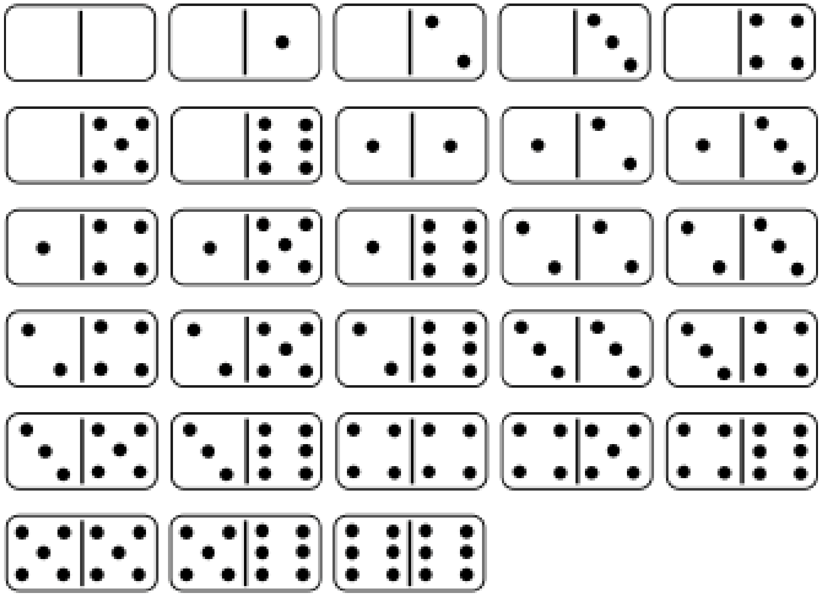
Grafuri euleriene
Aplicații

Grafuri euleriene

Problemă – joc domino

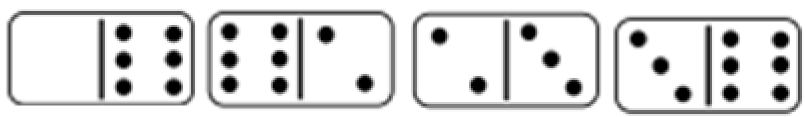
Piesă de domino – două fețe, numere 0..n, de obicei n=6



Grafuri euleriene

Problemă – joc domino

Șir de piese de domino – respectă regula de construcție: primul număr de pe piesa adăugată la șir = al doilea număr de pe ultima piesă din șir



Grafuri euleriene

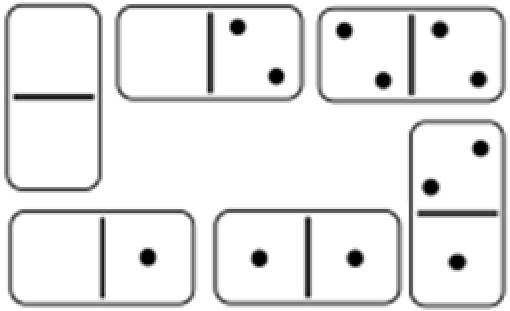
Problemă – joc domino

Se poate forma un șir de piese de domino care să conțină **toate piesele** + să se termine cu același număr cu care a început (un șir circular)?

Grafuri euleriene

Problemă – joc domino

Exemplu – daca folosim doar piese cu numere 0..2 putem forma un ciclu

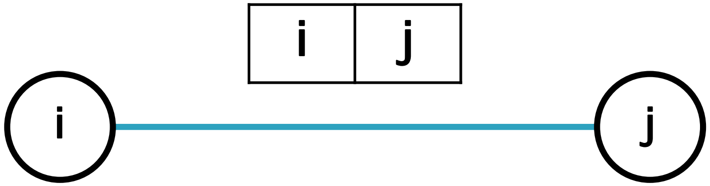


Grafuri euleriene

Problemă – joc domino

Graf asociat

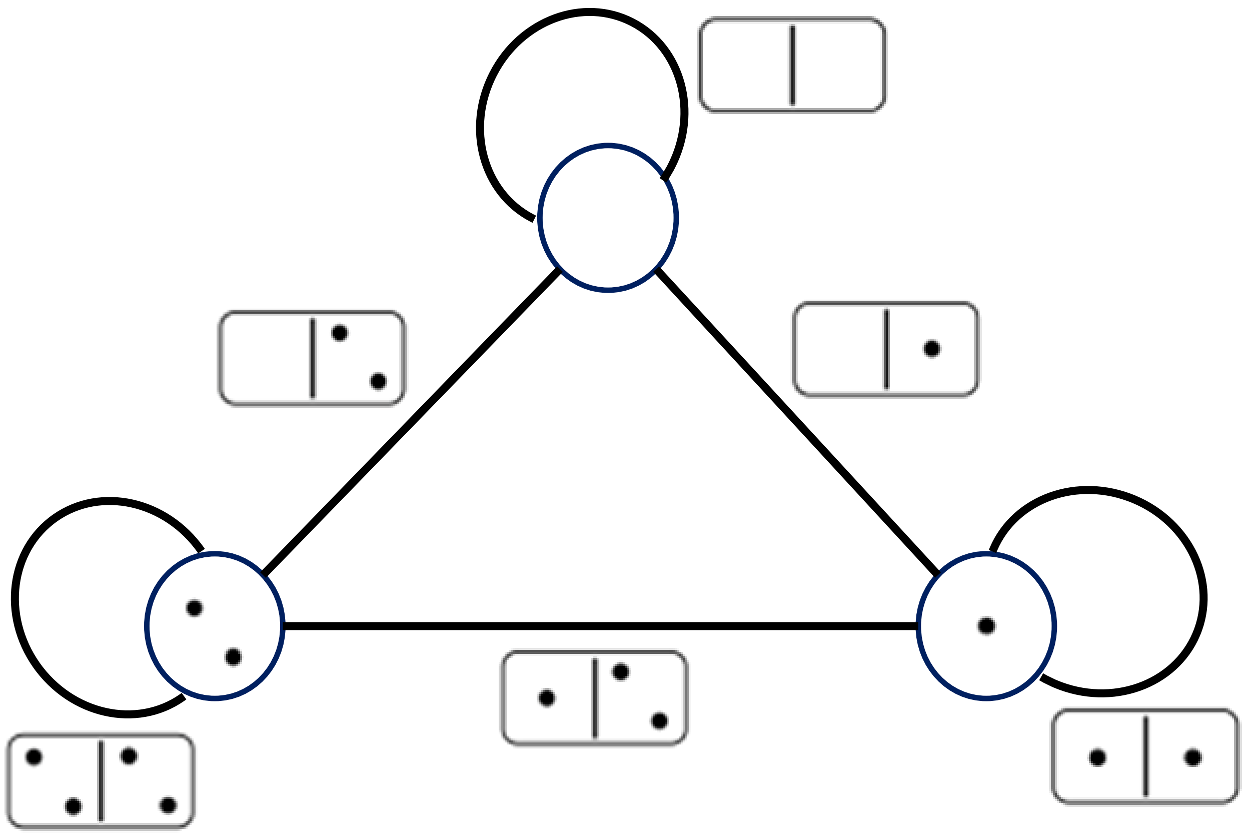
- vârfuri – numerele de pe piese
- muchii – perechi de numere (piesele)
- se pot lipi doar piese asociate muchiilor adiacente



