{\text{init}}, \varepsilon, w)$ is an accepting state

The Church-Turing Thesis

Theorem Every language that can be recognised by a **non-deterministic** TM can be recognised by a **deterministic** TM.

Proof:

- Given a NDTM M, we can construct a TM M' that writes out **all configurations** appearing in the branching computation tree of M.

- The machine M' enters the **accepting state** if it ever writes a configuration with whose state is the accepting state of M.

Q.E.D.

The Church-Turing Thesis

Theorem (The Church-Turing Thesis) Any language that can be effectively computed by some finite process can be recognised by a Turing machine.

End of Slides!