

## Specifica – parte IIA

Leggere Sez. 5.5.1, 5.5.2, 5.5.3  
Ghezzi et al.



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

## Specifiche operazionali

- Diagrammi di flusso di dati
- Diagrammi UML
- Macchine a stati finiti

Specifica 2A



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

2

## Diagrammi di flusso di dati

- Linguaggio semiformale
- Caratteristiche:
  - Notazione funzionale nella descrizione delle trasformazioni dei dati
  - Applicazione sistematica del concetto di raffinamento della specifica mediante scomposizione gerarchica di un'operazione in un insieme di operazioni più semplici
  - Adatti a sistemi informativi

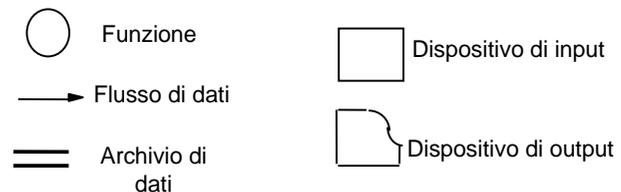
Specifica 2A



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

3

## DFD: elementi di base



- Frecche entranti in (uscanti da) una funzione: valori di input (output)
- Frecche entranti in (uscanti da) un archivio: dati inseriti nel (estratti dall') archivio
- Frecche uscenti da un dispositivo di input (entranti in un dispositivo di output): dati ricevuti (prodotti) nell'interazione con l'utente

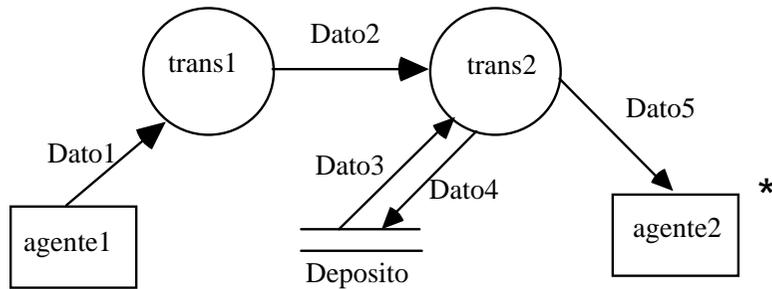
Specifica 2A



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

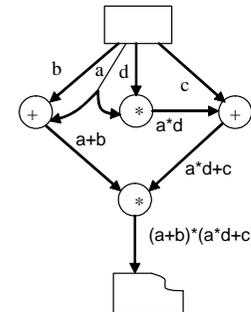
4

## Esempio



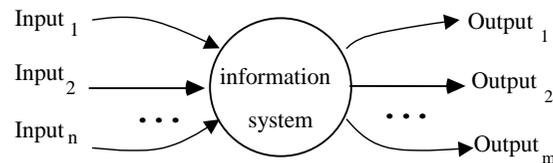
\* Alcuni autori non distinguono fra dispositivi di input e output

## DFD: esempio



- Gli input sono b,a,d,c
- La freccia uscente da una funzione trasporta l'output della funzione applicata ai suoi input
- Il risultato in output è  $(a+b)*(a*d+c)$

## Diagramma iniziale e raffinamenti

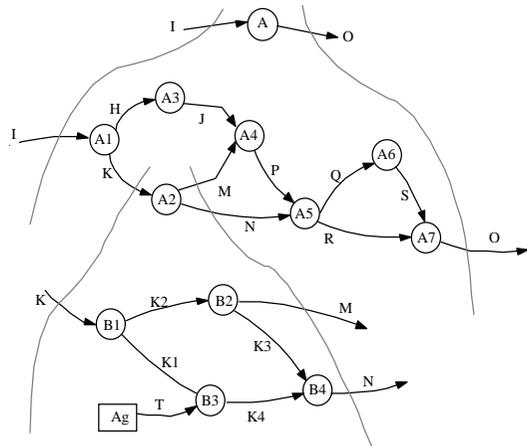


- Si parte da un'unica funzione che elabora gli input e produce gli output dell'intero sistema
- Si arriva alla specifica finale "esplodendo" le funzioni in interi DFD fino al livello di dettaglio desiderato