Grafuri euleriene

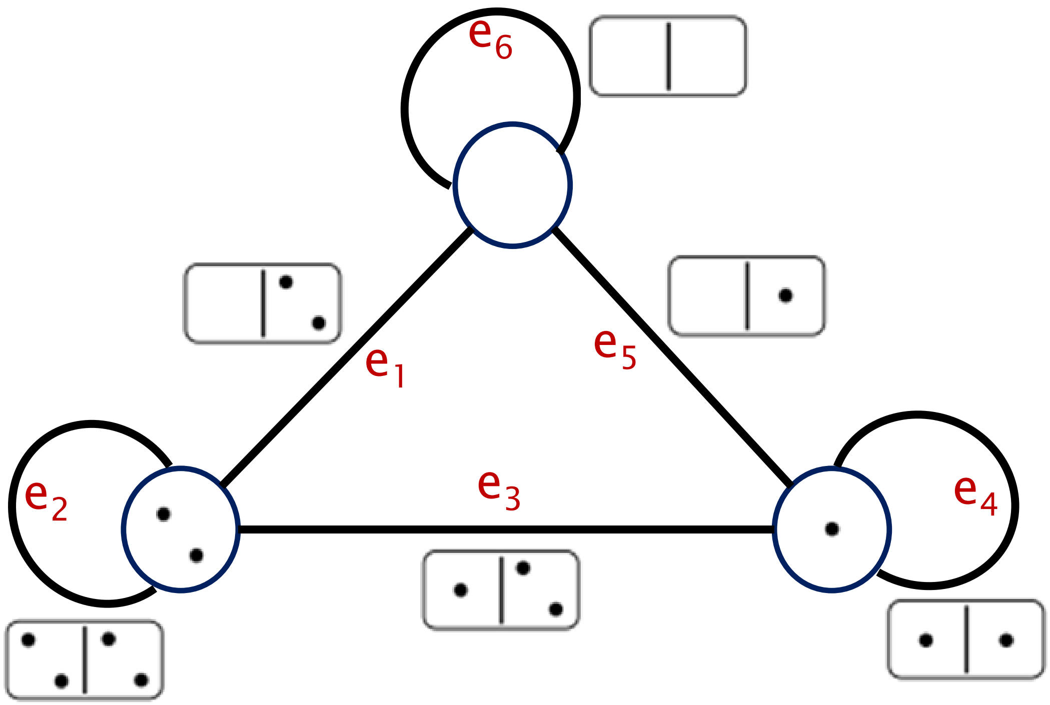
Problemă – joc domino

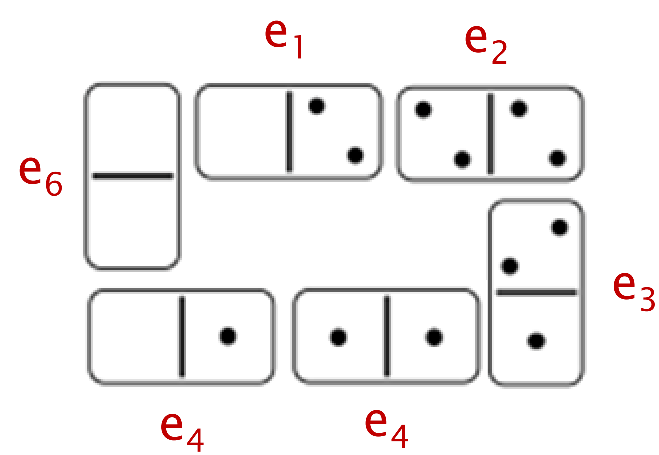
n=2



Grafuri euleriene

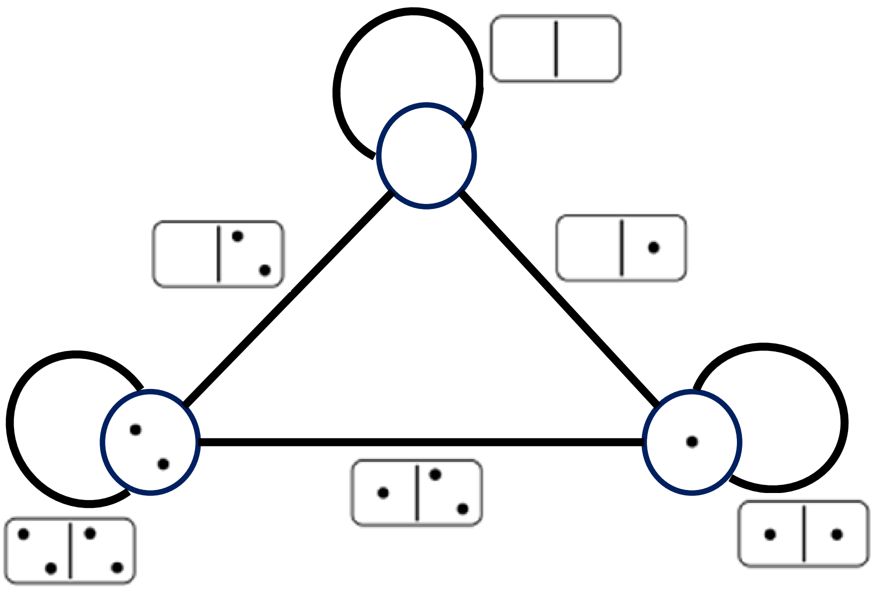
Există ciclu de piese ⇔ există ciclu eulerian în (multi)graf





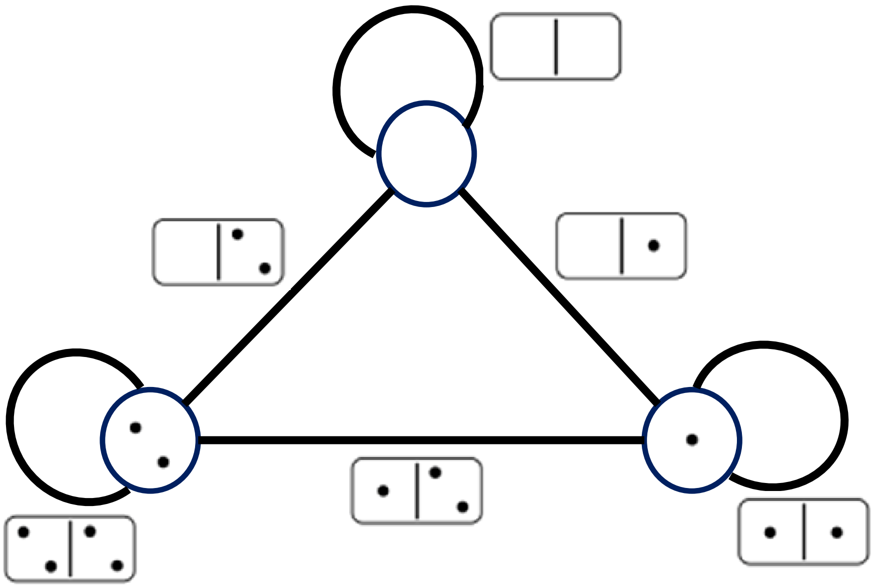
Grafuri euleriene

Pentru ce valori ale lui n există există ciclu eulerian în (multi)graful asociat?



Grafuri euleriene

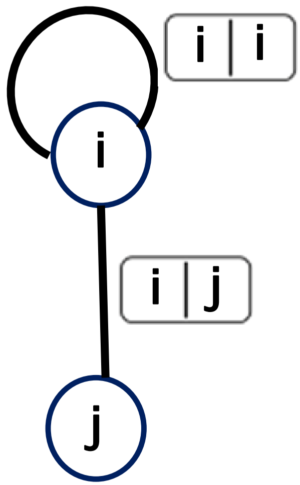
Pentru ce valori ale lui n există există ciclu eulerian în (multi)graful asociat?



d(i) = ? , pentru i = 0,…,n

Grafuri euleriene

Pentru ce valori ale lui n există există ciclu eulerian în (multi)graful asociat?



d(i) = n+2

(muchiile incidente în i sunt:

bucla etichetată (i,i) și

muchiile etichetate {i,j} cu $j \neq i$, $j \in \{0,\ldots,n\}$)

$\Rightarrow$ **trebuie ca n să fie par**

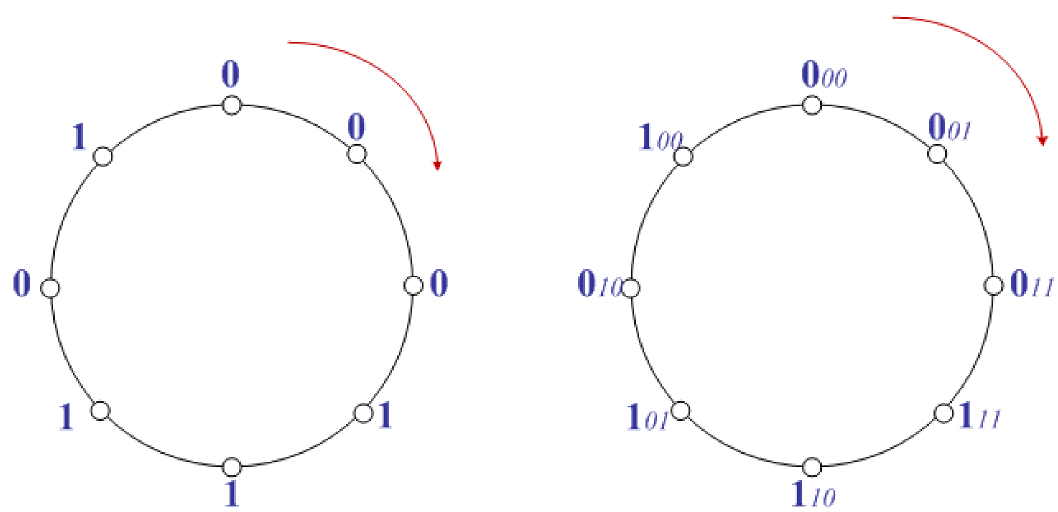
Grafuri de Bruijn

Grafuri euleriene

Problema lui POSTHUMUS

- $f(n)$ = numărul minim de cifre de 0 și 1 care se pot dispune circular a.î. între cele $f(n)$ secvențe de lungime n de cifre succesive apar toți cei 2^n vectori de lungime n peste {0,1} (citite în același sens).

