Algoritmi e Strutture Dati

Elaborato a.a. 2019-20

Prof.ssa Marina Zanella

ESPRESSIONI REGOLARI E AUTOMI

Definizione induttiva di espressione regolare

Dato un alfabeto Σ finito di simboli,

- ε (cioè il simbolo nullo) è una espressione regolare
- Ogni simbolo appartenente a Σ è una espressione regolare
- L'applicazione di un operatore monadico a una espressione regolare è una espressione regolare
- L'applicazione di un operatore diadico a una espressione regolare è una espressione regolare

Operatori diadici

Sono quelli (primitivi) indicati di seguito, unitamente alla loro rappresentazione formale. Date due espressioni regolari r e s come operandi,

- concatenazione: r s
- alternativa: r|s (oppure r + s)

Si noti che ε è l'elemento neutro della concatenazione, cioè $\varepsilon\, r = r\, \varepsilon = r$

Operatori monadici

Data una espressione regolare r come operando, l'unico operatore monadico primitivo è

- ripetizione (zero o più volte): r^*

Gli operatori monadici derivati sono definiti come segue:

- ripetizione (una o più volte): $r^+ = r\ r^*$
- opzionalità: $r? = r|\varepsilon$
  (un'altra rappresentazione formale è [r])

N.B. Sebbene in questa trattazione si usino talvolta gli operatori derivati, l'applicazione software di cui è richiesto lo sviluppo deve produrre espressioni regolari contenenti solo operatori primitivi. Inoltre non è richiesta alcuna «semplificazione» delle espressioni regolari prodotte; tale semplificazione richiederebbe l'applicazione di regole che qui non sono riportate (ad es. $r|r = r$)

Priorità

- Per forzare le precedenze nella valutazione delle espressioni regolari, si possono usare coppie di parentesi tonde
- In assenza di parentesi tonde, la massima precedenza è quella della ripetizione; la concatenazione ha precedenza sull'alternativa

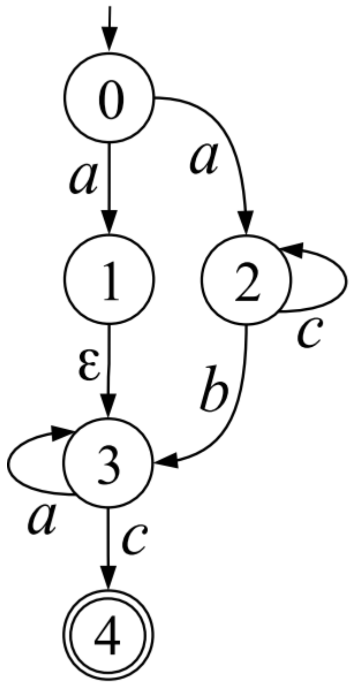
Automa a stati finiti (FA)

- È dotato di un insieme finito (non nullo) di stati è cui uno è lo stato iniziale e un insieme finito (eventualmente nullo) di transizioni
- Si distinguono gli stati di accettazione da quelli che non lo sono (lo stato iniziale può essere di accettazione o meno)
- Se l'insieme degli stati non è un singoletto, non esistono stati isolati
- Esiste una relazione che a ciascuna transizione fa corrispondere un simbolo, eventualmente nullo, cioè ε, di un alfabeto finito
- Lo stesso simbolo (anche nullo) può corrispondere a più transizioni
- Una transizione può uscire da uno stato e rientrare nel medesimo stato (auto-transizione)

Linguaggio accettato da un FA

- Ogni FA accetta un linguaggio regolare, che è un insieme i cui elementi sono tutte le stringhe sull'alfabeto considerato che portano dallo stato iniziale a uno dei suoi stati di accettazione
- Tale linguaggio è descritto da una espressione regolare sull'alfabeto finito dei simboli associati alle sue transizioni
- Il medesimo linguaggio può essere descritto da più espressioni regolari equivalenti; ad esempio, b(c|d) è equivalente a bc|bd

Esempio



$$a\,a^{*}c \mid a\,c^{*}b\,a^{*}c$$

$$=$$

$$a\left(a^{*} \mid c^{*}b\,a^{*}\right)c$$

$$=$$

$$a\left(c^{*}b\right)?\,a^{*}c$$

L'alfabeto dei simboli del FA $\mathcal{N}$ in figura (a sinistra) è $\Sigma = \{a,b,c\}$. L'insieme degli stati di accettazione è $\{4\}$. A destra sono indicate tre espressioni regolari (reciprocamente equivalenti) che definiscono il linguaggio accettato da $\mathcal{N}$

Linguaggio accettato da un FA

Lo pseudocodice dell'algoritmo che, dato un FA (N_{in}), costruisce un'espressione regolare corrispondente al linguaggio da esso accettato è fornito nelle prossime pagine, dove

- per ε-transizione si intende una transizione a cui è associato il simbolo nullo
- una transizione da uno stato n a uno stato n' contraddistinta dall'espressione regolare r è indicata come $\langle n, r, n' \rangle$
- vale la precondizione che l'insieme degli stati di accettazione di N_{in} non sia vuoto

EspressioneRegolare(N_{in})

if nello stato iniziale β_0 entra una transizione **then**
  creare un nuovo stato iniziale n_0 e una ε-transizione da n_0 a β_0
else
  indicare come n_0 lo stato iniziale
end if
if $\exists$ più stati di accettazione o $\exists$ una transizione uscente dall'unico stato di accettazione **then**
  creare lo stato finale n_q
  for $\forall$ stato di accettazione β_q **do**
    creare una ε-transizione da β_q a n_q
  end for
else
  indicare come stato finale n_q l'unico stato di accettazione
end if
sia $\boxed{\mathcal{N} \text{ l'automa così ottenuto}}$

while $\mathcal{N}$ comprende più di una transizione **do**

  if $\exists$ una sequenza $[\langle n, r_1, n_1\rangle, \langle n_1, r_2, n_2\rangle, \ldots, \boxed{\langle n_{k-1}, r_k, n'\rangle}]$ di transizioni, $k \geq 2$, dove ogni stato n_i, $i \in [1 \ldots (k-1)]$, non ha altre transizioni entranti o uscenti **then**

    sostituire la sequenza con la transizione $\boxed{\langle n, (r_1 r_2 \cdots r_k), n'\rangle}$

  else if $\exists$ un insieme $\{\langle n, r_1, n'\rangle, \langle n, r_2, n'\rangle, \ldots, \langle n, r_k, n'\rangle\}$ di transizioni parallele uscenti dallo stato n e dirette allo stato n' **then**

    sostituire l'insieme di transizioni con la transizione $\boxed{\langle n, (r_1 | r_2 | \cdots | r_k), n'\rangle}$

  else

    sia n uno stato nè iniziale nè finale di $\mathcal{N}$, cioè $n \neq n_0$ e $n \neq n_q$

    for all transizione $\langle n', r', n\rangle$ entrante in n, dove $n' \neq n$ **do**   no cicli

      for all transizione $\langle n, r'', n''\rangle$ uscente da n, dove $n'' \neq n$ **do**

