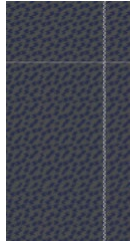


IDENTIFICAREA AUTOMATELOR

S.I. dr. Ing. Vlad-Cristian Miclea

Universitatea Tehnica din Cluj-Napoca
Departamentul Calculatoare



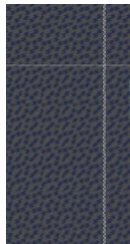
CUPRINS

- 1) Introducere
- 2) Identificarea automatelor
- 3) Procesul identificării automatelor
 - 1) Aducerea într-o stare cunoscută
 - 2) Verificarea starilor
 - 3) Verificarea tranzițiilor
- 4) Concluzii



PLAN CURS

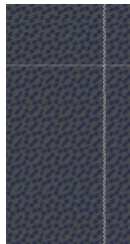
- Partea 1 – VHDL
 1. Limbajul VHDL – 1
 2. Limbajul VHDL – 2
 3. Limbajul VHDL – 3
- Partea 2 – Implementarea sistemelor numerice
 4. Microprogramare
 5. Partea 1 - Unitate de comanda – exemplu cuptor
 5. Partea 2 - Unitate de executie – exemplu cuptor
- Partea 3 – Automate
 6. Automate finite
 7. Stari
 8. Automate sincrone
 9. Automate asincrone
 - 10. Identificarea automatelor**
 11. Automate fara pierderi
 12. Automate liniare
- Partea 4 – Probleme si discutii



CONTEXT

Cursurile trecute

- Automate finite
 - Abstractizarea circuitelor secventiale
 - Clasificarea automatelor (Moore, Mealy)
- Stari ale automatelor
 - Reducerea numarului de stari
 - Codificarea eficienta a starilor
- Automate sincrone
 - Metode de eficientizare
- Automate asincrone
 - Analiza si sinteza automatelor asincrone
 - Curse Critice



IDENTIFICAREA AUTOMATELOR

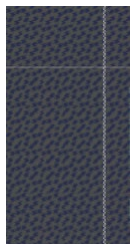
Introducere

- **Scop:** determinarea funcționării automatelor
- Există fizic automatul și tabelul lui de tranziții (black box)
- Verificăm:
 - Funcționarea corectă
 - Respectarea tabelului de tranziții
- Experimental:
 - Se aplică automatului secvențe de intrări
 - Se observă ieșirile
- Se determină dacă tranzițiile automatului sunt corecte
- Rezultatul verificării asigura (sau nu) ca automatul este: **redus, complet determinat** (specificat) și **conex** (orice stare este accesibilă din oricare altă stare)



IDENTIFICAREA AUTOMATELOR

- Procesul de identificare are **3 etape**:
 - 1. Aducerea automatului într-o stare cunoscută
 - Se realizează prin:
 - Aplicarea unei secvențe de sincronizare (dacă există!)
 - Aplicarea unei secvențe de inițializare (există întotdeauna)
 - 2. Verificarea stărilor automatului
 - Se realizează prin:
 - Aplicarea repetată a unor secvențe de distingere (dacă există!) pentru determinarea tuturor stărilor automatului
 - 3. Verificarea propriu-zisă a tranzițiilor



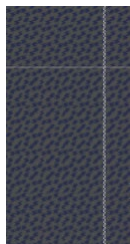
PROCESUL DE IDENTIFICARE

1. Aducerea automatului într-o stare cunoscută

- **Exemplu:** se dă automatul redus, complet specificat și conex cu tabelul de tranziții:

f, g \ x	0	1
A	A,1	D,0
B	A,0	A,1
C	C,0	B,0
D	D,1	C,1

Se observa ca se poate ajunge in orice stare (A, B, C, D), deci putem identifica automatul.