- Evident $f(n) \geq 2^n$. **Are loc chiar egalitate?**



Grafuri euleriene

Variante

- Construcția unui șir pentru un alfabet pentru care se cunosc k-subsecvențele sale (toate)
- Construcția unui șir de lungime minimă pentru care să conțină subsecvențe (distincte, de lungime fixă/ oarecare) date

Grafuri euleriene

- **Universal string problem (de Bruijn, 1946)**

Determinați, dacă există, un șir care conține ca subsecvente toate șirurile binare de lungime k o singură dată

k = 3

000, 001, 010, 011, 100, 101, 110, 111

R: **00011101**

- **Shortest superstring problem:**

Date șiruri distincte (de aceeași lungime / nu neapărat) determinați cel mai scurt șir care conține ca subsecvențe toate aceste șiruri.

ACG, GCA, CGC, CGT, GAC, GCG, GTA, TCG

R: TCGACGCGTA

Grafuri euleriene

Universal string problem (de Bruijn, 1946)

- **k = 3:   000, 001, 010, 011, 100, 101, 110, 111**

Varianta 1: Modelare cu **graf de suprapuneri**:

  - vârf = k−șir
  - arc = **se poate continua cu**

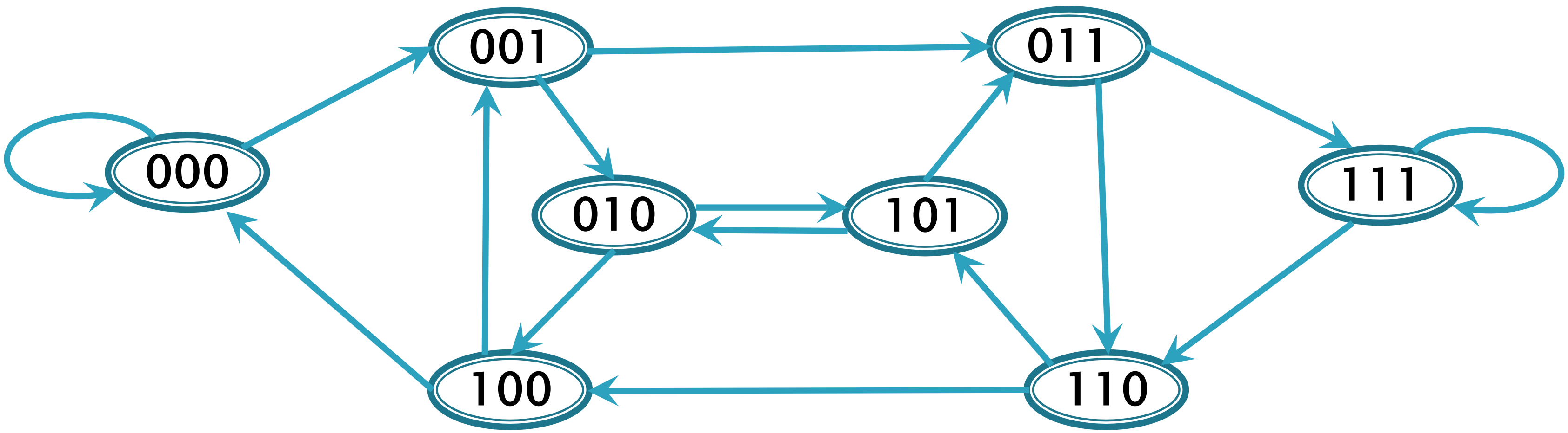
Grafuri euleriene

Universal string problem (de Bruijn, 1946)

- k = 3:   000, 001, 010, 011, 100, 101, 110, 111

Varianta 1: Modelare cu **graf de suprapuneri**:

- vârf = k-șir
- arc = **se poate continua cu**

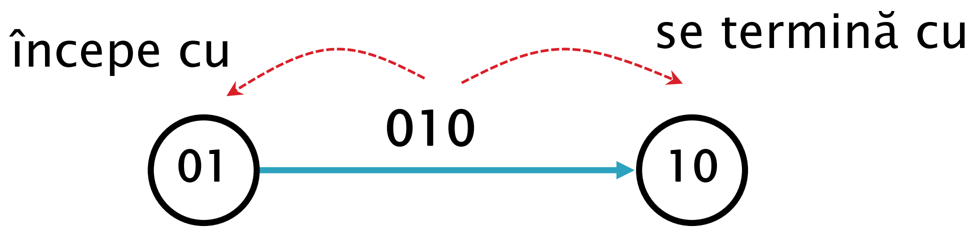


Pb se reduce la a determina dacă există drum hamiltonian în graf – dificil computațional

Grafuri euleriene

Varianta 2: Modelare de Bruijn a surpapunerilor – multigraf:

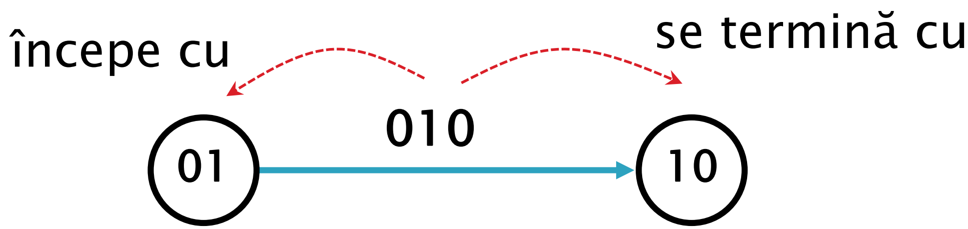
- Arcele corespund k șirurilor
- Vârfurile corespund (k-1)-șirurilor și arată suprapunerile

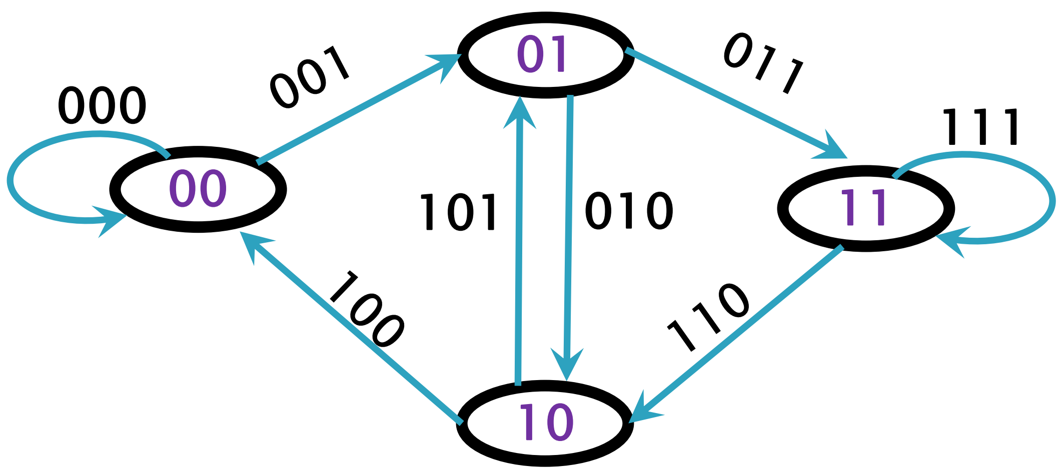


Grafuri euleriene

Varianta 2: Modelare de Bruijn a surpapunerilor – multigraf:

- Arcele corespund k șirurilor
- Vârfurile corespund (k-1)-șirurilor și arată suprapunerile

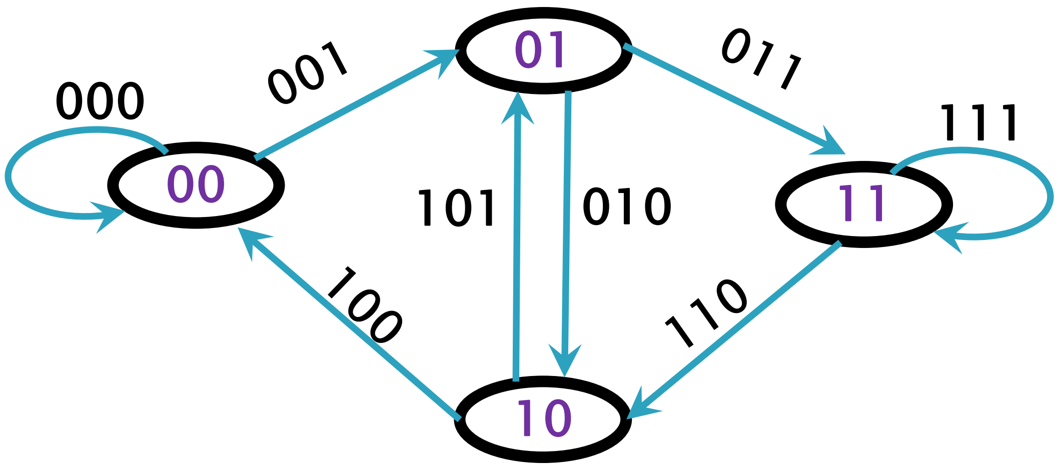




Problema se reduce la a determina dacă există drum/circuit **eulerian** **în graf – polinomial**