        if $\exists$ una auto-transizione $\langle n, r, n\rangle$ **then**

          inserire una nuova transizione $\boxed{\langle n', (r'(r)^* r''), n''\rangle}$

          se avessi ulteriore stato prima di 0 nel 2° esempio
          0,a(c)*b,3

        else

          inserire una nuova transizione $\boxed{\langle n', (r' r''), n''\rangle}$

        end if

      end for

    end for

    rimuovere n unitamente a tutte le sue transizioni entranti e uscenti

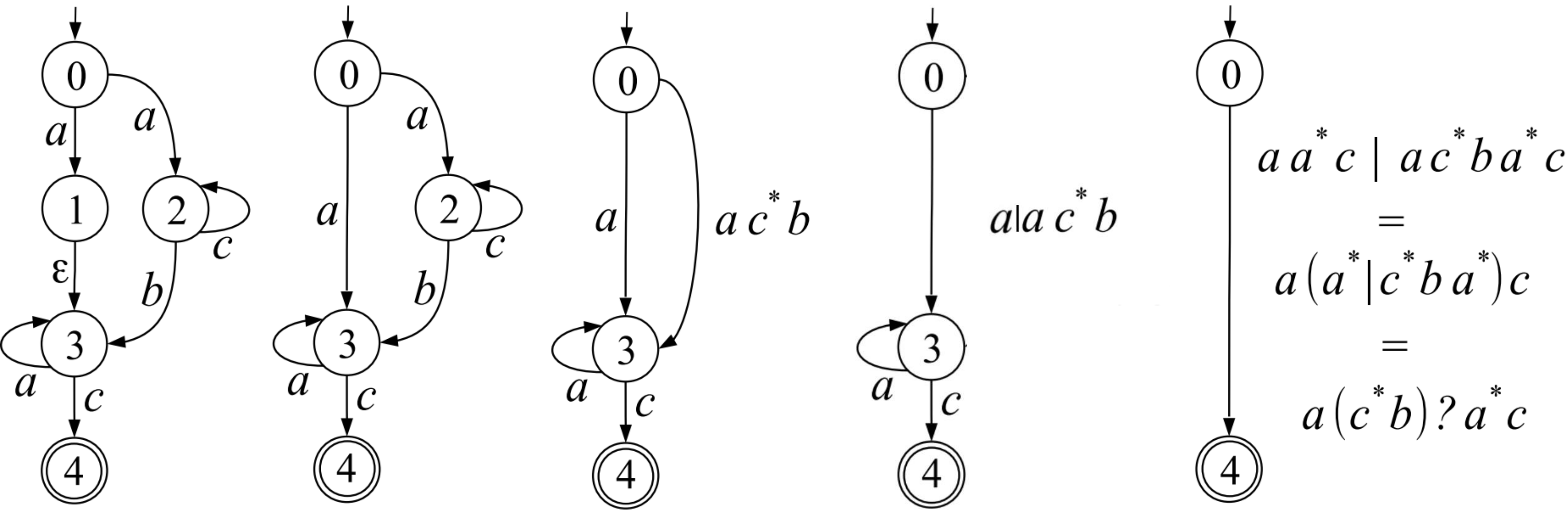
  end if

end while

$\boxed{\textbf{return } R\text{, dove } \langle n_0, R, n_q\rangle \text{ è l'unica transizione in } \mathcal{N}}$

Esempio di applicazione dell’algoritmo EspressioneRegolare(N_{in})

$N_{in} = \mathcal{N}$



qua 0-1-3-4 // 0-2-3-4

Da sinistra verso destra sono illustrate le trasformazioni subite dall’automa di partenza

Più espressioni regolari

- L'algoritmo EspressioneRegolare(N_{in}) costruisce una espressione regolare (unica) che rappresenta tutte le stringhe sull'alfabeto considerato che portano dallo stato iniziale di N_{in} a uno stato di accettazione
- Se invece si intende distinguere il linguaggio accettato da ciascuno stato di accettazione, producendo in uscita tante espressioni regolari quanti sono gli stati di accettazione di N_{in}, si può invocare EspressioneRegolare(N_{in}) una volta per ciascun distinto stato di accettazione, indicando tale stato come se esso fosse l'unico stato di accettazione presente

Più espressioni regolari (cont.)

- Un modo alternativo di produrre tutte le espressioni regolari relative agli stati di accettazione in una sola elaborazione è offerto dall'algoritmo EspressioniRegolari(N_{in}), il cui pseudocodice è fornito nelle pagine successive
- I simboli associati alle transizioni dell'automa in ingresso N_{in} non sono marcati da alcun pedice
- Nelle trasformazioni progressive operate dall'algoritmo sull'automa, esso marca con pedici distinti (uno per ciascuno stato di accettazione di N_{in}) espressioni regolari distinte (perché relative a linguaggi accettati da stati di accettazione diversi)
- Alla fine dell'elaborazione, le espressioni regolari di tutte le transizioni presenti nell'automa conclusivo sono marcate da pedici ed esiste esattamente una transizione per ciascun valore distinto di pedice (cioè per ciascuno stato di accettazione dell'automa di partenza N_{in})

Più espressioni regolari (cont.)

- Nello pseudocodice dell'algoritmo, una transizione da uno stato n a uno stato n' contraddistinta dall'espressione regolare r avente pedice p è indicata come $\langle n, r_{(p)}, n' \rangle$. Si noti che p è l'identificatore di uno stato di accettazione dell'automa di partenza N_{in}
- L'algoritmo produce in uscita un insieme di espressioni regolari, ciascuna contraddistinta da un pedice diverso

EspressioniRegolari(N_{in})

if nello stato iniziale β_0 entra una transizione **then**
    creare un nuovo stato iniziale n_0 e una ε-transizione da n_0 a β_0
else
    indicare come n_0 lo stato iniziale
end if
creare uno stato finale n_q
for $\forall$ stato di accettazione β_q **do**
    creare una ε-transizione da β_q a n_q
end for
sia $\mathcal{N}$ l'automa così ottenuto

while $\mathcal{N}$ comprende più di due stati o più transizioni marcate con lo stesso pedice **do**

  if $\exists$ una sequenza $[\langle n, r_1, n_1\rangle, \langle n_1, r_2, n_2\rangle, \ldots, \langle n_{k-1}, r_k, n'\rangle]$ di transizioni, $k \geq 2$, dove ogni stato n_i, $i \in [1 \ldots (k-1)]$, non ha altre transizioni entranti o uscenti **then**

    if $n' \neq n_q$ e n_{k-1} <u>non</u> è uno stato di accettazione **then**

      sostituire la sequenza con la transizione $\boxed{\langle n, (r_1 r_2 \cdots r_k), n'\rangle}$? pedice terminazione

    else

      sostituire la sequenza con la transizione $\boxed{\langle n, (r_1 r_2 \cdots r_{k-1})_{\underline{(n_{k-1})}}, n'\rangle}$

    end if

  else if $\exists$ una sequenza $[\langle n, r_1, n_1\rangle, \langle n_1, r_2, n_2\rangle, \ldots, \langle n_{k-1}, r_{k\underline{(n_p)}}, n'\rangle]$ di transizioni, $k \geq 2$, dove ogni stato n_i, $i \in [1 \ldots (k-1)]$, non ha altre transizioni entranti o uscenti **then**

    sostituire la sequenza con la transizione $\boxed{\langle n, (r_1 r_2 \cdots r_k)_{(n_p)}, n'\rangle}$

  else if $\exists$ un insieme $\{\langle n, r_1, n'\rangle, \langle n, r_2, n'\rangle, \ldots, \langle n, r_k, n'\rangle\}$ di transizioni parallele uscenti dallo stato n e dirette allo stato n' **then**

    sostituire l'insieme di transizioni con la transizione $\boxed{\langle n, (r_1 | r_2 | \cdots | r_k), n'\rangle}$

  else if $\exists$ un insieme $\{\langle n, r_{1\underline{(n_p)}}, n'\rangle, \langle n, r_{2\underline{(n_p)}}, n'\rangle, \ldots, \langle n, r_{k\underline{(n_p)}}, n'\rangle\}$ di transizioni parallele uscenti dallo stato n e dirette allo stato n', tutte marcate dal pedice n_p **then**

    sostituire l'insieme di transizioni con la transizione $\boxed{\langle n, (r_1 | r_2 | \cdots | r_k)_{(n_p)}, n'\rangle}$

  else

sia n uno stato nè iniziale nè finale di $\mathcal{N}$, cioè $n \neq n_0$ e $n \neq n_q$

for all transizione $\langle n', r', n \rangle$ entrante in n, dove $n' \neq n$ **do**

  for all transizione $\langle n, r'', n'' \rangle$ uscente da n, dove $n'' \neq n$ **do**

    if $n'' = n_q$ e <u>n è uno stato di accettazione</u> **then**

      if $\exists$ una <u>auto-transizione</u> $\langle n, r, n \rangle$ per n **then**

        inserire una nuova transizione $\boxed{\langle n', (r'(r)^*)_{(n)}, n'' \rangle}$

      else

        inserire una nuova transizione $\boxed{\langle n', r'_{(n)}, n'' \rangle}$

      end if

    else if $\exists$ una auto-transizione $\langle n, r, n \rangle$ per n **then**

      inserire una nuova transizione $\boxed{\langle n', (r'(r)^* r''), n'' \rangle}$

    else

      inserire una nuova transizione $\boxed{\langle n', (r' r''), n'' \rangle}$

    end if

  end for

for all transizione $\langle n, r''_{(n_p)}, n'' \rangle$ uscente da n, dove $n'' \neq n$ **do**

  if $\exists$ una auto-transizione $\langle n, r, n \rangle$ per n **then**

    inserire una nuova transizione $\boxed{\langle n', (r'(r)^* r'')_{(n_p)}, n'' \rangle}$

  else

    inserire una nuova transizione $\boxed{\langle n', (r' r'')_{(n_p)}, n'' \rangle}$

