# Calculabilitate & Complexităţi
# Subiectul 5

Complexitate spaţiu

## Ce tre să știi?

Nota 6:

- modelul de masina Turing pe care se face evaluarea masurii spatiu

- definitia masurii spatiu

- definirea claselor de complexitate spatiu

- comprimarea benzilor (enunturi)

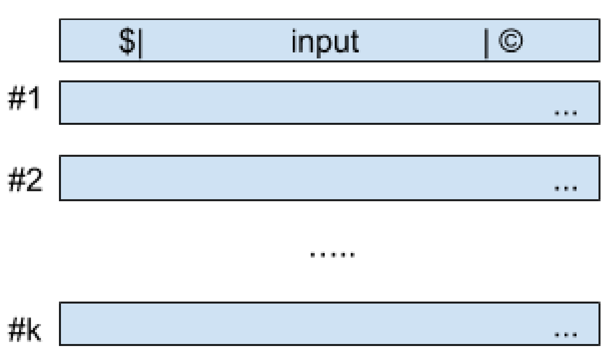
- eliminarea constantelor (enunturi)

- ierarhii de complexitate (enunturi)

Fiecare demonstratie, la alegere: 2p

## Modelul de mașină Turing folosit

Folosim Mașina Turing Offline - are o bandă pentru input read-only și k benzi auxiliare (de lucru) infinite doar la un capăt.



## Definiții

- **Space$_M$(n)** = Numărul maxim de celule utilizate de către M pe una dintre auxiliare pentru a decide orice intrare de lungime n.
- **(D/N)SPACE$_k$(f(n))** = {L | există o mașină Turing M **deterministă/nedeterministă** cu **k** benzi astfel încât **L(M) = L** și există $n_0$ cu **SPACE$_M$(n)** <= f(n) pentru orice n >= $n_0$}.
- Funcția f(n) se numește spațiu-construibilă dacă există o mașină Turing M și un n0 astfel încât **SpaceM(n) = f(n)** pentru orice n > n0.
- Funcția f(n) se numește spațiu-construibilă complet dacă există o mașină Turing M astfel încât **SpaceM(n) = f(n)** pentru orice n.

## Teoreme

- ((D/N)SPACE$_k$(f(n)) = (D/N)SPACE$_k$(c f(n)), unde c este o constanta pozitivă nenulă
- ((D/N)SPACE$_k$(f(n)) = (D/N)SPACE$_1$(f(n)), pentru orice k > 1 și orice f
- Oricare ar fi f(n) recursivă, există un limbaj recursiv L astfel încât L $\notin$ DTIME(f(n)).
    - Se aplică și pentru DSPACE, NTIME, NSPACE.
- Dacă S1(n) și S2(n) sunt spațiu construibile complet și S1(n)/S2(n) tinde la 0 când n tinde la infinit, DSPACE(S2(n)) \ DSPACE(S1(n)) $\neq \Phi$

# Ierarhie

- DTIME(f(n)) $\subseteq$ DSPACE(f(n))
- DSPACE(f(n)) $\subseteq$ DTIME($c^{f(n)}$), pentru f(n) $\geq$ log(n)
- NTIME(f(n)) $\subseteq$ NSPACE(f(n))
- NTIME(f(n)) $\subseteq$ DTIME($c^{f(n)}$)
- NSPACE(f(n)) $\subseteq$ DSPACE(f(n)$^2$), pentru f(n) $\geq$ log(n) și f spațiu construibilă complet (Teorema lui Savitch)
- DSPACE(log n) $\subseteq$ P $\subseteq$ NP $\subseteq$ NSPACE = PSPACE și DSPACE(log n) $\subset$ PSPACE

## Demonstrații

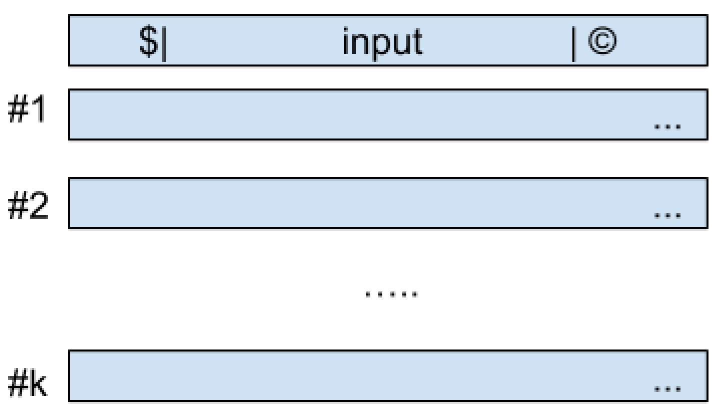
### ((D/N)SPACE$_k$(f(n)) = (D/N)SPACE$_k$(c f(n)), unde c este o constanta pozitivă nenulă.