PROCESUL DE IDENTIFICARE

1. Aducerea automatului într-o stare cunoscută

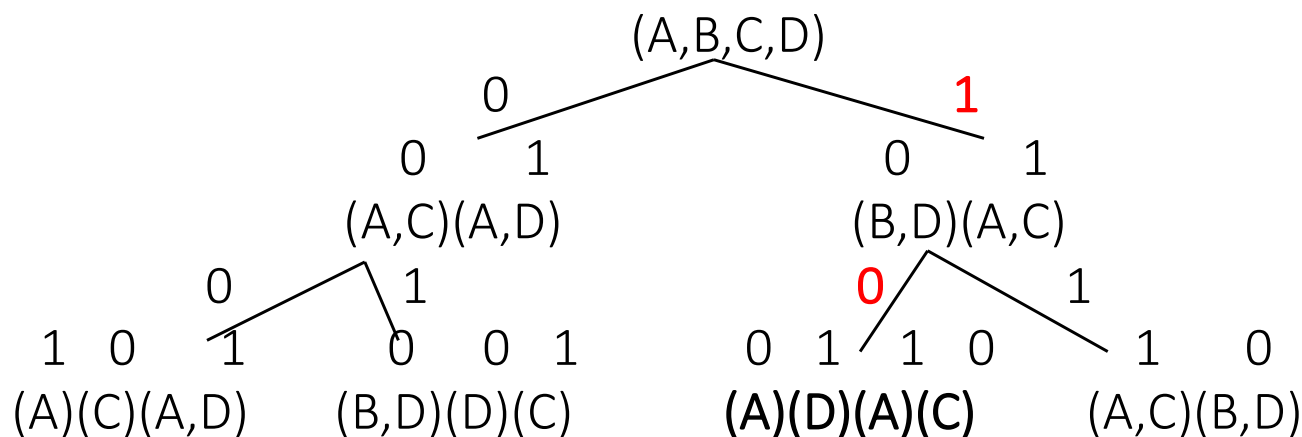
■ Secvența de inițializare SI

- Determinarea se face cu ajutorul unui **arbore de inițializare** (AI)
- Arborele de inițializare:
 - Conține în noduri grupuri de stări
 - Ramificațiile din noduri corespund numărului intrărilor automatului
 - Nodul rădăcină conține toate stările
 - Un nod de pe nivelul k nu se mai dezvoltă dacă este identic cu un nod de pe un nivel anterior
 - Alg se consideră încheiat când pentru prima dată un nod din arbore conține stări singulare (eventual duplicate)
 - Calea de la nodul rădăcină la nodul cu stări singulare, obținută prin concatenarea intrărilor corespunzătoare, determină secvența de inițializare SI
- Prin aplicarea SI automatul este adus într-o stare cunoscută

PROCESUL DE IDENTIFICARE

1. Aducerea automatului într-o stare cunoscută

- Secvența de inițializare SI
- Exemplu: arborele de inițializare este:



- Secvența de inițializare este SI = 10

PROCESUL DE IDENTIFICARE

1. Aducerea automatului într-o stare cunoscută

■ Secvența de inițializare SI

■ Observații:

- Orice automat are o secvență de inițializare
- Dacă rezultă mai multe secvențe de inițializare se alege cea mai scurtă

■ Interpretarea SI:

Stare inițială necunoscută	Răspunsul la SI = 10 (ieșiri)	Stare finală cunoscută
A	01	D
B	11	A
C	00	A
D	10	C