  end if

end for

end for

rimuovere n unitamente a tutte le sue transizioni entranti e uscenti

end if

end while

uscita $\leftarrow \emptyset$

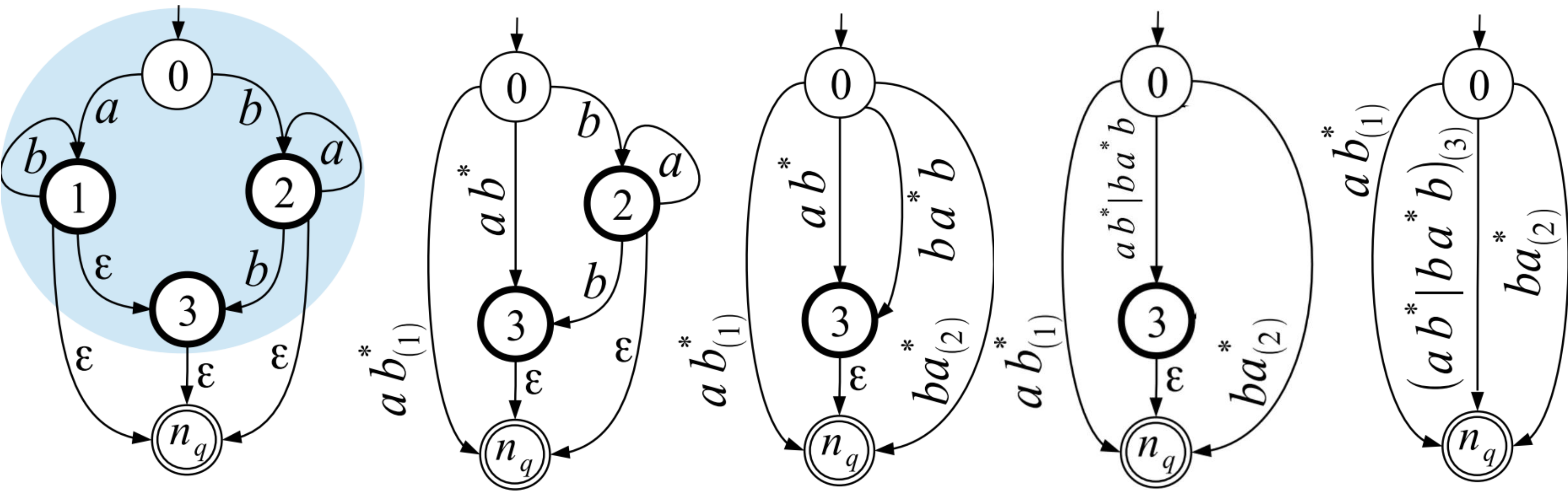
for all transizione $\langle n_0, r_{(n)}, n_{\mathrm{q}} \rangle$ è in $\mathcal{N}$ **do**

  uscita $\leftarrow$ uscita $\cup \{r_{(n)}\}$

end for

return uscita

Esempio di applicazione dell'algoritmo EspressioniRegolari(N_{in})



Su sfondo azzurro è riportato l'automa di partenza (cioè N_{in}), dove gli stati con bordo in grassetto sono quelli di accettazione. Lo stato n_q è quello aggiunto, unitamente alle sue transizioni entranti, nelle righe 6-9 dello pseudocodice. I quattro automi successivi rappresentano le trasformazioni operate dall'algoritmo, sino al raggiungimento della situazione finale, dove le espressioni regolari relative ai tre stati di accettazione (1, 2 e 3) di N_{in} sono R(1) = ab*, R(2) = ba*, R(3) = ab*|ba*b

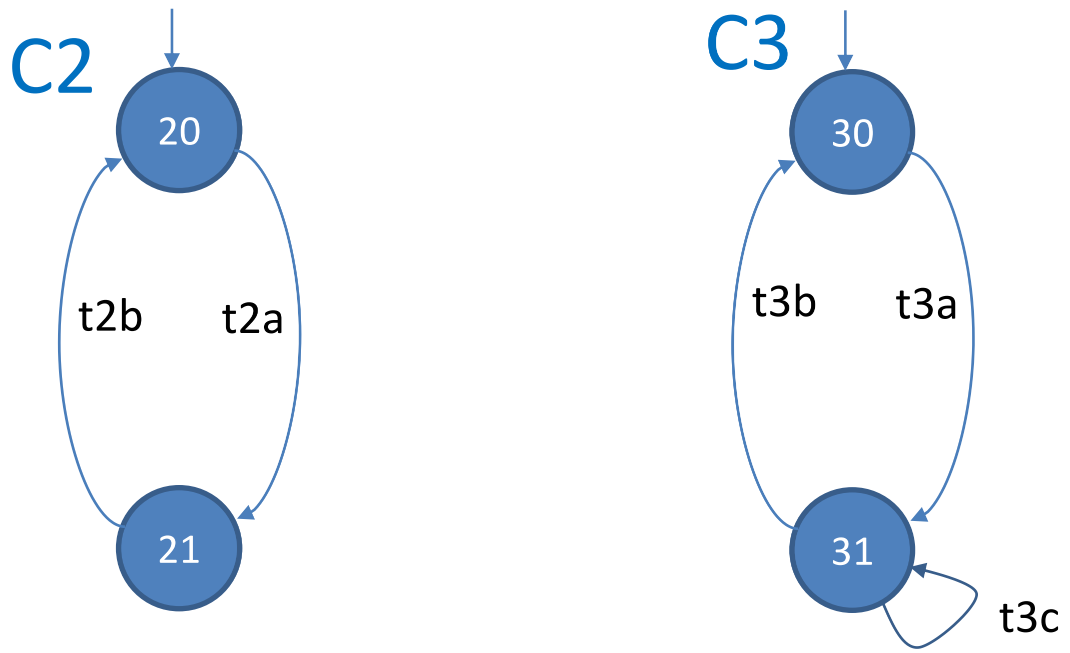
L'algoritmo produce in uscita l'insieme {$ab^*_{(1)}$, $ba^*_{(2)}$, $(ab^*|ba^*b)_{(3)}$}

CONTESTO

FA come modello comportamentale

- Un FA usato come modello comportamentale (o, più semplicemente, un FA comportamentale) è un FA in cui l'insieme degli stati di accettazione è vuoto
- Ogni transizione è dotata di un evento in ingresso (che può essere nullo) e di un insieme di eventi in uscita (che può essere vuoto)
- Possono esistere più transizioni aventi lo stesso stato sorgente e lo stesso stato destinazione, che differiscono per l'evento in ingresso e/o per gli eventi in uscita
- Possono esistere più transizioni distinte uscenti dal medesimo stato dotate del medesimo evento in ingresso (magari nullo), cioè in generale un FA comportamentale è un FA non deterministico (NFA) sull'alfabeto degli eventi in ingresso
- Condizione necessaria affinché una transizione uscente da uno stato s sia abilitata allo scatto è che s sia lo stato corrente dell'automa

Due FA comportamentali

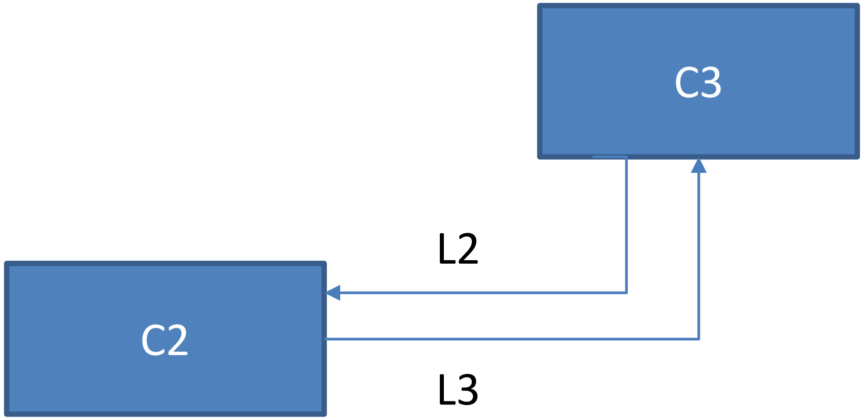


In questa rappresentazione grafica, entro ogni stato compare il suo nome (univoco) e a ogni transizione è affiancato il nome (univoco) della stessa

