

# Analiza Algoritmilor

## Tema 1 - Codul Morse

Termen de predare: **7.Nov.2023, 23:50**

Data publicării: *22.Oct.2023*  
Ultima actualizare: *22.Oct.2023*

# 1 Cerință

În cadrul temei vom scrie tabelul de tranziție pentru o Mașină Turing ce se ocupă cu decodificarea de cod Morse. Alfabetul Morse este format din simbolurile . și -. O secvență în Morse codifică o literă din alfabetul latin conform tabelului din Figura 1.

A ● -	J ● - - -	S ● ● ●
B - ● ● ●	K - ● -	T -
C - ● - ●	L ● - ● ●	U ● ● -
D - ● ●	M - -	V ● ● ● -
E ●	N - ●	W ● - -
F ● ● - ●	O - - -	X - ● ● -
G - - ●	P ● - - ●	Y - ● - -
H ● ● ● ●	Q - - ● -	Z - - ● ●
I ● ●	R ● - ●	

Figure 1: Codificarea Morse.

**1. Decodificarea mesajelor.** Pe bandă se află un text codificat în cod Morse. Pentru această cerință este suficient să decodificați textul în alfabetul latin. Literele sunt despărțite prin spațiu, codificat prin simbolul \*. Cuvintele sunt despărțite de simbolul /.

---

Input: ...\*.\*.-.\*.-./--\*.  
Output: HELP/ME

---

**2. Identificarea cuvintelor cheie.** Dacă pe bandă se află unul din cuvintele cheie **SOS** sau **HELP** atunci mașina va semnala acest lucru prin tranziția în starea finală Y. În cazul în care niciun cuvânt cheie nu se regăsește pe bandă, mașina va tranziționa în starea finală N.

---

Input: ...\*- - - \*...  
Stare final : Y

---

---

Input: -. - \*.\*.\*. - . \* / \* - - - \*.. - \* -  
Stare final : N

---

Pentru punctaj parțial, puteți opta să implementați doar una dintre aceste verificări.

Dacă nu ați implementat ambele funcționalități, va trebui ca, în fișierul README, prima linie să conțină unul dintre următorii termeni, în funcție de cerința implementată.

- `translate` – pentru validarea decodificării
- `filter` – pentru validarea identificării cuvintelor cheie

Dacă ați implementat integral cerința temei, pe prima linie din README nu se va pune nici unul dintre termeni.

## 2 Format input

Inputul constă dintr-un șir ce reprezintă encodarea în alfabetul Morse a unui text alcătuit din literele alfabetului englez. Codificarea Morse folosește simbolurile . și -. Literele sunt despărțite de simbolul \*. Cuvintele sunt despărțite de simbolul /.

Formal, alfabetul de input este:

$$\Sigma_{morse} = \{., -, *, /\}$$

### 3 Format tabelă de tranziție

Pentru a redacta Mașina Turing există două posibilități:

1. Folosind formatul simulatorului online <sup>†</sup>.
2. Scriind funcția de tranziție în formă de tabel într-un spreadsheet.

#### 3.1 Sintaxa de simulator online

Descrierea unei mașini Turing se face într-un fișier text. Extensia recomandată este “.tms”.

Descrierea mașinii trebuie să înceapă cu două linii speciale:

1. Prima linie începe cu cuvântul cheie “init”, urmat de două puncte, “:”, apoi de numele stării inițiale. E.g.: “init: q0”
2. A doua linie este “accept: Y”. Simulatorul online vă permite să alegeți un nume arbitrar pentru starea acceptoare, ba chiar să aveți mai multe. *În contextul temei, este obligatoriu să aveți una singură, numită Y.*

După care urmează tranzițiile, înșirate una după alta. Specificarea fiecărei tranziții este formată din două linii:

1. Prima linie conține starea curentă și simbolul citit pentru care se aplică tranziția respectivă; starea și simbolul sunt separate printr-o virgulă
2. A doua linie conține starea în care mașina va trece, simbolul care trebuie scris pe bandă și direcția de mișcare a capului de citire, reprezentată de unul din simbolurile:
  - <: stânga
  - >: dreapta
  - -: rămâne pe loc

Pentru o descriere completă a sintaxei, urmăriți indicațiile de pe site.

*Nu este necesar* să definiți explicit toate valorile funcției de tranziție. Este posibil să existe combinații de (stare curentă, simbol citit) care să nu poată apărea în rularea mașinii pe niciun input. În acest caz, puteți să omiteți acea tranziție; acțiunea ei va fi tratată ca: N, \_, -.

#### 3.2 Formatul de spreadsheet excel

Descrierea unei mașini se face într-un fișier spreadsheet cu extensia .xlsx.

Fișierul trebuie să conțină un singur worksheet, ce conține tabelul de tranziție.