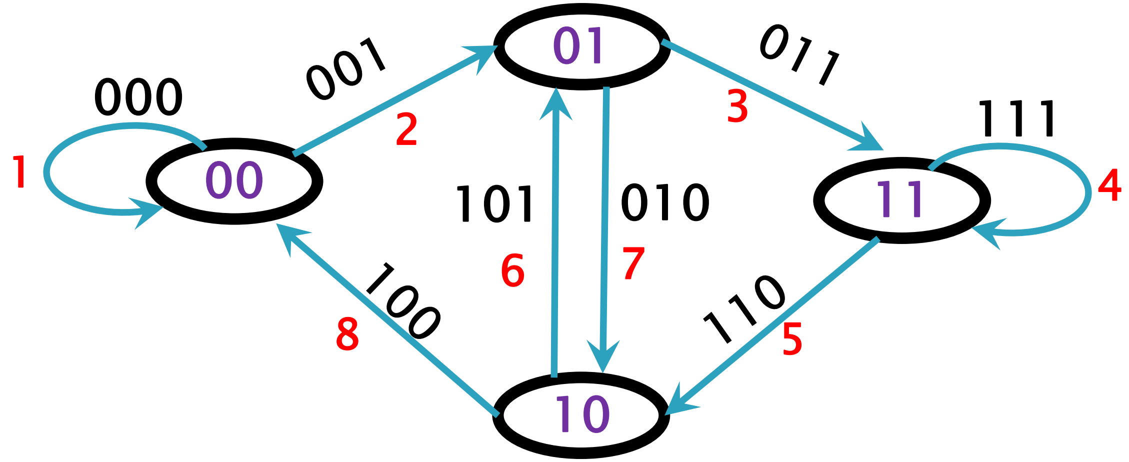
Grafuri euleriene

- Este eulerian graful de Bruijn asociat k-secvențelor binare?



Grafuri euleriene

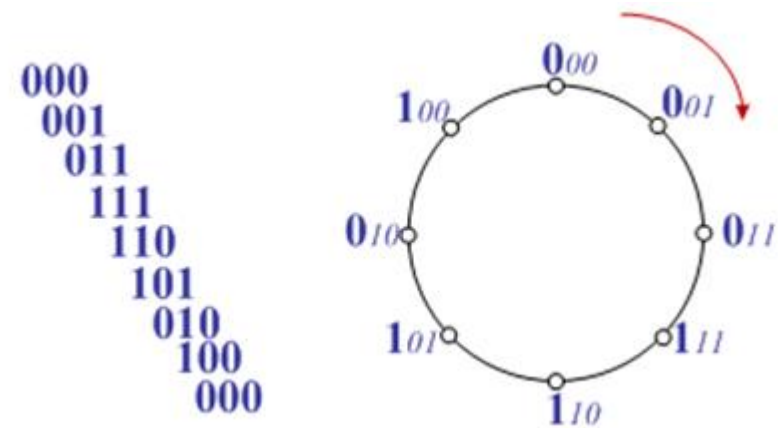
- Este eulerian graful de Bruijn asociat k-secvențelor binare?



Circuit eulerian 00 – 00 –01 –11 –11 – 10 – 01 –10 – 00

indicat cu numere pe arce – soluție la problema lui POSTHUMUS pentru n=3

R: 00011101
(prima cifră din etichetele arcelor)



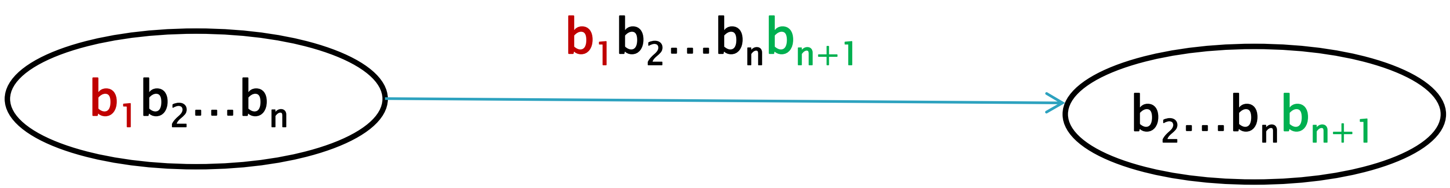
Grafuri de Bruijn

Multigraf

- $V(B_n) = \Sigma^n$ (unde $\Sigma = \{0,1\}$ mai general $\{0,1,\ldots, p\}^n$)

  (sau cuvinte de lungime n peste un alfabet finit)

- $E(B_n)$ etichetate cu Σ^{n+1}

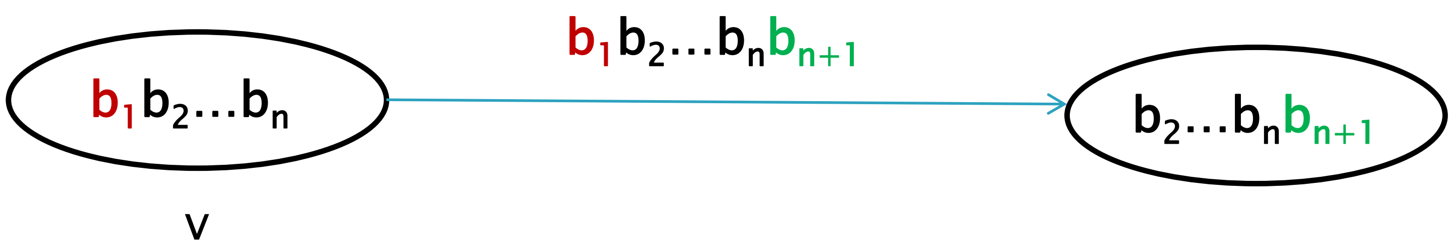


Grafuri de Bruijn

- B_n este eulerian?

  $d^+(v) = ?$

  $d^-(v) = ?$



orice b_{n+1} din alfabet

Grafuri de Bruijn

- B_n este eulerian

$$d^+(v) = |\,\Sigma\,|$$

$$d^-(v) = d^+(v)$$

$$\underset{v}{b_1b_2\dots b_n} \xrightarrow{b_1b_2\dots b_nb_{n+1}} b_2\dots b_nb_{n+1}$$

orice b_{n+1} din alfabet

Grafuri de Bruijn

- **Observație**

Circuit eulerian in $B_{n-1} \leftrightarrow$ circuit hamiltonian in B_n

Shortest superstring problem

Shortest superstring problem

Date s_1, s_2, ..., s_n distincte să se determine cel mai scurt șir s care conține ca **subsecvențe** șirurile s_1, s_2, ..., s_n

ACG, CGC, CGA, CGT, GAC, GCG, GTA, TCG

R: TCGACGCGTA

Shortest superstring problem

Notăm

- suprapunere(a,b) = lungimea maximă a unui sufix al lui a care este prefix al lui b
- suprapunere(“AGCT”, “CTAA”) = 2
- suprapunere(“AGCTA”, “CTAA”) = 3

Shortest superstring problem

- NP–completă
- Algoritm greedy – nu optim:

Shortest superstring problem

- NP-completă
- Algoritm greedy – nu optim:

Fie $S = \{s_1, \ldots, s_n\}$

Cât timp $|S|>1$:

- Alege 2 șiruri a și b cu suprapunerea maximă
- elimină a și b din S
- adaugă in S șirul obținut prin suprapunerea lui a și b

Shortest superstring problem

- NP–completă
- Algoritm greedy – nu optim:

Fie $S = \{s_1,\ldots,s_n\}$

Cât timp $|S|>1$:

- Alege 2 șiruri a și b cu suprapunerea maximă
- elimină a și b din S
- adaugă in S șirul obținut prin suprapunerea lui a și b

Lungime șir obținut ~ 2,5 *optim

Shortest superstring problem

- Exemplu algoritm greedy:

AAA AAB ABB BBA BBB

AAAB ABB BBA BBB

AAAB ABBA BBB

AAABBA BBB

AAABBABBB => superstring

Există însă un superstring mai scurt: AAABBBA

https://www.coursera.org/learn/dna-sequencing

Shortest superstring problem

- Modelare cu grafuri:

 – reducere la determinarea unui drum hamiltonian –
(! probleme NP-complete + algoritmi suboptimali)

– în anumite cazuri reducere la determinarea unui drum eulerian – polinomial

Shortest superstring problem

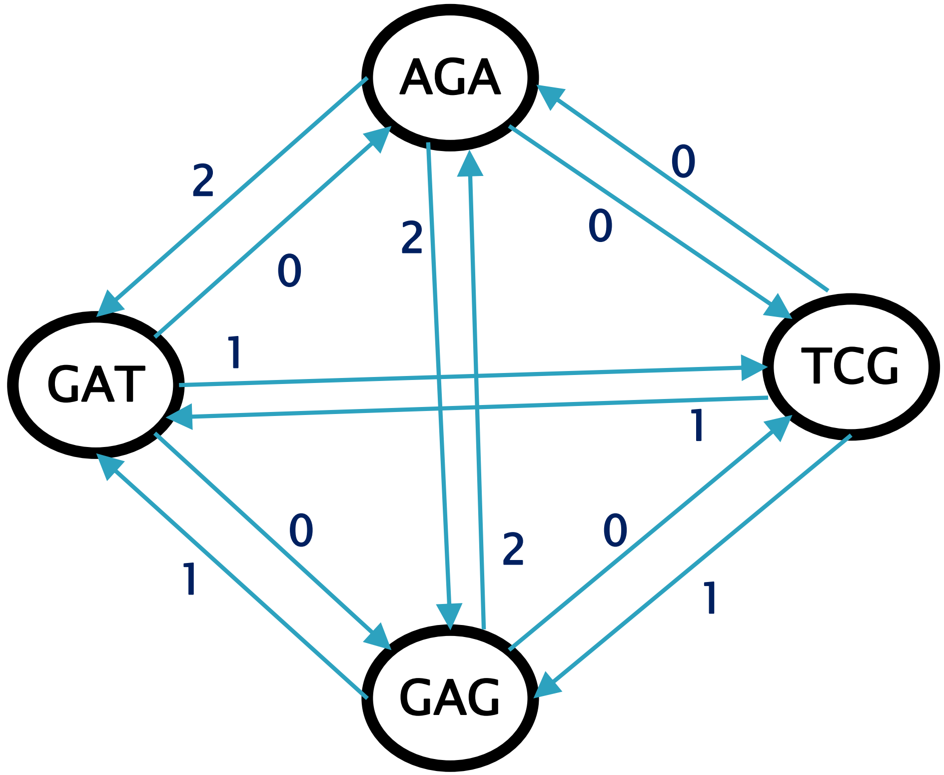
- Modelare cu grafuri:

Graful suprapunerilor

- Vârfuri – șirurile $s_1,\ldots,s_n$
- Arce – între oricare doua vârfuri a și b distincte, cu ponderea suprapunere(a,b)

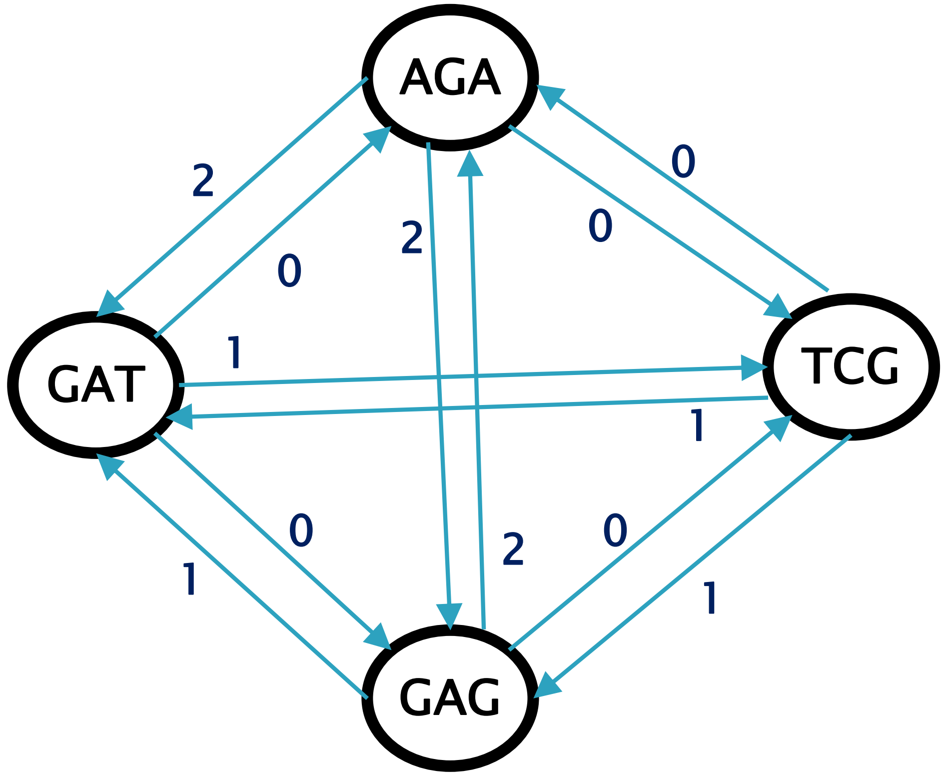
Shortest superstring problem

S = {AGA, GAT, GAG , TCG }



Shortest superstring problem

S = {AGA, GAT, GAG , TCG }



Drum hamiltonian de cost maxim (costul suprapunerilor)
=> cel mai scurt supersir (TSP - NP-complet)

Shortest superstring problem

- Modelare cu grafuri:

Graful suprapunerilor

- Vârfuri – șirurile $s_1,\ldots,s_n$
- Arce – între oricare doua vârfuri a și b distincte, cu ponderea suprapunere(a,b)

- Superstring de lungime maximă $\Leftrightarrow$ drum hamiltonian de cost maxim (sau de cost minim dacă luăm negativul ponderilor) => TSP NP-completă
- este însă utilizat în probleme de asamblare – algoritmi aproximativi

Shortest superstring problem

Caz particular – știm **toate subsecvențele de lungime k ale unui șir** (**k-spectrul șirului**). Să se determine un cel mai scurt superstring al lor (!!nu e unic)

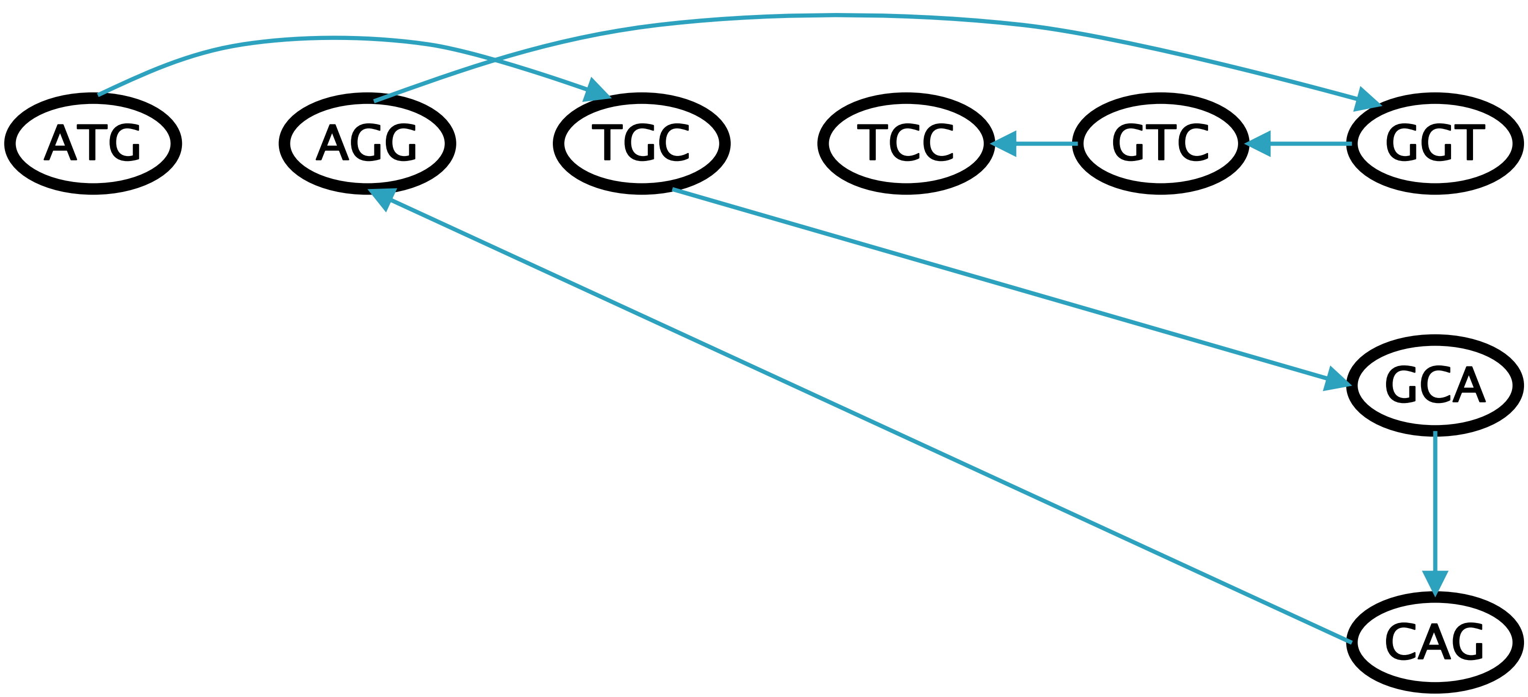
Spectru(GTATCT,2) = Spectru(GTCTAT,2) = {AT, CT, GT, TA, TC}