Presupunem fără a restrânge generalitatea că c > 1. Dacă c = 1, este evident, iar daca c < 1, atunci considerăm 1/c.

Demonstrăm prin dublă incluziune.

Este evident că (**(D/N)SPACE$_k$(f(n))** este inclus în **(D/N)SPACE$_k$(c f(n))**, pur și simplu considerăm o mașină Turing care ocupă (inutil) c*f(n) celule pe o bandă.

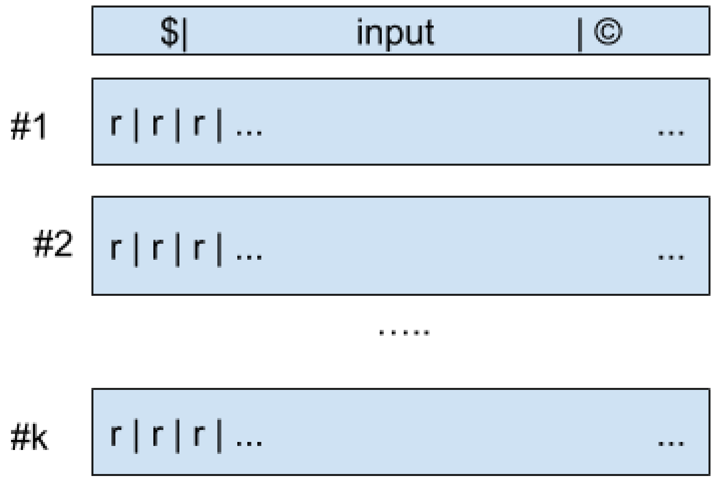
Considerăm M o mașină Turing:




Alegem un număr natural **r** cu proprietatea că **r > 2c**.

Mai întâi, împărțim benzile de la #1 la #k în segmente de câte r celule:


**Construim mașina M'**: fiecare celula a benzilor de lucru corespunde câte unui grup de r simboluri de pe benzile de lucru ale mașinii M (celulele devin vectori de lungime r).



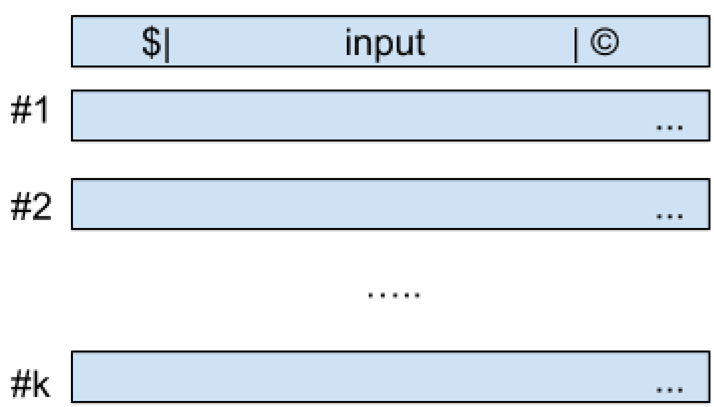

Mașina M’ va simula mișcările mașinii M.

**$Space_{M'}$(n)** <= parte întreagă superioară ($Space_M$(n) / r]    <= parte întreagă superioară (c * f(n) / r) <= **f(n)**

### ((D/N)$SPACE_k$(f(n)) = (D/N)$SPACE_1$(f(n)), pentru orice k > 1 și orice f.

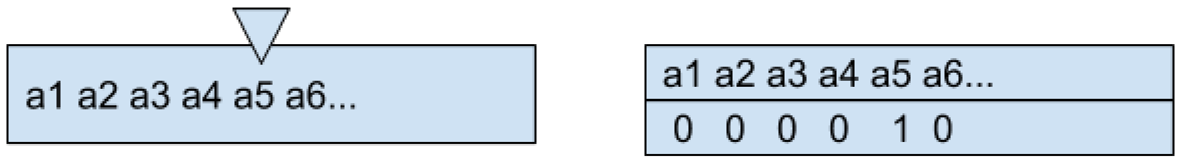
Demonstrația pornește de faptul că o mașină cu k benzi este echivalentă cu o mașină cu o singură bandă.

Avem mașina M, cu k benzi:




Și acum, construim mașina M’ astfel:

- M’ are o singură bandă auxiliară, iar elementele ei vor fi **vectori cu 2k piste:**
    - Pe pista 2 * i - 1 se află conținutul benzii i a mașinii M
    - Pista 2 * i conține 0-uri mai puțin pe o poziție - are 1 unde se afla capul de citire-scriere al benzii i a mașinii M.




Mașina M’ citește conținutul benzii auxiliare de la stânga la dreapta și memorează simbolurile ce ar trebui citite. Apoi, când ajunge la finalul benzii auxiliare, simulează mișcările pe care le-ar fi făcut mașina M. Parcurge din nou banda auxiliară, dar de la dreapta la stânga, și actualizează conținutul benzii.