Rete (finita) di FA comportamentali

- È costituita da almeno un FA comportamentale
- Gli FA comportamentali sono disposti secondo una topologia distribuita orientata connessa, dove ogni singolo FA è un componente
- Ciascuna connessione orientata fra componenti distinti prende il nome di link
- Possono esistere più link aventi il medesimo componente sorgente e il medesimo componente destinazione

Topologia



In questa rappresentazione grafica della topologia, ogni componente è raffigurato come un rettangolo e ogni link come una freccia (il cui verso è quello di percorrenza del link). Alla rappresentazione di ogni link è affiancato il nome dello stesso

Link

- Si tratta di un buffer di capacità unitaria
- Ogni evento non nullo in ingresso a una transizione proviene da un link
- Gli eventi prodotti in uscita da una medesima transizione sono inviati ciascuno a un link distinto (necessariamente vuoto)
- Un link può servire in ingresso tante transizioni distinte del componente destinazione del link stesso
- Un link può servire in uscita tante transizioni distinte del componente sorgente del link stesso
- Eventi omonimi possono transitare su link distinti

Transizioni

C2

Evento in ingresso

- t2a: e2(L2)/{e3(L3)}
- t2b: /{e3(L3)}

Evento in ingresso nullo

C3

Insieme degli eventi in uscita

- t3a: /{e2(L2)}
- t3b: e3(L3)
- t3c: e3(L3)

Insieme vuoto di eventi in uscita

In questo esempio, fortuitamente, l'insieme degli eventi in uscita di una transizione è vuoto oppure è un singoletto. Nel caso generale, tale insieme può invece contenere molteplici eventi.
Analogamente, in questo esempio su ogni link transita un singolo tipo di evento (e2 su L2, e3 su L3) ma, in generale, sul medesimo link è consentito il transito di più tipi di evento

Transizioni e link

- Una transizione dotata di evento in ingresso è abilitata allo scatto solo se tale evento è effettivamente disponibile sul link di provenienza
- Una transizione che genera eventi in uscita è abilitata allo scatto solo se i link destinatari di tali eventi sono vuoti
- Una transizione con evento in ingresso nullo e un insieme vuoto di eventi in uscita può sempre scattare, purché abilitata

Osservabilità e rilevanza

- Siano Ω e F due insiemi di etichette, la cui unica intersezione è l'etichetta nulla (ε)
- Le relazioni di osservabilità e di rilevanza fanno corrispondere a ogni transizione dei componenti della rete rispettivamente un'etichetta di Ω e un'etichetta di F
- Una transizione a cui corrisponde un'etichetta non nulla di Ω si dice osservabile e le etichette non nulle in Ω prendono anche il nome di etichette/eventi osservabili
- Una transizione a cui corrisponde un'etichetta non nulla di F si dice rilevante
- Ogni etichetta non nulla in Ω e F deve essere l'immagine di almeno una transizione
- La medesima etichetta in Ω, così come la medesima etichetta in F, può corrispondere a più transizioni, anche di componenti distinti

Osservabilità

C2

- t2a: o2
- t2b: ε

C3

- t3a: o3
- t3b: ε
- t3c: ε

Rilevanza

C2

- t2a: ε
- t2b: r

C3

- t3a: ε
- t3b: ε
- t3c: f

In questo esempio, fortuitamente, ciascuna etichetta distinta di osservabilità (rilevanza) è associata a una singola transizione. Nel caso generale, essa può essere associata a più transizioni, anche di componenti distinti. Inoltre, in questo esempio a ciascuna transizione corrisponde al più una etichetta non nulla, mentre in generale alla medesima transizione possono corrispondere sia un'etichetta non nulla di osservabilità sia un'etichetta non nulla di rilevanza

SPAZIO COMPORTAMENTALE

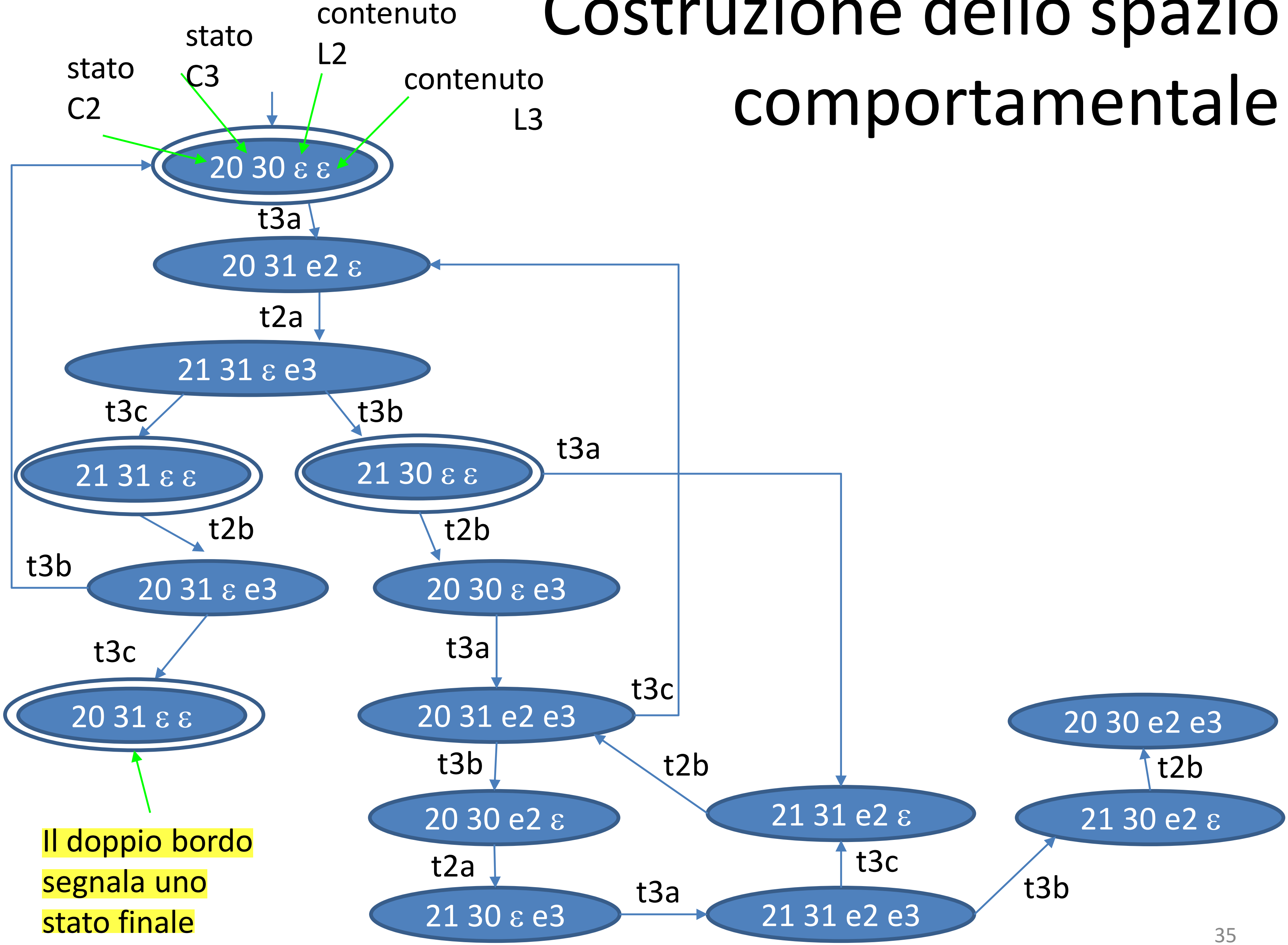
Stato di una rete di FA comportamentali

- Lo stato di una rete di FA comportamentali è costituito dallo stato di tutti i componenti e di tutti i link, dove per stato di un link si intende il contenuto (eventualmente vuoto) del link stesso
- Lo stato iniziale della rete è quello in cui ciascun componente è nel suo stato iniziale e i link sono tutti vuoti
- Uno stato della rete è finale se in esso tutti i link sono vuoti
- Lo scatto di una transizione abilitata di un singolo componente determina un passaggio di stato della rete
- Per traiettoria di una rete di automi si intende una sequenza di transizioni di componenti che porta dallo stato iniziale della rete a uno stato finale