## Raffinamenti

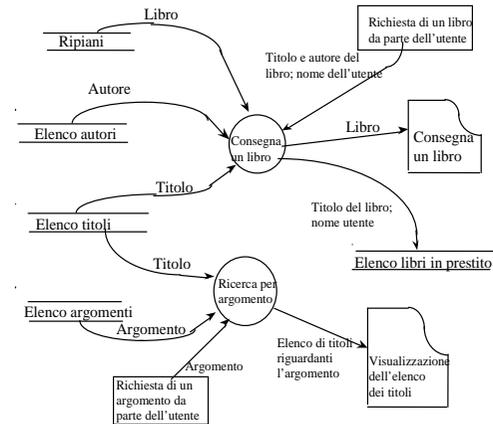
- Vincolo di *continuità del flusso informativo*: il DFD in cui si espone la funzione deve presentare gli stessi flussi netti di dati
- Si possono introdurre nel nuovo DFD nuovi agenti o depositi di dati, purché non esistano flussi di dati fra questi e le funzioni esterne al sistema

# Raffinamenti



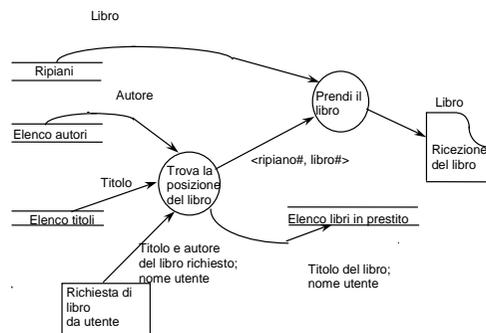
Specifica 2A

# Sistema informativo semplificato per biblioteche



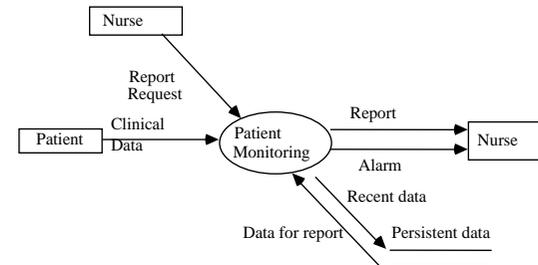
Specifica 2A

# Raffinamento della funzione "consegna un libro"



Specifica 2A

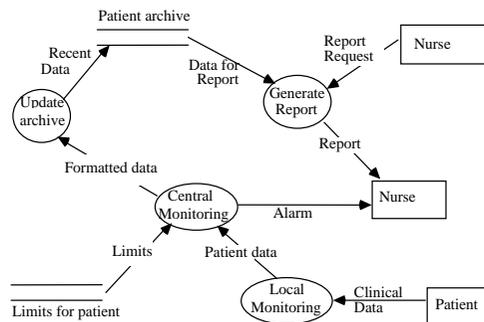
# Esempio: monitoraggio di pazienti



- DFD iniziale: la funzione *Patient Monitoring* individua l'intero sistema da specificare

Specifica 2A

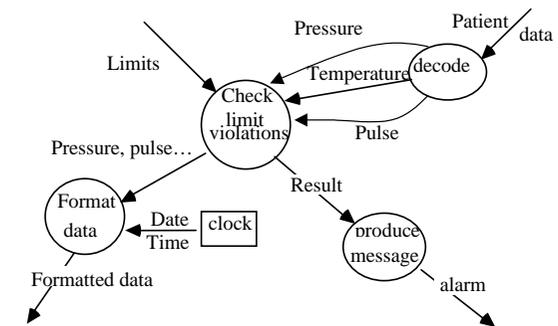
## Raffinamento di *Patient Monitoring*



Specifica 2A

13

## Ulteriore raffinamento: *Central Monitoring*



Specifica 2A

14

## Stesura di DFD

- Il DFD rappresenta uno stato stabile, “a regime”, in cui l’uscita dipende dall’ingresso e si ignorano le procedure per
  - Inizializzazione
  - Terminazione
  - Gestione delle eccezioni