- în graful suprapunerilor ponderile sunt k-1

Shortest superstring problem

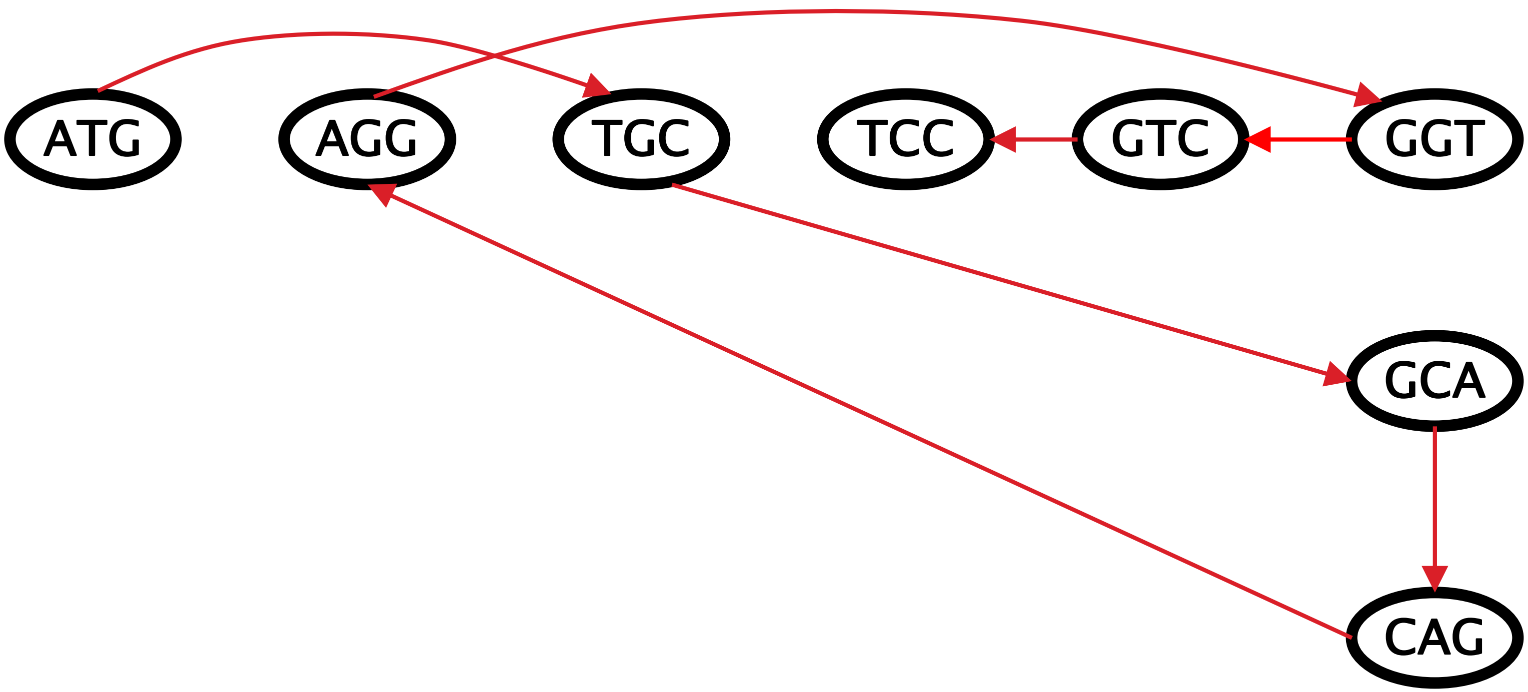
S = {ATG, AGG, TGC, TCC, GTC, GGT, GCA, CAG}

Graful suprapunerilor asociat



Shortest superstring problem

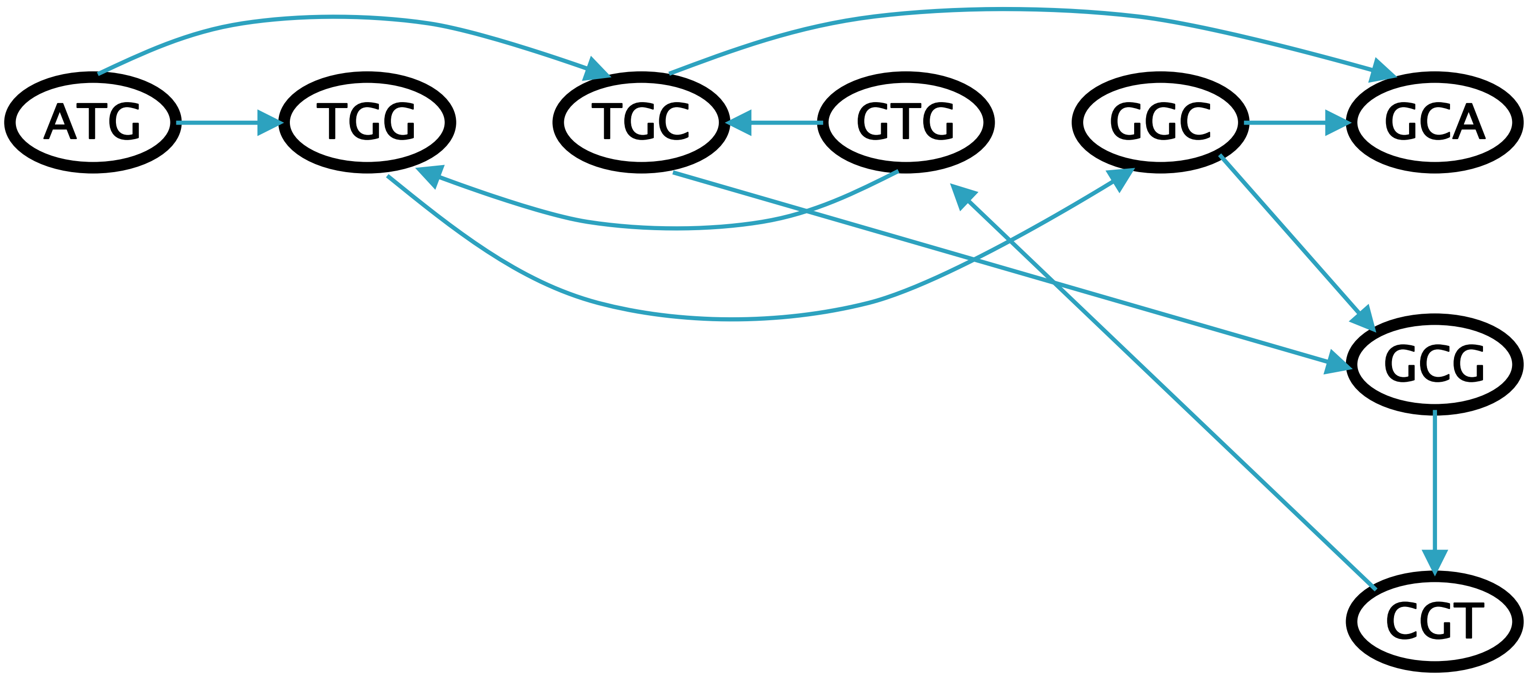
S = {ATG, AGG, TGC, TCC, GTC, GGT, GCA, CAG}



Drum Hamiltonian unic: ATG, TGC, GCA, CAG, AGG, GGT, GTC, TCC => superstring **ATGCAGGTCC**