Spazio comportamentale di una rete di automi

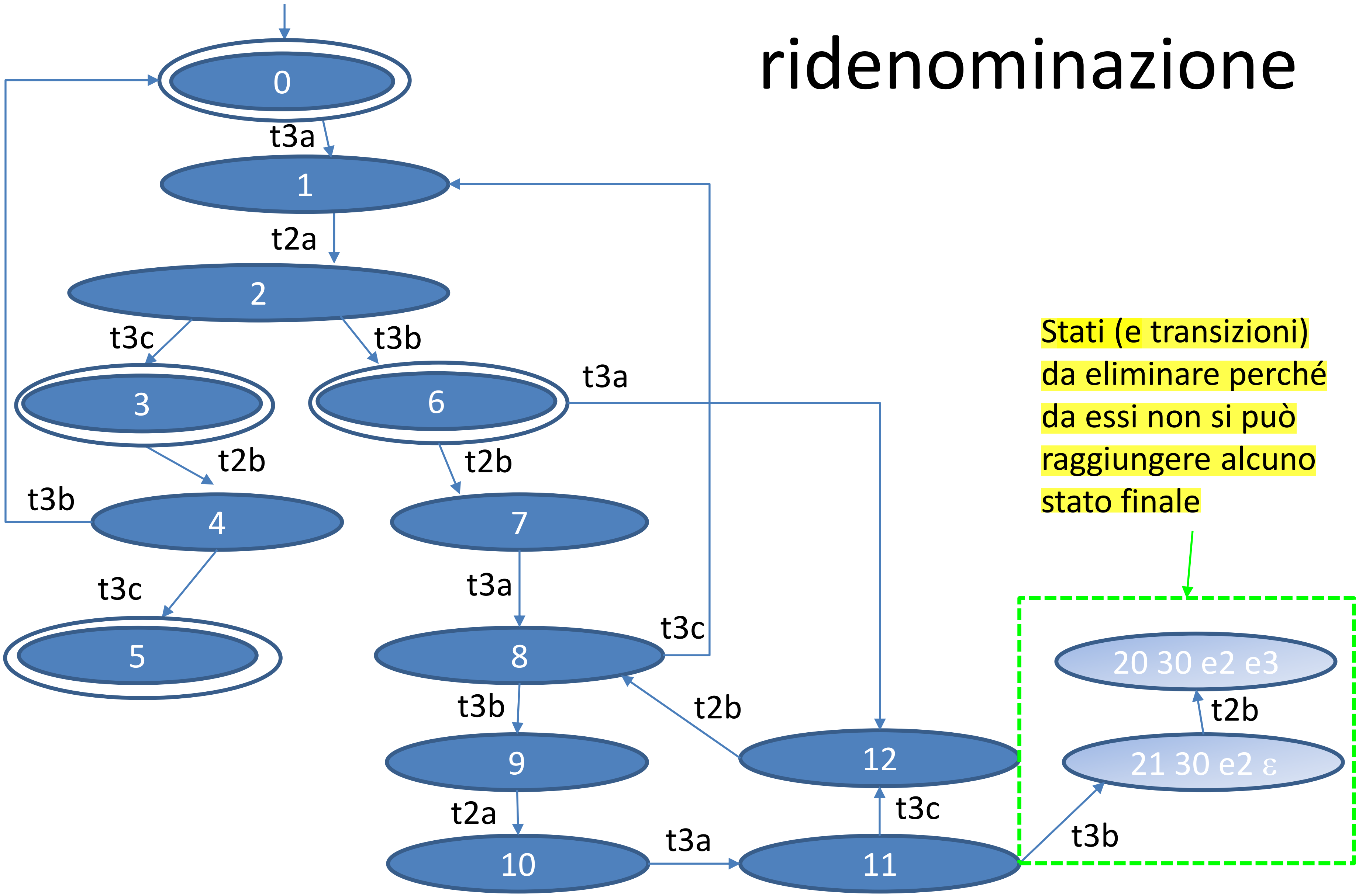
- È un FA deterministico (DFA) sull'alfabeto i cui simboli sono gli identificatori di tutte le transizioni dei componenti della rete; il linguaggio di tale DFA è l'insieme (che può contenere infiniti elementi) delle traiettorie della rete
- Si ottiene applicando tutti i possibili passaggi di stato, a partire dallo stato iniziale della rete
- A ogni passaggio di stato, lo stato destinazione (della rete) si ottiene creando una copia dello stato sorgente (della rete) e aggiornando entro la copia stessa lo stato del componente a cui la transizione che scatta si riferisce
- Una sequenza di transizioni che porta da uno stato della rete a un altro prende il nome di cammino

Costruzione dello spazio comportamentale



Il doppio bordo segnala uno stato finale

Potatura e ridenominazione

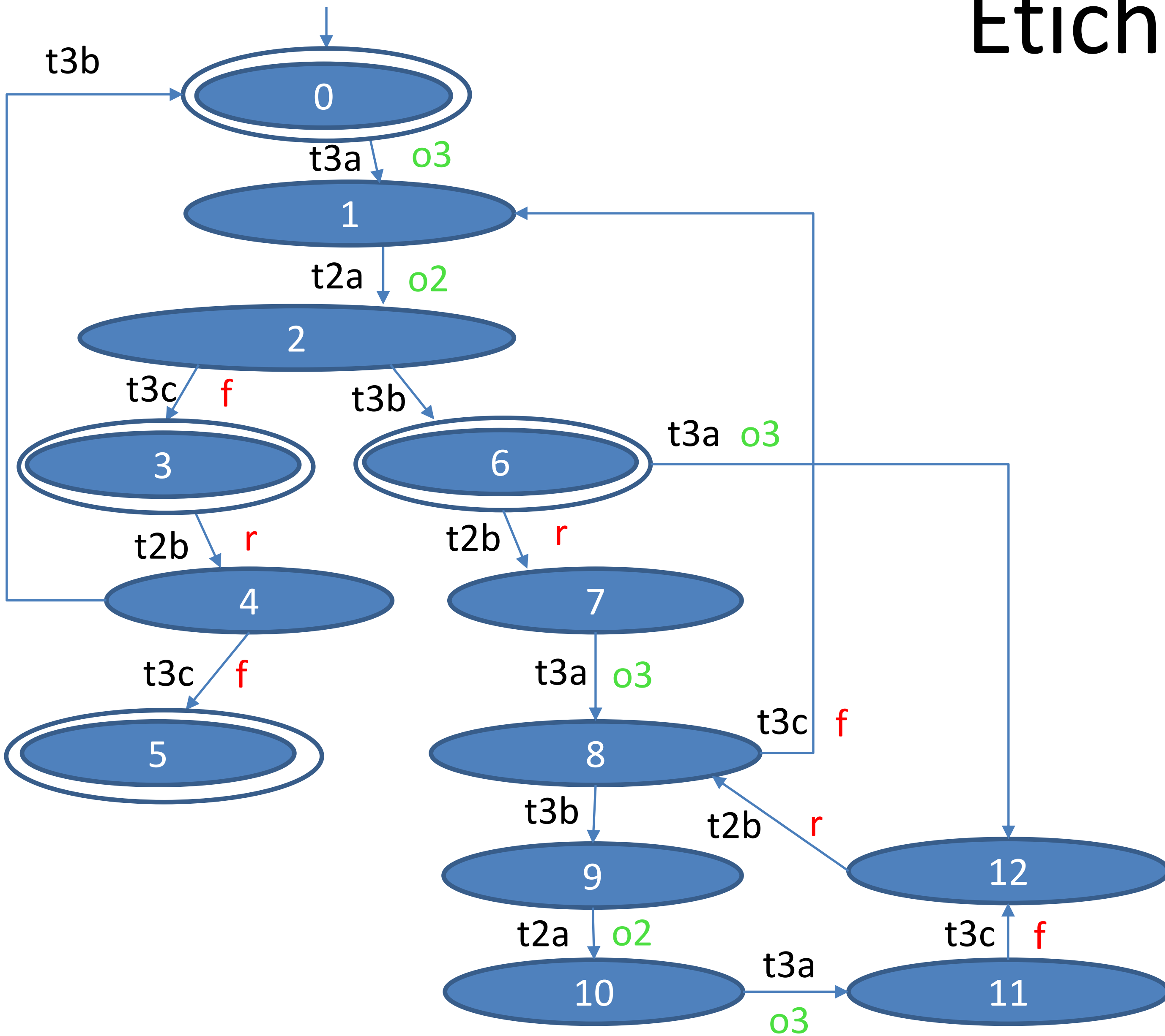


Stati (e transizioni) da eliminare perché da essi non si può raggiungere alcuno stato finale