PROCESUL DE IDENTIFICARE

1. Aducerea automatului într-o stare cunoscută

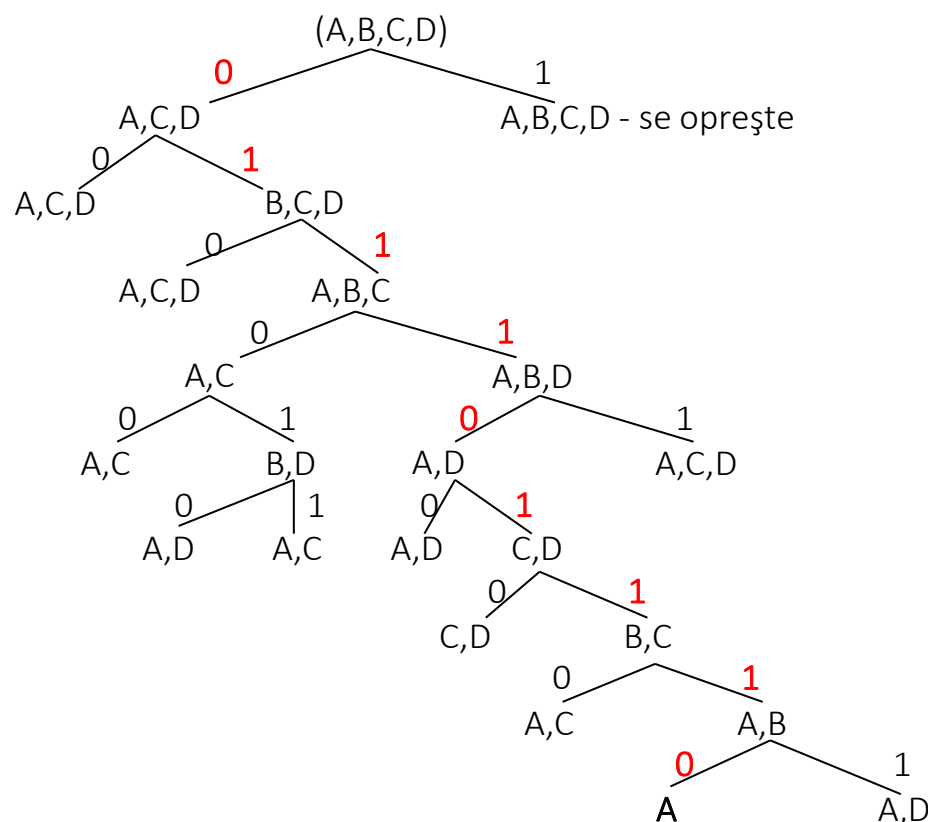
■ Secvența de sincronizare SS

- Determinarea se face cu ajutorul unui **arbore de sincronizare** (AS)
- Arborele de sincronizare:
 - Conține în noduri grupuri de stări
 - Nu ne interesează ieșirile automatului
 - Ramificațiile din noduri corespund numărului intrărilor automatului
 - Nodul rădăcină conține toate stările
 - Stările care apar duplicate în noduri se scriu o singură dată
 - Un nod de pe nivelul k nu se mai dezvoltă dacă este identic cu un nod de pe un nivel anterior
 - Dacă într-un nivel j apar mai multe noduri cu aceeași informație se continuă dezvoltarea doar a unuia dintre aceste noduri
 - AS se consideră încheiat când pentru prima dată un nod din arbore conține o stare singulară
 - Calea de la nodul rădăcină la nodul cu stare singulară, obținută prin concatenarea intrărilor corespunzătoare, determină secvența de sincronizare SS
- Indiferent de starea în care se află automatul, prin aplicarea SS se ajunge într-o stare finală cunoscută, unică

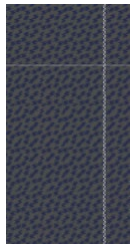
PROCESUL DE IDENTIFICARE

1. Aducerea automatului într-o stare cunoscută

- Secvența de sincronizare SS
- Exemplu: arborele de sincronizare este:



- Secvența de sincronizare este $SS = 011101110$; starea finală este A



PROCESUL DE IDENTIFICARE

2. Verificarea stărilor automatului

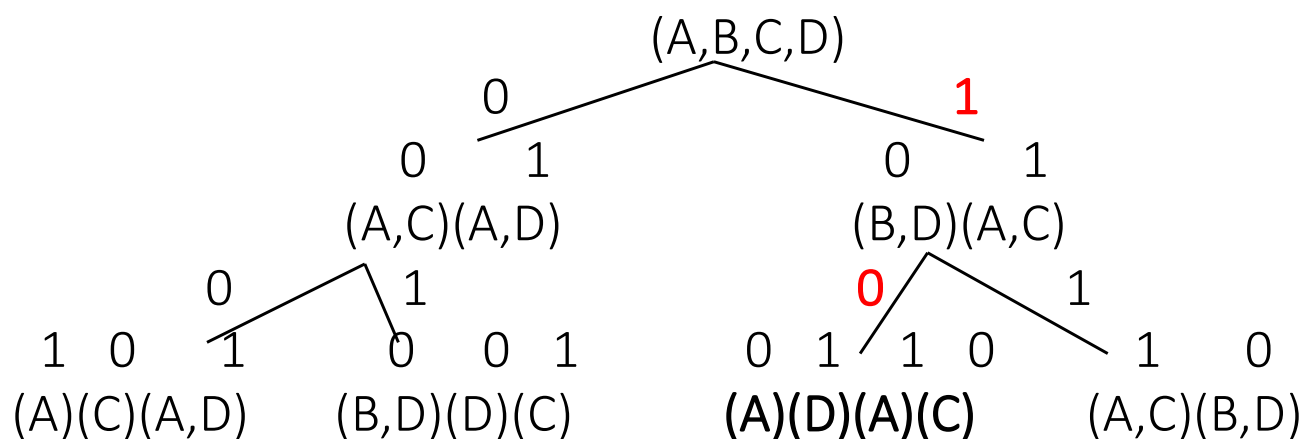
■ Secvența de distingere SD

- Determinarea se face cu ajutorul unui **arbore de distingere** (AD)
- Arborele de distingere (AD) are structură asemănătoare cu cel de inițializare (AI), cu diferențele:
 - Dacă un nod conține stări duplicate, el nu se mai dezvoltă
 - AD se consideră încheiat când pentru prima dată un nod din arbore conține stări singulare
 - Calea de la nodul rădăcină la nodul cu stări singulare, obținută prin concatenarea intrărilor corespunzătoare, determină secvența de distingere SD
- **Observații:**
 - Nu orice automat are secvență de distingere
 - Se poate întâmpla ca AI să fie un subarbore al AD

PROCESUL DE IDENTIFICARE

2. Verificarea stărilor automatului

- Secvența de distingere SD
- Exemplu: arborele de distingere este:



- Secvența de distingere SD = 10 și coincide cu secvența de inițializare

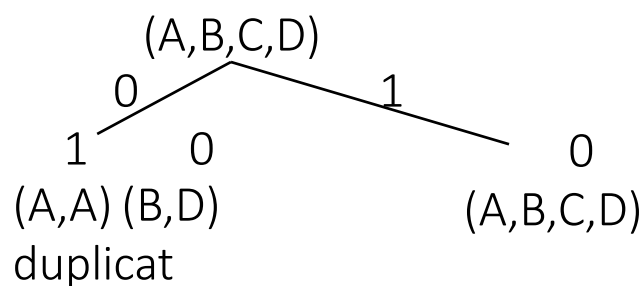
PROCESUL DE IDENTIFICARE

2. Verificarea stărilor automatului

■ Secvența de distingere SD

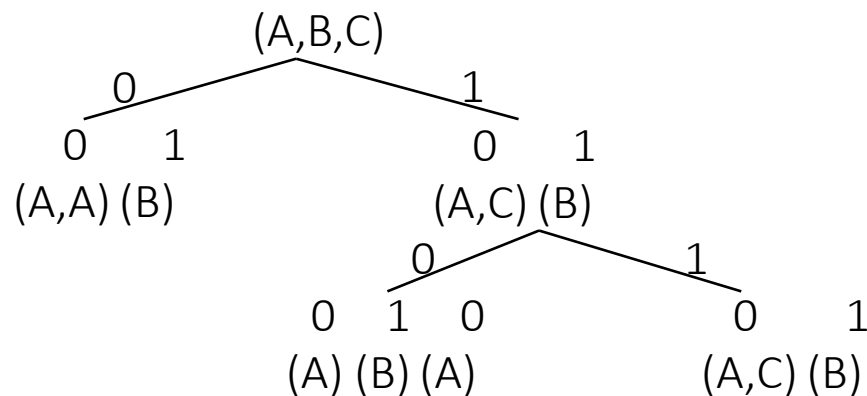
■ Exemplu 1: Automat fără secvență de distingere

