Esercitazioni: Macchine di Turing (~2h15m)

Informatica Teorica 23/24; Docente: Fabio Zanasi; Tutor: Gabriele Lobbia.

In questa esercitazione simuliamo delle Macchine di Turing usando delle applicazioni da browser.

Esercitazione 1: Macchine di Turing Classiche MT.zip (~1h30m)

Prerequisiti: Definizione di Macchina di Turing ed esempi, definizione di stringa accettata/rigettata, scaricare i file **MT.zip**.

Istruzioni iniziali

a. Scaricare il file **MT.zip** sul proprio computer.
b. Fare l'unzip (scompattare) il file suddetto in un folder qualsiasi (e.g. \informatica-teorica-esercizi).
c. Dare in input al browser il file **index.html** che si trova nella cartella precedentemente scompattata di cui al punto precedente a. Per questo si puo' per esempio trascinare il file index.html (drag) entro il browser).
d. Nella pagina del browser ci sono istruzioni su come programmare una Macchina di Turing (vedi sezione "**Definire la propria MT**").
e. Eseguire la macchina di "addizione di 1 in notazione binaria" preimpostata. Per prendere confidenza con lo strumento, svolgere uno o più degli esercizi indicati nel browser.

Esercizi

I seguenti esercizi sono riportati in ordine di difficoltà crescente. Il tempo indicativo per lo svolgimento è indicato in parentesi all'inizio. Si consiglia di svolgerne almeno due, con priorità suggerita agli esercizi sottolineati. In tutti gli esercizi bisogna programmare una macchina che, dato l'input descritto, svolga la funzione richiesta.

1. **Ripeti 01** [*10min*]
   Programmare una macchina che scrive indefinitamente su nastro vuoto una sequenza di "0" e "1". Questo è il primo esempio di Macchina di Turing dato da Turing stesso nel suo articolo "On Computable Numbers, with an Application to the Entscheidungsproblem" (1936).
2. **Copia 1** [*15min*]

Dato in input una stringa di “1” seguita da una stringa di “0” (P.e. 1111100000000) copiare la stringa di “1” dopo uno “0”: P.e. **11**000… => 110**11**00000.

Correzione insieme di un esercizio.

3. **Palindromi** [*20min*]
   Programmare una macchina che riceve come input una stringa di caratteri {“a”,”b”} e la accetta se questa è palindroma (P.e. a, ab aba, aabbaa) e la rigetta altrimenti.
4. **Addizione Binaria** [*25min*]
   Prendere come input due numeri binari divisi da un simbolo (P.e. 10011+101, oppure 1001*101) e restituire la somma di essi (P.e. in notazione binaria 10+1=11, 11+1=100).

Correzione insieme di un esercizio e introduzione al busy beaver game.

5. **Busy Beaver**
   Un “busy beaver game” consiste nel fissare un numero di stati e simboli e programmare una Macchina di Turing che si ferma e massimizzi la durata di esecuzione (numero di step/transizioni di stati) e/o la lunghezza dell’output.
   a. [*Spiegato in classe*] Risolvere il “busy beaver game” per una macchina con 2 stati (escluso quello di fermata) e 2 simboli {“0”,”1”}. Suggerimento: cercare di trovare una macchina che scriva il massimo numero di “1” sul nastro fermandosi; uno dei due stati deve per forza reagire ad uno dei due simboli con la fermata.
   b. [*Difficile, per casa*] Pensare a come risolvere il “busy beaver game” per una macchina con 4 stati (escluso quello di fermata) e 2 simboli {“0”,”1”}. Non si richiede di saper dimostrare che il programma descritto è effettivamente quello che massimizza la durata e/o lunghezza dell’output, ma provare a ragionare su come raggiungere l'obiettivo.

   Curiosità: La soluzione al busy beaver con 4 stati e 2 simboli è stata data da Allen Brady nel 1983. Ad oggi, non è nota la soluzione per 5 o più stati. Per ora sono state costruite macchine a 5 stati che si fermano in 47176870 step e a 6 stati in 7,4*10^36534 step (vedi “Given that 5-state 2-symbol halting Turing machines Collatz-like congruent functions, it may be very hard to find [the next busy beaver]” (https://oeis.org/A060843).

Pausa (15m)

Intermezzo Lezione: Equivalenza tra TMs e linguaggi di programmazione di alto livello (~30m)

Lezione di teoria sull’equivalenza tra macchine di turing e linguaggi di programmazione di alto livello, come WHILE.

Esercitazione 2: TM con Python MTPy.zip (~20m)

Prerequisiti: Definizione di Macchina di Turing ed esempi, definizione di stringa accettata/rigettata, scaricare i file **MT.zip** e **MTPy.zip**, lezione sull'equivalenza tra Macchine di Turing e linguaggi.

Istruzioni iniziali

Ora vediamo una definizione alternativa di diverse Macchine di Turing in Python.

a. Scaricare il file **MTPy.zip** (contiene una applicazione python).
b. Fare l'unzip (scompattare) il file suddetto in un folder qualsiasi (P.e. \informatica-teorica-esercizi).
c. Leggere il file **README.txt** e provare l'esempio eseguendo il comando li specificato.
d. Per definire una Macchina di Turing in questo modo bisogna descrivere il linguaggio (in formato json). Per esempio, lo stato "q0" nella macchina sum.json e' cosi' definito :

```
"q0": {
    " ": {
        "write": " ",
        "move": 1,
        "nextState": "q1"
    },
    "1": {
        "write": "1",
        "move": 0,
        "nextState": "q1"
    }
},
```

Esercizi

Riscrivere le Macchine di Turing descritte negli esercizi 1, 2, 3, 4 e 5 in questo linguaggio.