Ridenominazione degli stati ed etichettatura delle transizioni

- A ogni stato dello spazio comportamentale può essere assegnato un identificatore univoco, come avvenuto nella pagina precedente; questa ridenominazione non modifica lo spazio
- Lo spazio comportamentale può essere annotato associando a ogni transizione osservabile la relativa etichetta in Ω (evento osservabile) e/o associando a ogni transizione rilevante la relativa etichetta in F (evento rilevante)
- Tale etichettatura, mostrata nella pagina successiva, non modifica lo spazio di cui sopra (ovvero non aggiunge né nuovi stati né nuove transizioni)

Etichettatura



A ogni transizione corrisponde una etichetta di osservabilità (verde) e una di rilevanza (rossa). Ogni etichetta (verde o rossa) mancante in figura è una etichetta nulla (ε). Le transizioni a cui non è affiancata alcuna etichetta non sono né osservabili né rilevanti

Compito

- Scrivere lo pseudocodice di un algoritmo che produca lo spazio comportamentale di una rete finita di FA comportamentali data
- Realizzare una applicazione software che incarni tale algoritmo
- ATTENZIONE: la ridenominazione operata potrà essere usata anche in ulteriori uscite dell'applicazione (di cui ai compiti successivi). Pertanto è necessario che all'utente venga resa nota la corrispondenza fra il nome assegnato a ciascuno stato e il contenuto dello stato stesso

SPAZIO COMPORTAMENTALE RELATIVO A UNA OSSERVAZIONE LINEARE

Osservazione lineare

- È una sequenza di eventi osservabili, ad esempio O = [o3, o2], verificatisi (nell'ordine stabilito dalla sequenza) lungo una traiettoria della rete di FA comportamentali
- Convenzionalmente, O[i], con i $\in$[1..length[O]] è l'i-mo evento dell'osservazione lineare
- La stessa osservazione lineare può corrispondere a più traiettorie

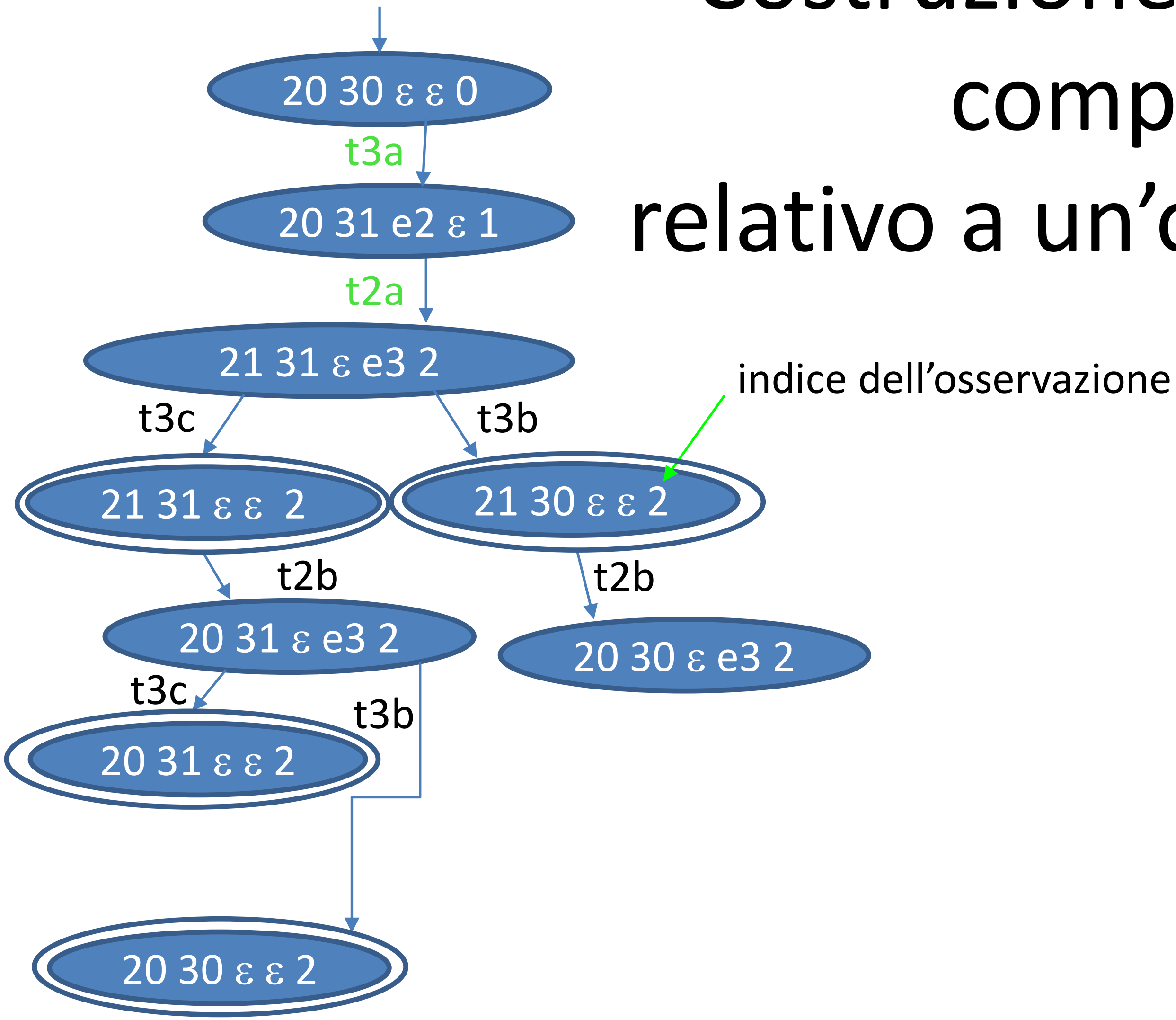
Spazio comportamentale relativo a una osservazione lineare

- È la porzione di spazio comportamentale che contiene tutte e sole le traiettorie che producono l'osservazione lineare data
- Esso può essere costruito con un algoritmo analogo a quello di costruzione dello spazio comportamentale, dove ogni stato dello spazio, in aggiunta agli stati dei componenti e dei link, contiene anche un indice dell'osservazione lineare data

Spazio comportamentale relativo a una osservazione lineare (cont.)

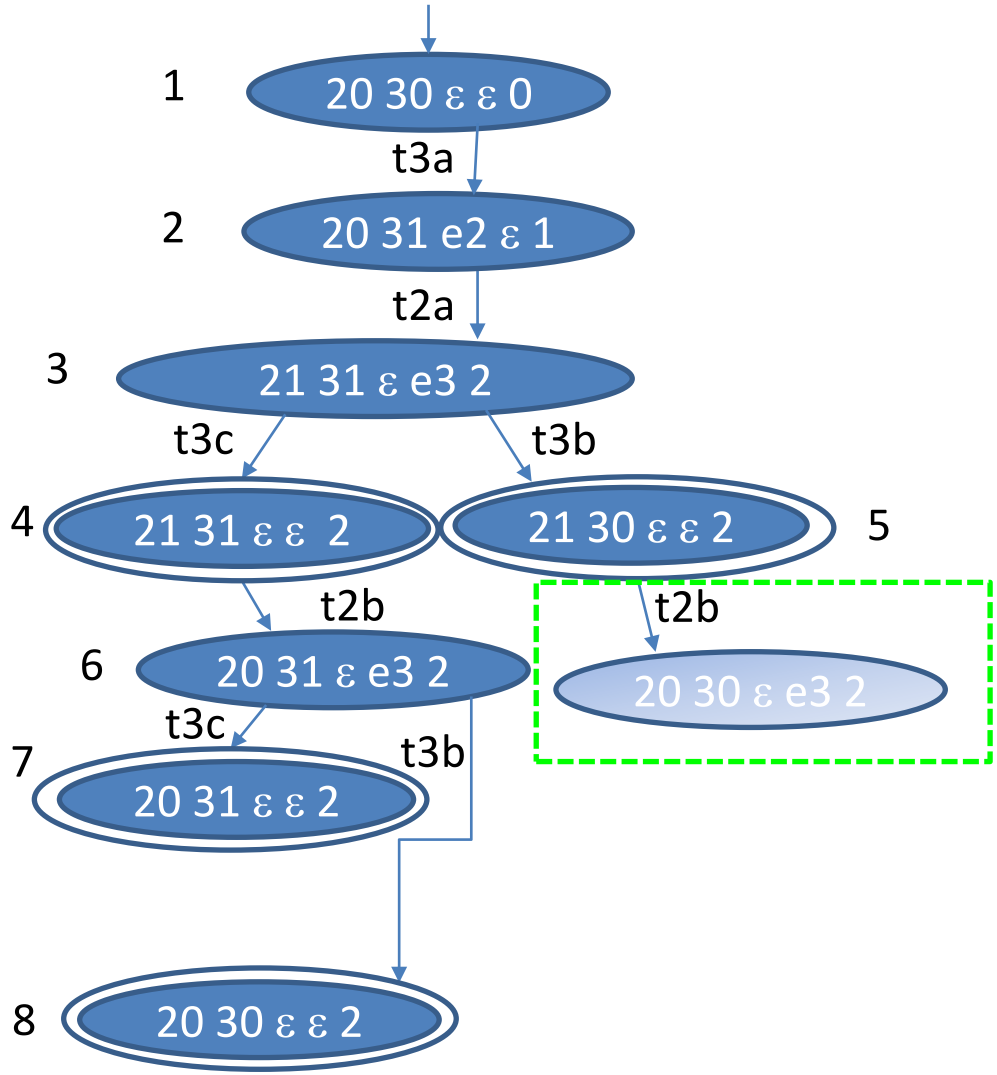
- Lo stato iniziale dello spazio comportamentale relativo a un’osservazione lineare O contiene lo stato iniziale di tutti i componenti e di tutti i link nonché il valore 0 (zero) dell’indice dell’osservazione
- Uno stato dello spazio comportamentale relativo a un’osservazione lineare è finale se tutti i link sono vuoti e il valore dell’indice è pari a length[O]
- Sia p uno stato dello spazio comportamentale relativo a un’osservazione e q l’indice dell’osservazione in esso contenuto. <u>A partire da p, una transizione osservabile non è abilitata se essa produce un evento osservabile diverso da O[q+1]</u>. Quando q = length[O], nessuna transizione osservabile è abilitata