	0	1
A	B,0	D,0
B	A,1	C,0
C	D,0	A,0
D	A,1	B,0



■ Exemplu 2: Automat cu SI diferit de SD

	0	1
A	A,0	C,0
B	A,0	B,1
C	B,1	A,0



■ SI = 0 și SD = 10

PROCESUL DE IDENTIFICARE

2. Verificarea stărilor automatului

■ Identificarea stărilor

	SS= 011101110	SD=10	SD=10	SD=10	111	SD=10
Stare inițială	X	A	D	C	A	B
Răspuns (ieșiri)	-	01	10	00	010	11
Stare finală	A	D	C	A	B	A
Concluzie	Faza de inițializare	A este identificată	D este identificată	C este identificată		B este identificată

- În cazul în care nu există SS, se utilizează SI pentru faza de inițializare
- Deoarece aplicând SD nu se poate ajunge în starea B, se aplică intrările 111, care generează ieșirile 010



PROCESUL DE IDENTIFICARE

3. Verificarea tranzițiilor

- Trebuie verificată fiecare tranziție din tabelul de tranziții al automatului
- Verificarea se face prin aplicări repetate de intrări și eventual de secvențe de distingere (dacă există)
- Pentru o succesiune de intrări aplicate automatului trebuie să rezulte o succesiune corespunzătoare de ieșiri, identică cu cele deduse din tabelul de tranziții

PROCESUL DE IDENTIFICARE

3. Verificarea tranzițiilor

■ Exemplu:

- Dacă se aplică pe intrări SS = 011101110 automatul va ajunge în starea A

SS	0	SD	0	SD	0	SD	1
X → A → A → D → D → C → C → A → D →							
	1	01	1	10	0	00	0

- Dacă în A se aplică 0 pe intrare și se obține 1 pe ieșire, conform tabelului de tranziții automatul va rămâne în starea A
- Dacă se aplică o secvență de distingere în A și pe ieșire se obține 01 știm că se ajunge în D
- Dacă în D se aplică 0 pe intrare și se obține 1 pe ieșire, conform tabelului de tranziții automatul va rămâne în starea D
- Dacă se aplică o secvență de distingere în D și pe ieșire se obține 10 știm că se ajunge în C
- Etc.

PROCESUL DE IDENTIFICARE

3. Verificarea tranzițiilor

- Exemplu: dacă nu se folosește secvența de distingere SD

- Dacă se aplică pe intrări SS = 011101110 automatul va ajunge în starea A

- Apoi se aplică intrări astfel încât să se verifice toate tranzițiile posibile (asta înseamnă că din fiecare stare trebuie să se aplice și intrare 0 și intrare 1)

SS	0	1	0	1	0	1	0	1	1	1	1													
X	→	A	→	A	→	D	→	D	→	C	→	C	→	B	→	A	→	D	→	C	→	B	→	A
		1		0		1		1		1		0		0		0		0		1		0		1

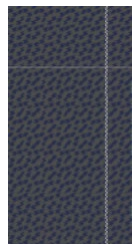
- Dacă după ce ajungem în A se aplică pe intrare secvența 01010101111 și se obține pe ieșire secvența 10110000101, atunci toate tranzițiile posibile ale automatului au fost verificate și sunt corecte



PROCESUL DE IDENTIFICARE

Alte metode de identificare a automatelor

- Automatele au memorie \Rightarrow evoluția lor poate depinde de evenimente trecute
- **Definiție:** Deschiderea (distanța) memoriei reprezintă cantitatea de informație determinată de intrările și ieșirile trecute, care determină comportarea viitoare a automatului
- Dacă se cunoaște starea inițială și secvența de intrări aplicate trebuie să se determine în mod unic starea finală și secvența de ieșiri rezultată
- Identificarea automatelor în funcție de deschiderea memoriei se face în 3 situații:
 - În raport cu secvența de intrare / ieșire
 - În raport cu secvența de intrare
 - În raport cu secvența de ieșire



CONCLUZII

Identificarea automatelor

- Generalitati
- Procesul identificarii automatelor
 - Aducerea intr-o stare cunoscuta
 - Verificarea starilor
 - Verificarea tranzitiilor
- Data viitoare – automate fara pierderi