Specifica 2A

15

## Stesura di DFD

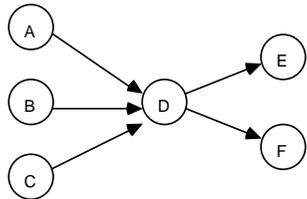
- Ignorare, perché non significativi
  - Flusso di controllo
  - Sincronizzazione fra processi
- Evidenziare entrate e uscite nette
- Assegnare a flussi e funzioni nomi significativi

Specifica 2A

16

## Limiti dei DFD

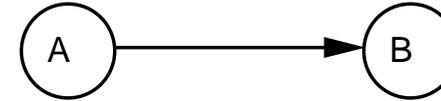
- Ambiguità intrinseca
- Esempio:



- Gli output di A, B, C sono tutti necessari?
- Gli input di E e F sono prodotti tutti? Con lo stesso valore? Nello stesso istante?

## Limiti dei DFD

- Assenza di informazioni di controllo



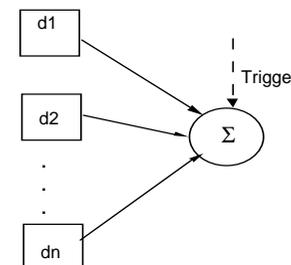
- Possibili interpretazioni:
  - A produce il dato e attende che B lo consumi
  - B può leggere il dato molte volte senza consumarlo
  - A e B comunicano attraverso una pipe

## Valutazione dei DFD

- Semplici e intuitivi
- Semantica non definita (notazione semiformale):
  - Non sufficienti come specifica di progetto
  - Non utilizzabili come modello per un simulatore

## Superamento dei limiti dei DFD

- Integrare la specifica con altre notazioni formali e adatte a specificare il controllo
- Estendere i DFD. Esempio:



Il calcolo è eseguito su tutti gli input quando il trigger presenta un segnale

## Superamento dei limiti dei DFD

- Formalizzazione
  - Aspetti legati a produzione e consumo di dati
  - Necessità o meno di tutti gli input e modalità di produzione degli output (annotazioni)
  - Specifica formale del calcolo eseguito da una funzione

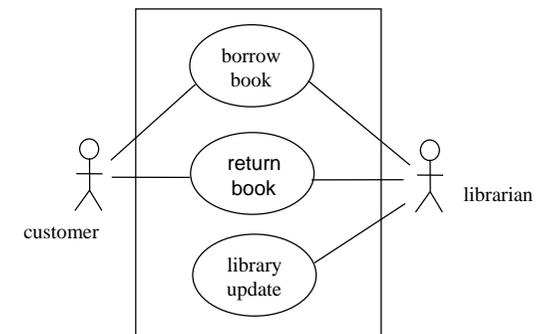
## Diagrammi UML

- Alcuni diagrammi semplici e intuitivi per modellare gli aspetti dinamici di un sistema e il comportamento degli attori
  - Use case diagram
  - Sequence diagram
  - Collaboration diagram

## Use case diagram

- Descrivono le *funzioni del sistema* dal punto di vista degli *attori*.
- Sistema: rettangolo
- Attore: uomo stilizzato
- Caso d'uso: ellisse, unita da tratti agli attori interessati

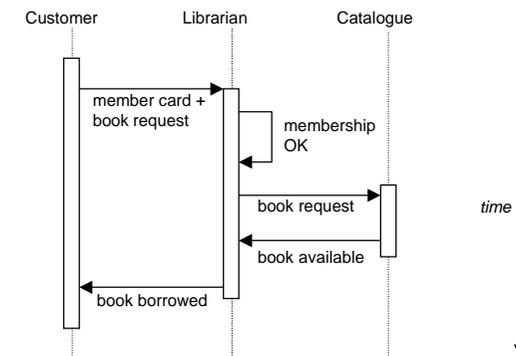
## Esempio



## Sequence diagram

- Rappresentano l'interazione fra oggetti come scambio di messaggi
- Indicano la progressione del tempo dall'alto verso il basso
- Oggetti: linee verticali tratteggiate
- Azioni: (in conseguenza di messaggi) rettangoli sopra gli oggetti