Tabelul de tranziții va conține pe prima linie lista cu toate simbolurile care fac parte din alfabetul de lucru,  $\Gamma$ , al mașinii, începând cu cea de-a două casuță (B1); prima casuță rămâne liberă. Simbolul blank poate fi reprezentat:

- fie de simbolul: □ (cod Unicode U+25A1)
- fie de un underscore: \_

Pe prima coloană, vor fi listate toate stările din mulțimea de stări  $Q$  a mașinii, începând cu cea de-a doua casuță (A2); prima casuță rămâne liberă. Prin convenție, prima stare din coloană (cea din casuță A2) este *starea inițială*  $q_0$  a mașinii. O stare este reprezentată dintr-un nume care poate consta în oricâte caractere.

În fiecare celulă din tabelul rezultat trebuie scris un triplet care corespunde valorii funcției de tranziție  $\delta$  pentru perechea formată din starea corespunzătoare liniei respective și din simbolul corespunzător coloanei respective. Tripletul constă în numele stării următoare, noul simbol scris pe bandă și direcția în care trebuie deplasat capul de citire:

- ← (caracterul Unicode U+2190) SAU < SAU L: stânga
- → (caracterul Unicode U+2192) SAU > SAU R: dreapta

---

<sup>†</sup><https://turingmachinesimulator.com/>

- - SAU H: rămâne pe loc

Nu este necesar să definiți explicit toate valorile funcției de tranziție. Este posibil să existe combinații de (stare curentă, simbol citit) care să nu poată apărea în rularea mașinii pe niciun input. În acest caz, puteți lăsa celula corespunzătoare din tabel goală; ea va fi tratată ca: N, □, -.

## 4 Checker și teste automate

Pentru a facilita dezvoltarea temei, vă punem la dispoziție un checker automat și o suită de teste.

Checkerul este scris în Python 3, dar nu e nevoie să interacționați cu codul său, ci doar să-l rulați; deci nu e nicio problemă dacă nu sunteți familiari cu Python. Pentru a-l rula aveți nevoie de pachetul `openpyxl`.

---

```
python3 -m pip install --user openpyxl
```

---

Checkerul are rolul principal de a citi și interpreta mașina redactată de voi și de a-i simula comportamentul pe un input.

În continuare vă prezentăm toate facilitățile checkerului, care ar putea fi utile în redactarea temei.

### 4.1 Rulare teste

Pentru a rula toate testele necesare, rulați:

---

```
./checker.py --tm morse.<tms sau xlsx> --run-tests
```

---

Checkerul va rula toate testele și va afișa, după fiecare, "PASS" sau "FAIL" după caz. Dacă rularea necesită mai mult de 100000 de tranziții, va fi oprită și va fi afișat "SLE" (Step Limit Exceeded). Pentru fiecare test cu rezultatul "PASS" veți primi un punct.

În urma rulării, checkerul creează un director `logs/` în care se vor găsi, pentru fiecare test, câte un fișier cu lista de configurații prin care a trecut mașina pe inputul respectiv, de la configurația inițială la cea finală (sau la configurația cu numărul 100000). Pentru testul cu numărul `i`, veți găsi fișierul `logs/dbglog_i`.

Testele se găsesc în fișierul `tests.py`. Formatul lor este: șirul de input între ghilimele, separat cu un ":" de o pereche (șir\_de\_caractere, stare\_finală). Perechea este formată dintr-un string ce reprezintă șirul de caractere din alfabetul englez care ar trebui să se regăsească pe bandă în urma decodificării de la cerința 1 și o stare 'Y'/'N' reprezentând starea finală în care ar trebui să tranziționeze mașina în urma implementării funcționalității de la cerința 2.

### 4.2 Rulare pe un singur input

Pentru a rula mașina voastră pe un input anume (e.g. ....\*.\*-..\*.-.\*/\*-\*.), rulați:

---

```
./checker.py --tm morse.<tms sau xlsx> --test-input "....*.*-..*.-.*/*-*."
```

---

Dacă rularea necesită mai mult de 100000 de tranziții, va fi oprită și va fi afișat mesajul "Step Limit Exceeded!".

În urma rulării, checkerul creează un fișier `debug.log` în directorul curent, ce conține lista de configurații prin care a trecut mașina pe inputul respectiv, de la configurația inițială la cea finală (sau la configurația cu numărul 100000).

### 4.3 Rulare pe criterii anume

Dacă vreți să rulați toate testele folosind verificarea pentru doar unul din criteriile de validare, puteți folosi argumentul `-validation-type`.

---

```
./checker.py --tm morse.<tms sau xlsx> --run-tests --validation-type translate
```

---

Criteriile posibile sunt: `translate`, `filter`, `all`. Mai multe criterii trebuiesc despărțite printr-un spațiu.

## 4.4 Conversie de la un format la altul

Dacă vrei să vezi reprezentarea mașinii în celălalt format (e.g. ai lucrat cu sintaxa de simulator online și vrei să vezi forma de table excel), poți converti mașina astfel:

```
./checker.py --tm morse.<tms sau xlsx> --output converted\_morse.<xlsx sau tms>
```

Checkerul “știe” ce format să citească verificând extensia; similar, știe ce format să scrie.