Costruzione dello spazio comportamentale relativo a un’osservazione



L’osservazione considerata è O = [o3, o2]

Potatura e ridenominazione



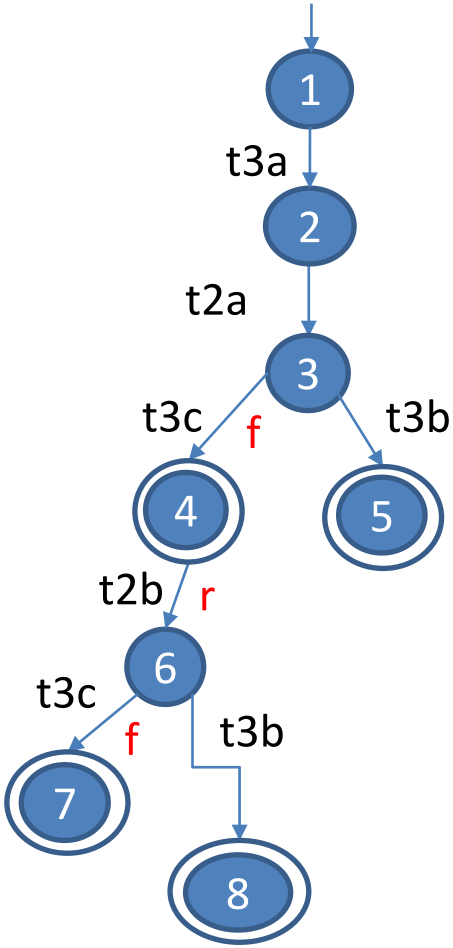
Stato (e transizione) da eliminare perché da esso non si può raggiungere alcuno stato finale

Compito

- Scrivere lo pseudocodice di un algoritmo che produca lo spazio comportamentale di una rete finita di FA comportamentali data relativo a una osservazione data. Tale pseudocodice non deve richiedere né la creazione né la conoscenza dello spazio comportamentale globale della rete
- Estendere l'applicazione software già realizzata in modo da rendere disponibile questa nuova funzionalità

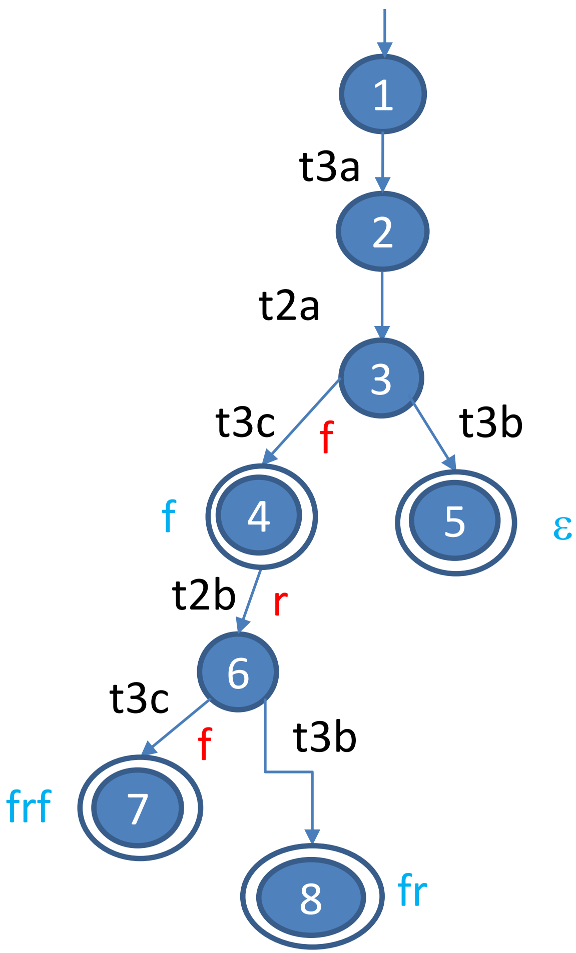
DIAGNOSI RELATIVA A UNA OSSERVAZIONE LINEARE

Etichettatura di rilevanza dello spazio comportamentale relativo a una osservazione



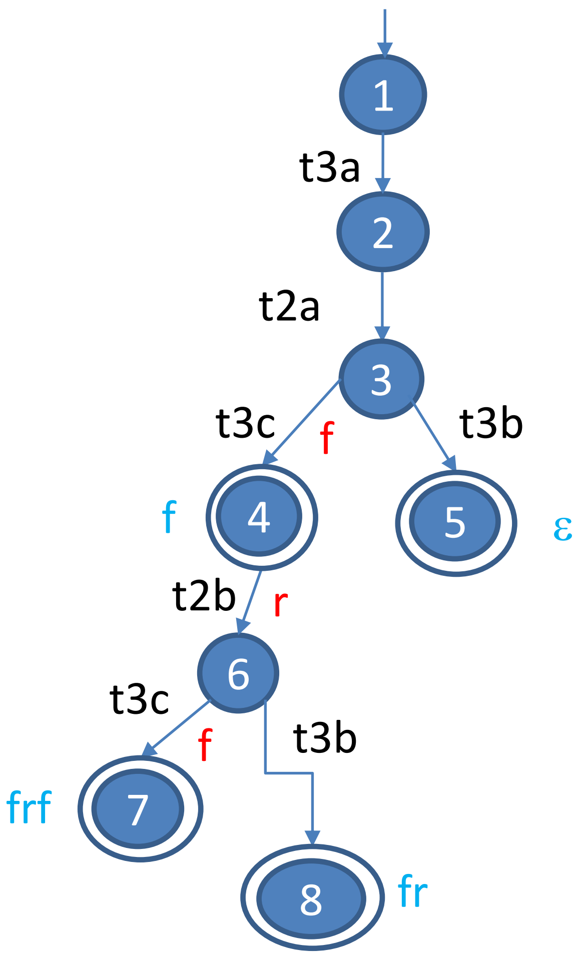
Questo automa è lo stesso di pag. 45 (dopo la potatura), cioè è lo spazio comportamentale relativo all'osservazione [o3, o2], alle cui transizioni sono state affiancate (in rosso) le etichette di rilevanza

Espressioni regolari (di rilevanza) relative agli stati finali



A ogni stato finale dello spazio di rilevanza relativo a un'osservazione corrisponde l'espressione regolare (in azzurro) ottenuta concatenando le etichette di rilevanza (quelle in rosso) lungo ciascuna traiettoria che porta dallo stato iniziale allo stato finale considerato. Se uno stato finale è raggiunto da più traiettorie (in questo esempio non succede), l'espressione regolare associata è l'alternativa delle espressioni relative a tali traiettorie