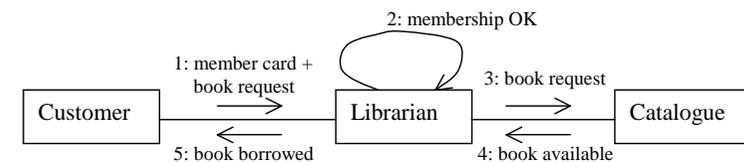
## Esempio



## Collaboration diagram

- Rappresentano le interazioni fra oggetti e il loro ordine
- Equivalgono ai sequence diagrams
- Oggetti: rettangoli (uniti da tratti a indicare interazione)
- Messaggi: frecce con etichette (numerate a indicare l'ordine)

## Esempio



## Modelli basati su macchine astratte

- Si specifica il sistema da modellare (o parte di esso) con una macchina astratta, la cui esecuzione simula il comportamento del sistema
- Definite da
  - *Stati* in cui si può trovare il sistema
  - *Transizioni* da uno stato a un altro
- Automi a stati finiti, reti di Petri

## Automi a stati finiti

- FSM (Finite State Machine, macchine a stati finiti) o DFA (Deterministic Finite Automaton/a)
- Modello semplice e molto conosciuto
- Adatti a modellare sistemi:
  - che possono trovarsi in ogni istante in uno di un insieme finito di stati
  - in cui le variazioni di stato si hanno in seguito a eventi, modellabili con un insieme finito di simboli

## FSM: definizione

Una macchina a stati finiti è definita da:

- Un insieme finito  $Q$  di *stati*;
- Un insieme finito  $I$  di *input*;
- Una *funzione di transizione*

$$\delta: Q \times I \rightarrow Q$$

- $\delta$  può essere una funzione parziale (cioè non definita per tutti gli elementi di  $Q \times I$ )

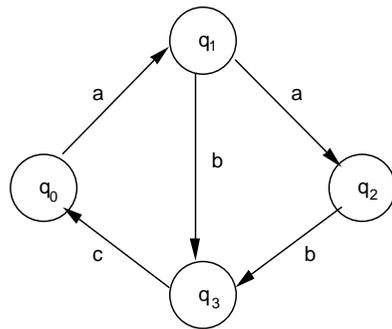
## Rappresentazione grafica

Una FSM si rappresenta con un grafo diretto in cui:

- Ogni stato è rappresentato da un nodo
- Un arco da un nodo  $q_j$  a un nodo  $q_k$  etichettato con un input  $i$  indica che

$$\delta(q_j, i) = q_k$$

## Esempio



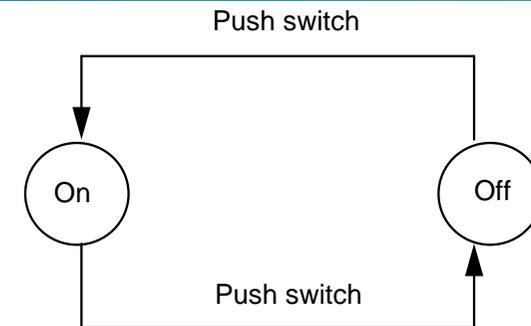
- $Q = \{q_0, q_1, q_2, q_3\}$
- $I = \{a, b, c\}$
- $\delta$ :

|       | a     | b     | c     |
|-------|-------|-------|-------|
| $q_0$ | $q_1$ | -     | -     |
| $q_1$ | $q_2$ | $q_3$ | -     |
| $q_2$ | -     | $q_3$ | -     |
| $q_3$ | -     | -     | $q_0$ |