**Atenție 1.** Dacă ai lucrat într-un table excel și dorești să o converțiți în formatul `.tms` pentru a vedea live cum funcționează mașina voastră în simulatorul online, aveți grijă că setul de simboluri permis este mai restrâns; dacă folosiți caractere în afara celor ASCII, probabil simulatorul online le va respinge. Checkerul înlocuiește simbolul blank  cu underscore `_`

## 4.5 Setare număr maxim de pași

Pentru a rula mașina voastră pe un input (sau pe teste) cu o limită de pași custom (e.g. 10000000) poți folosi argumentul:

```
--max-steps=10000000
```

Scopul acestei opțiuni este strict ca să vă ajute la depanarea comportamentului mașinii. Tema va fi testată cu o limită de 100000.

## 5 Punctaj

Tema valorează 10 puncte din totalul de 60 de puncte de parcurs. Tema va fi punctată de checker de la 0 la 100, după următoarele criterii.

- decodificare morse **50 de puncte**
- identificare cuvinte cheie **50 de puncte**

Există 20 de teste; în funcție de ce e specificat pe prima linie din README, checkerul se va uita doar la criteriile respective (dacă în README nu e specificat nimic, checkerul va verifica validarea tuturor celor trei criterii).

Fiecare test valorează 5 puncte, din care 2.5 puncte se acordă pentru implementarea corectă a cerinței 1, iar alte 2.5 puncte pentru implementarea cerinței 2. Astfel, dacă ai implementat doar decodificarea mesajelor corespunzătoare cerinței 1 (criteriul `translate`), checkerul va acorda maxim 50 de puncte; dacă ai implementat doar identificarea mesajelor cheie corespunzătoare cerinței 2 (criteriul `filter`), checkerul va acorda maxim 50 de puncte.

Punctajul acordat de checker valorează 90% din punctajul temei, 10% din punctaj se obține pentru explicațiile oferite în fișierul README.

Restul de 10 puncte vor fi acordate în urma unei examinări manuale ce are ca scop evaluarea clarității implementării și a prezentării acesteia în README. Evaluarea va porni de la 10 și se vor scădea puncte pentru criterii ca:

- stări în plus în tabelă, la care nu se poate ajunge
- simboluri în plus în tabelă, care nu apar în input și nici nu sunt scrise
- lipsă README
- explicație insuficientă în README

**Atenție 2.** Scopul testelor este să ajute cu dezvoltarea cerinței și cu punctarea temei.

Voi trebuie însă să rezolvați *cerința*, indiferent de teste. În cazul în care implementarea voastră nu respectă cerința, punctajul oferit de checker poate fi anulat total sau parțial în urma examinării manuale.

De exemplu:

- dacă descoperiți că un test e greșit, semnalati acest lucru pe forum, nu modificați mașina încât să treacă testul greșit.
- o mașină care acceptă orice input va primi punctaj 0, chiar dacă pe checker vor trece toate testele în care inputul reprezintă un șir de caractere ce conține unul dintre cuvintele cheie.

## 6 Trimitere temă

Tema voastră va consta într-un fișier cu tabela de tranziție, cu unul din următoarele nume, în funcție de formatul folosit:

- morse.tms
- morse.xlsx

Alături de un fișier README în care să descrieți, high-level și succint, designul mașinii implementate. Fișierul trebuie să se numească README, nu ReadMe, readme etc. Trebuie să nu aibă nicio extensie (".txt", ".md" etc.) și să fie un format plaintext (nu pdf, doc, rich-text etc.)

Acestea trebuie puse într-o arhivă .zip și încărcate pe [Moodle](#).

## 7 Forum

Pentru orice întrebări sau neclarități legate de conținutul temei, folosiți [forumul temei](#). Evitați punerea de întrebări pe canale private (e.g. Teams, mail).

Nu postați pe forum fișierul cu implementarea voastră, sau fragmente din acesta. Nu postați pe forum fișierul cu lista de configurații obținute pe un input, sau fragmente din acesta.

## 8 Plagiat

**Rezolvarea temei este individuală.**

Copierea totală sau parțială a temei de la un alt student va rezulta în aplicare sancțiunilor menționate în regulamentul<sup>‡</sup> pentru *toți studenții implicați*, fiind irelevant cine de la cine a luat.

Desigur, aveți voie să discutați între voi idei generale de soluții. Dar nu vă uitați pe *implementarea* unui coleg; nu preluați exemple concrete de tranziții, nu cereți ajutor cu depănarea concretă a implementării voastre (i.e. nu prezentați cuiva lista cu configurații pe un anumit input).

În situațiile în care nu sunteți siguri ce constituie un act de plagiat, întrebați pe forumul temei.

---

<sup>‡</sup><https://ocw.cs.pub.ro/ppcarte/doku.php?id=aa:intro:rules>