Shortest superstring problem

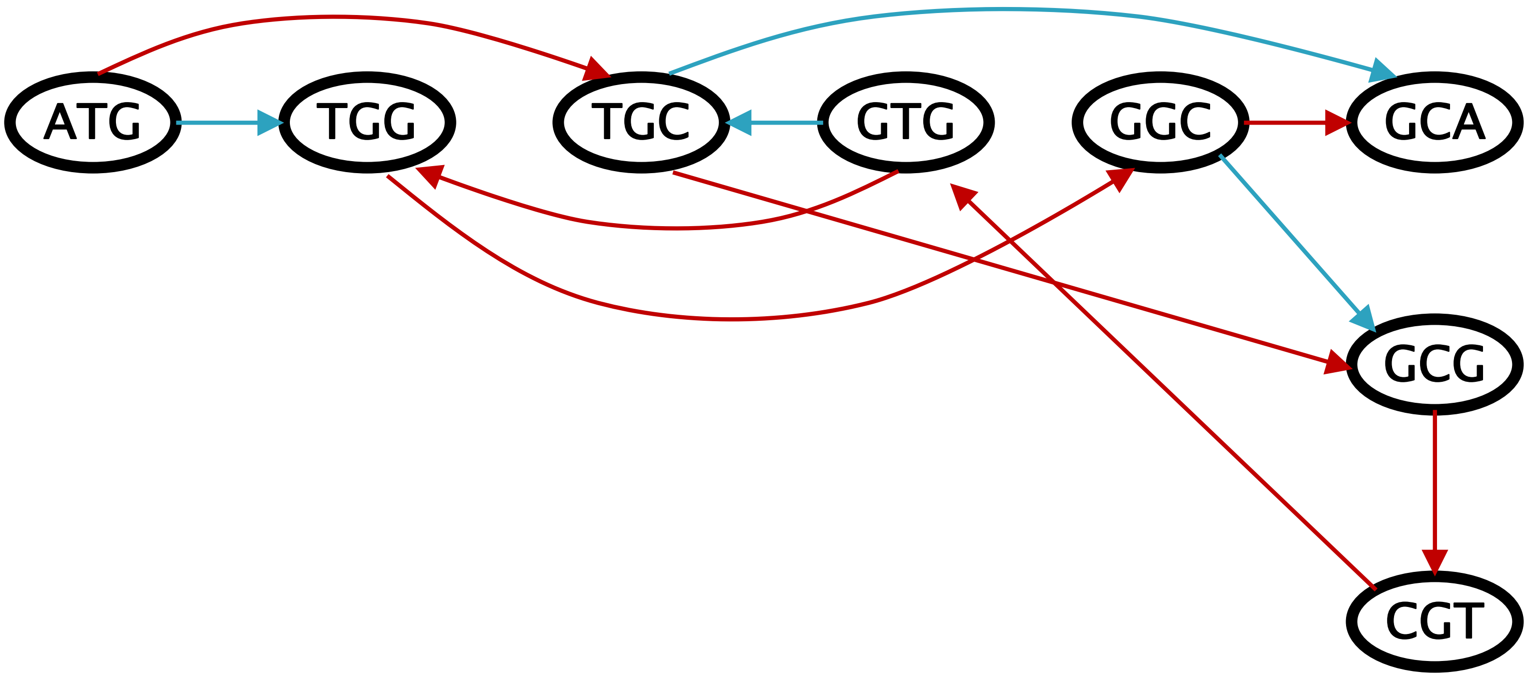
S = {ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT}



Drum Hamiltonian – nu este unic neapărat

Shortest superstring problem

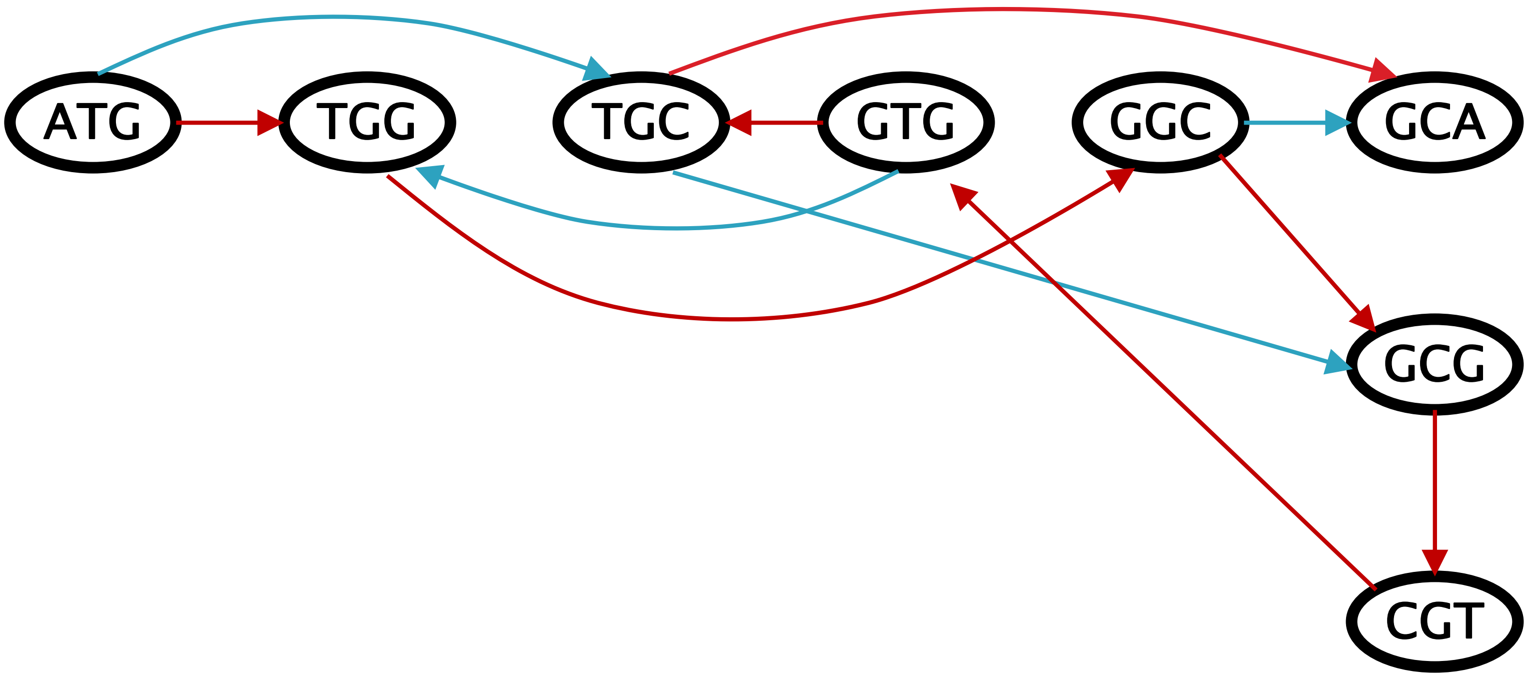
S = {ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT}



P1: ATG, TGC, GCG, CGT, GTG, TGG, GGC, GCA =>
Superstring=ATGCGTGGCA

Shortest superstring problem

S = {ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT}



P2: ATG, TGG, GGC, GCG, CGT, GTG, TGC, GCA=>
Superstring=ATGGCGTGCA

Shortest superstring problem

Caz particular – știm toate subsecvențele de lungime k ale unui sir (k-spectrul șirului).

Cu graf de suprapuneri – problema se reduce la determinare de drum hamiltonian

Shortest superstring problem

Caz particular – știm **toate** subsecvențele de lungime k ale unui sir (k-spectrul șirului).

Putem asocia Graf de Bruijn

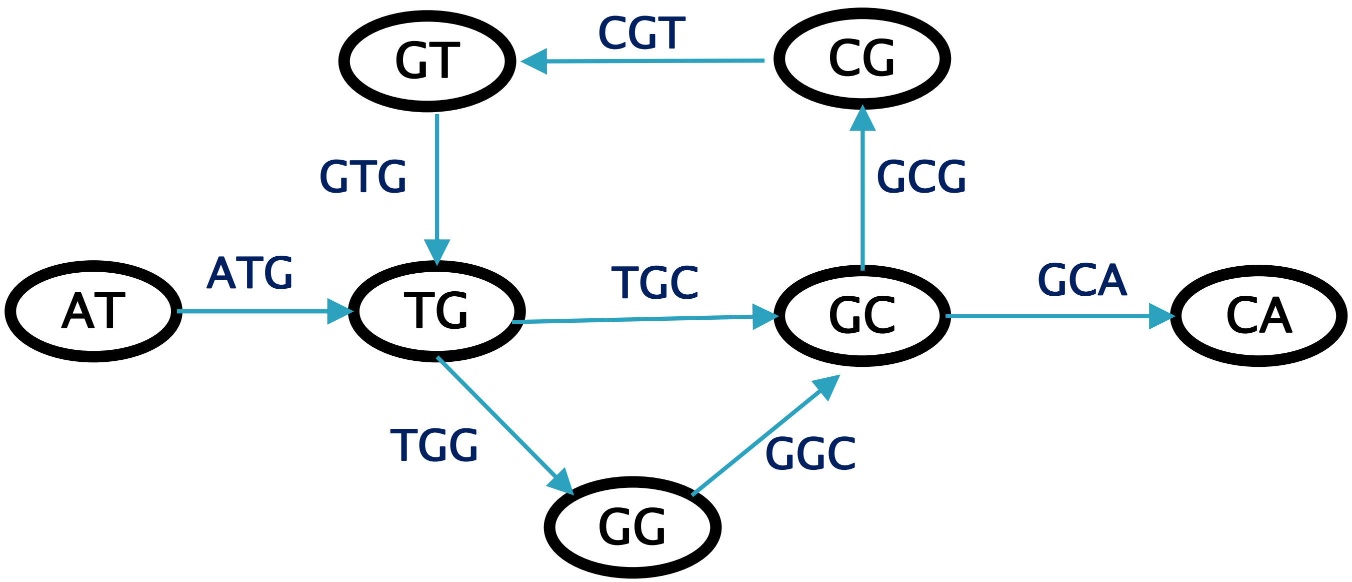
- **Arcele corespund k-subsecventelor**
- **Vârfurile – (k-1)-subsecvente** (corespunzătoare suprapunerilor, toate suprapunerile au valoare k-1)