Specifica 2A

33

## Esempio: lampada

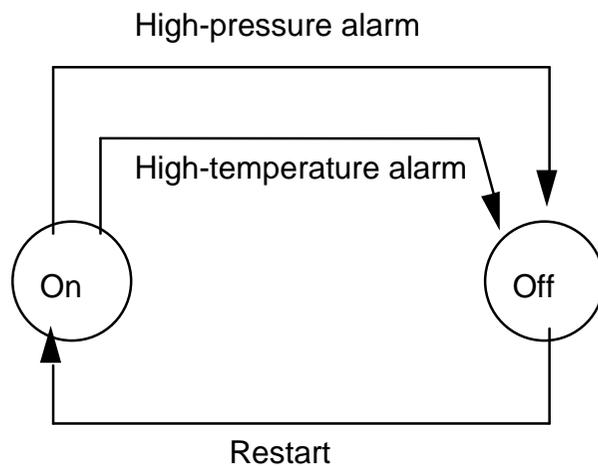


- $Q = \{\text{On}, \text{Off}\}$
- $I = \{\text{Push switch}\}$

Specifica 2A

34

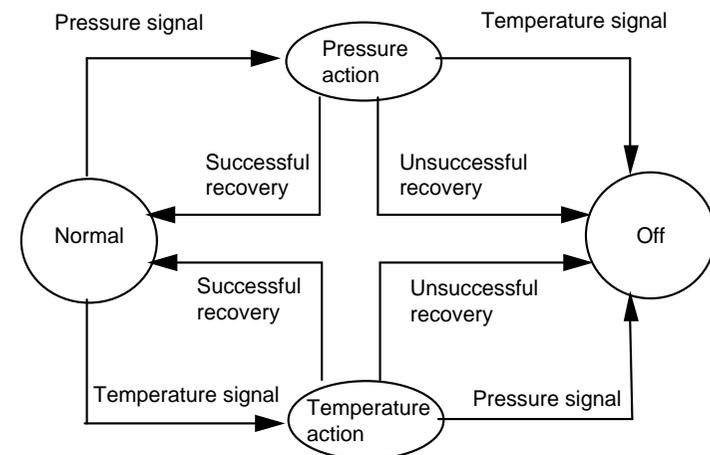
## Esempio: controllo di un impianto



Specifica 2A

35

## Esempio: controllo di un impianto

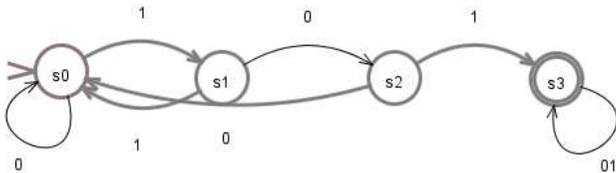


Specifica 2A

36

## Simulatore di FSM

- Visual Automata Simulator (VAS)
- <http://www.cs.usfca.edu/~jbovet/vas.html>
- Simula anche macchine di Turing
- Esempio: riconoscitore di sequenza 101



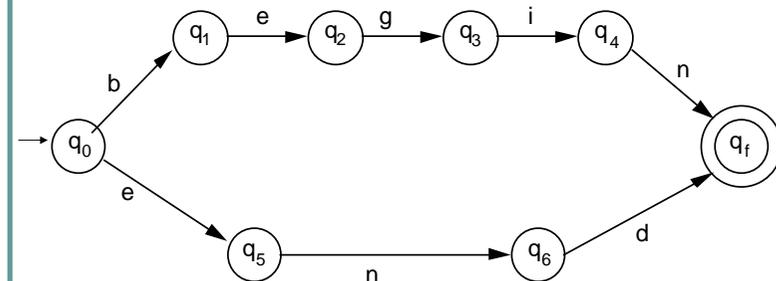
## Applicazioni delle FSM

- Sistemi di controllo
- Compilazione
- Riconoscimento di sequenze
- Protocolli
- Hardware

## FSM riconoscitrici

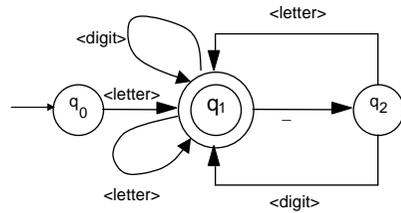
- Specificano sequenze accettabili di segnali di input
- In generale, specificano l'appartenenza di una *stringa* a un *linguaggio (regolare)*.
- Si specifica uno *stato iniziale* (freccia entrante) e uno o più *stati finali* (doppio cerchio).
- La sequenza è accettabile se porta dallo stato iniziale a uno degli stati finali.

## Esempio



- Riconosce le parole chiave “begin” e “end”
- La funzione di transizione è parziale

## Esempio: identificatori di un linguaggio di programmazione

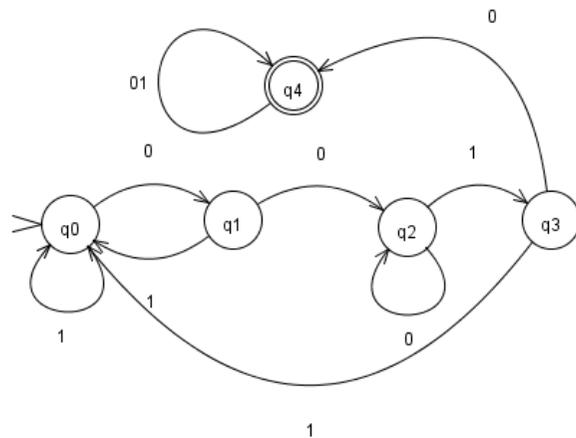


Legend:   
 <letter> is an abbreviation for a set of arrows labeled a, b, ..., z, A, ..., Z,   
 <digit> is an abbreviation for a set of arrows labeled 0, 1, ..., 9, respectively

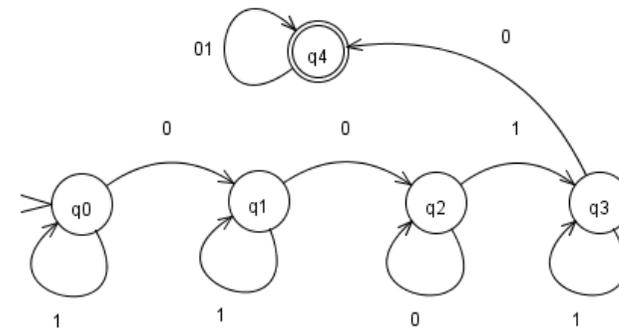
## Esercizi

- Progettare una FSM che:
  - Abbia come insieme di input  $\{0,1\}$ .
  - Riconosca le sequenze di input che contengono la stringa 0010.
- Progettare una FSM che:
  - Abbia come insieme di input  $\{0,1\}$ .
  - Riconosca le sequenze di input in cui compaiono in ordine i simboli 0 0 1 0, anche non consecutivamente.

## Soluzione es. 1



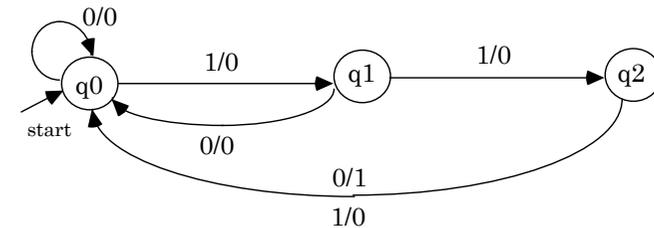
## Soluzione es. 2



## FSM con output

- Si possono arricchire le FSM con *segnali di output* da un insieme finito  $O$ , corrispondenti a transizioni di stato.
- Per queste FSM  
 $\delta: Q \times I \rightarrow Q \times O$
- Le transizioni sono etichettate con  $\langle \text{valore di input} \rangle / \langle \text{valore di output} \rangle$

## FSM con output: esempio



- Combinazione di una cassaforte: produce 1 in output quando riconosce la combinazione 110