- **Problema se reduce la drum eulerian**

Shortest superstring problem

S = {ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT}

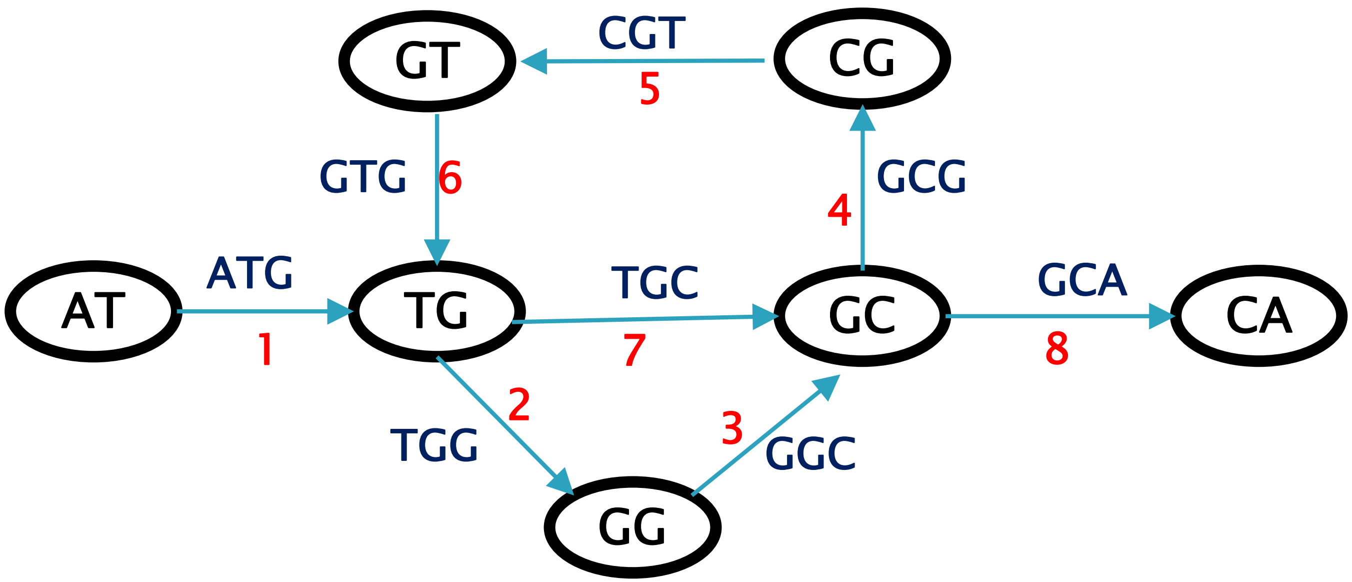
2-secvențe prefixe, sufixe distincte: AT, TG, GC, GT, GG, CA, CG



Shortest superstring problem

S = {ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT}

2-secvențe prefixe, sufixe distincte: AT, TG, GC, GT, GG, CA, CG

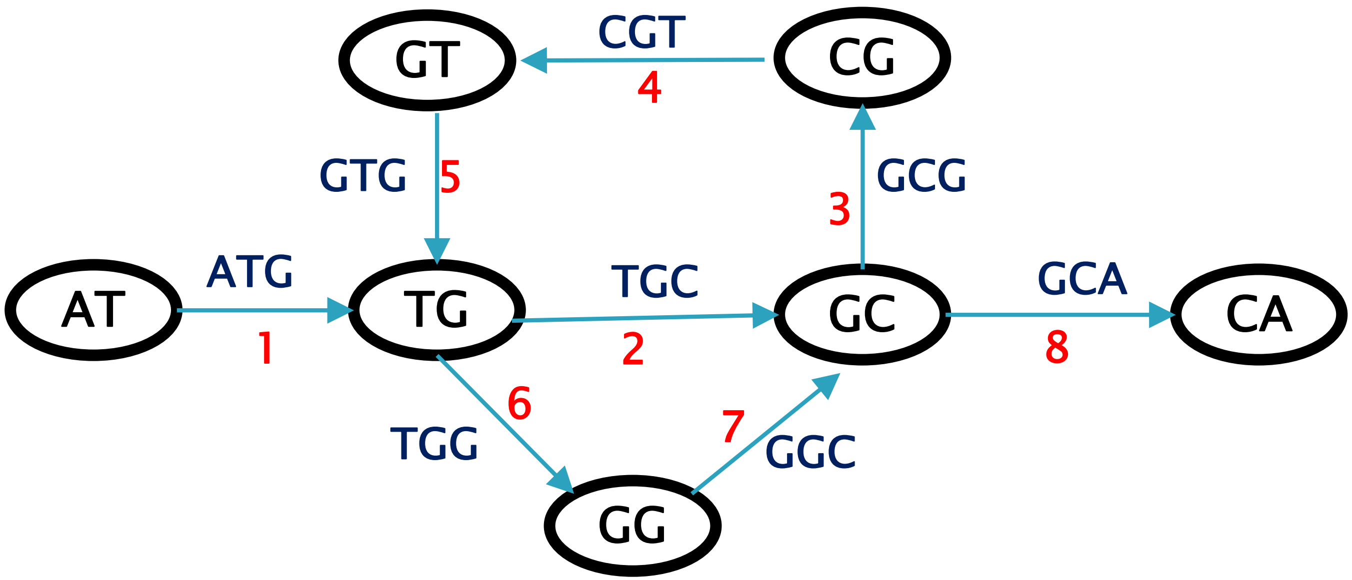


Drum **EULERIAN**: ATG, TGG, GGC, GCG, CGT, GTG, TGC, GCA=>
Superstring =ATGGCGTGCA

Shortest superstring problem

S = {ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT}

2-secvențe prefixe, sufixe distincte: AT, TG, GC, GT, GG, CA, CG



Drum **EULERIAN 2**: ATG, TGG, GGC, GCG, CGT, GTG, TGC, GCA=>
Superstring =ATGGCGTGCA

Descompuneri euleriene în lanțuri

Descompuneri euleriene în lanțuri

- **k-descompunere euleriană în lanțuri** a unui graf G =

  o mulțime de k lanțuri simple, muchie-disjuncte

  $\Delta = \{P_1, P_2, \ldots, P_k\}$

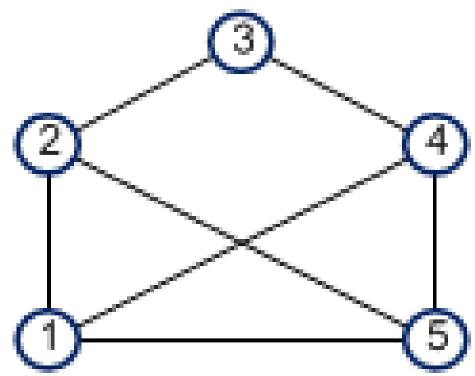
  ale căror muchii induc o k-partiție a lui E(G):

  $E(G) = E(P_1) \cup E(P_2) \cup \ldots \cup E(P_k)$

Descompuneri euleriene în lanțuri

- **Interpretare**

  De câte ori (minim) trebuie să ridicăm creionul de pe hârtie pentru a desena diagrama?



Descompuneri euleriene în lanțuri

Teoremă – Descompunere euleriană

Fie G=(V, E) un graf orientat, conex (= graful neorientat asociat este conex), cu **exact 2k vârfuri de grad impar** (k>0). Atunci există o k-descompunere euleriană a lui G și k este cel mai mic cu această proprietate.