## DFA non deterministici (NFA)

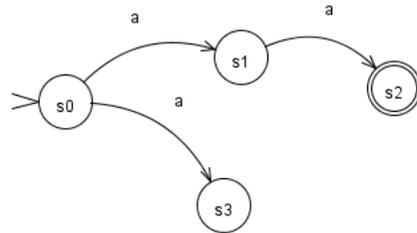
- Uno stesso input può portare a più transizioni.
- La funzione di transizione ha valori non in  $Q$ , ma in  $2^Q$  (*insieme delle parti* o *powerset* di  $Q$ ).
- $\delta: Q \times I \rightarrow 2^Q$ .

## Riconoscimento con NFA

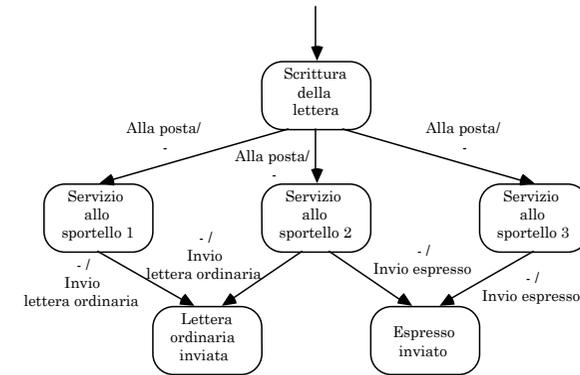
- Un NFA riconosce una stringa se, fra tutte le sequenze di transizioni possibili, almeno una porta dallo stato iniziale a quello finale
- NFA e DFA sono equivalenti: per ogni NFA si può costruire un DFA che riconosce le stesse sequenze.

## Esempio

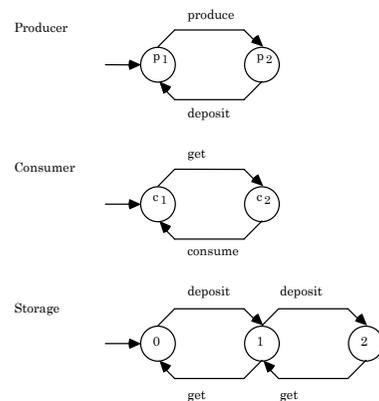
- Alfabeto: a
- Riconosce la stringa "aa"



## NFA: esempio

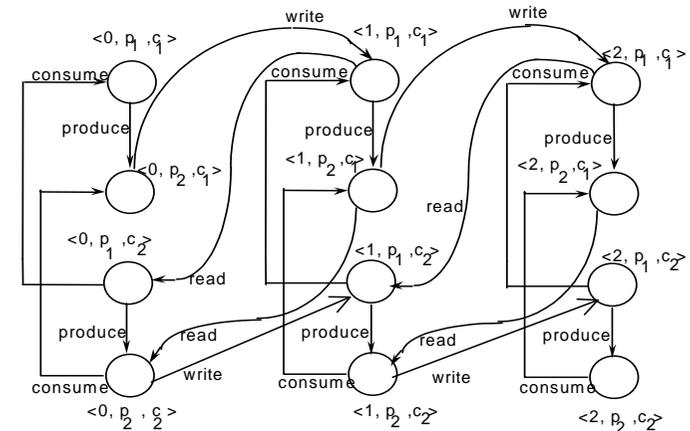


## Esempio: produttore-consumatore



- Deposito con due spazi
- Per modellare l'interazione occorre disegnare un DFA unico.
- Lo spazio degli stati è  $Q_p \times Q_c \times Q_s$

## DFA complessivo



## Limiti dei FA

- Memoria finita: problema se le transizioni dipendono da valori reali (ad es. differenza di temperatura)-> arricchimento del modello con predicati e funzioni
- Esplosione dello spazio degli stati
  - Dati N FSM con  $k_1, k_2, \dots, k_N$  stati, la loro composizione ha  $k_1 * k_2 * \dots * k_N$  stati
- Non modellano transizioni asincrone nello stesso sistema (modelli sincroni)