Diagnosi relativa a una osservazione lineare



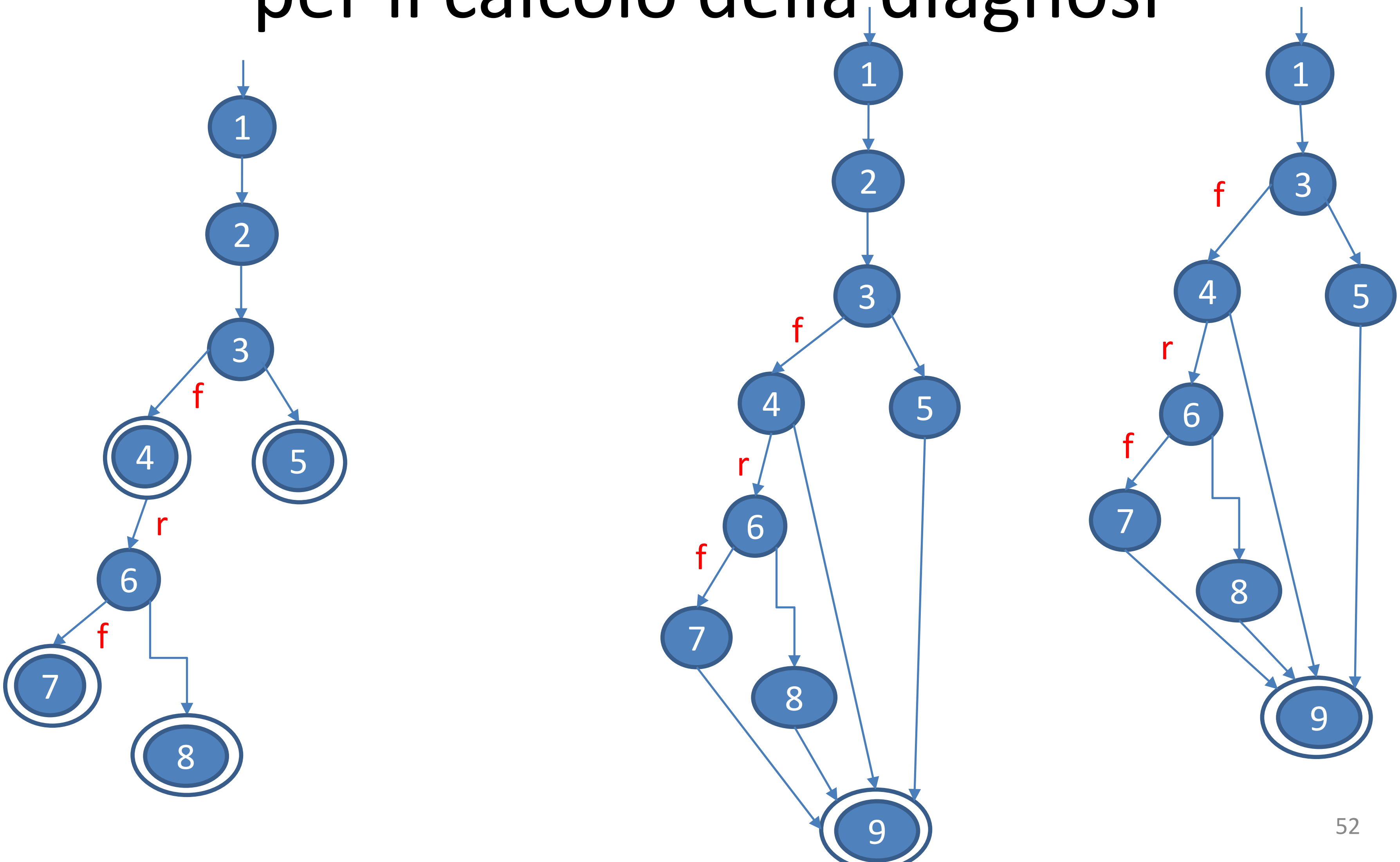
Per definizione, la diagnosi relativa a un'osservazione lineare è l'alternativa delle espressioni regolari di rilevanza (in azzurro) corrispondenti agli stati finali dello spazio comportamentale inerente all'osservazione stessa

Diagnosi relativa all'osservazione [o3, o2] = (f(r|rf)?)?

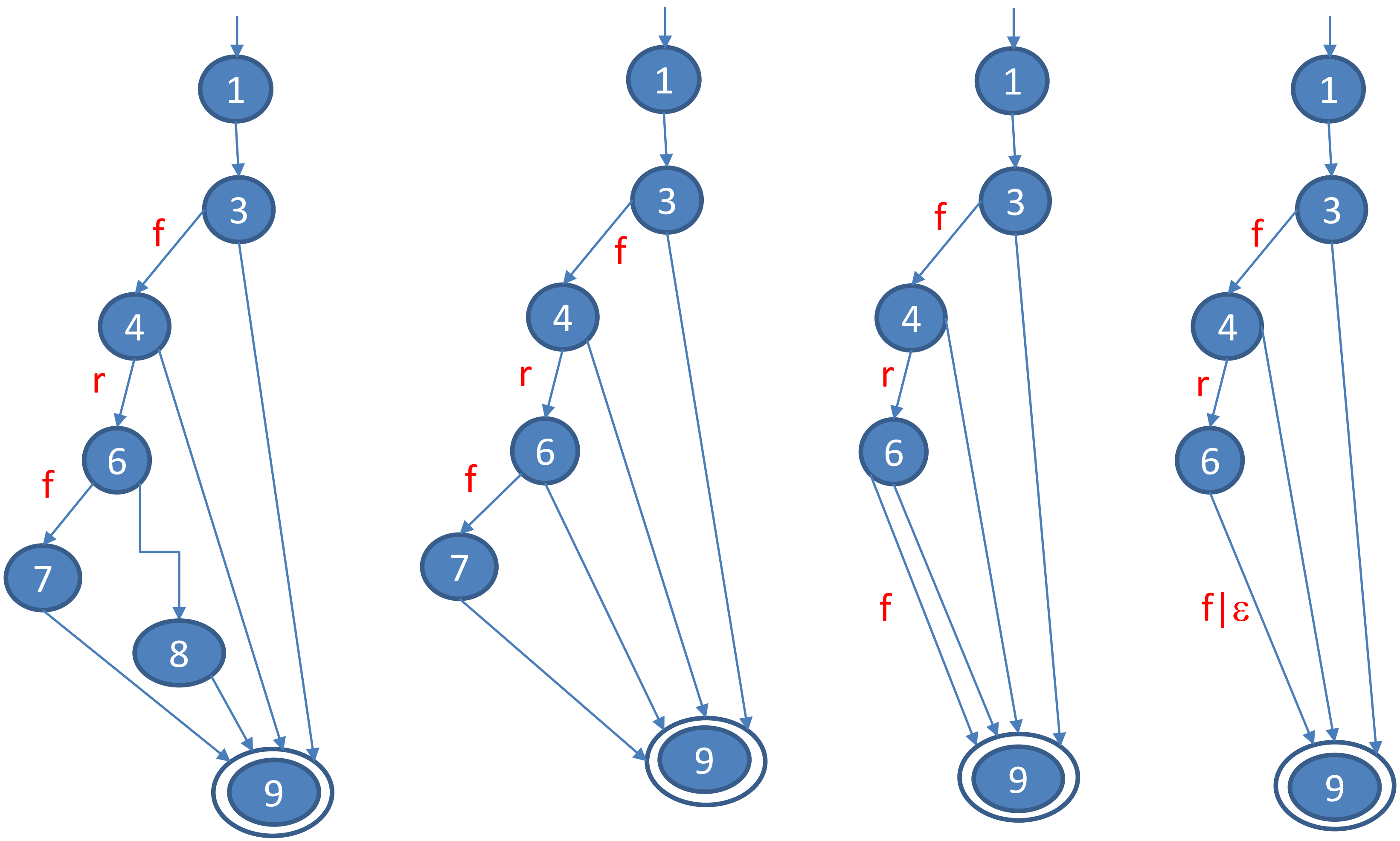
Calcolo della diagnosi relativa a una osservazione lineare

- La diagnosi relativa a una osservazione lineare può essere calcolata invocando l'algoritmo EspressioneRegolare(N_{in}), dove N_{in} è lo spazio comportamentale relativo all'osservazione - considerando i suoi stati finali come stati di accettazione dell'automa - e l'alfabeto dei simboli associati alle transizioni è quello delle etichette di rilevanza
- Le pagine successive illustrano le trasformazioni effettuate da tale algoritmo sull'automa che rappresenta lo spazio comportamentale di cui alla pagina precedente al fine di produrre l'espressione regolare che definisce il linguaggio accettato, cioè la diagnosi

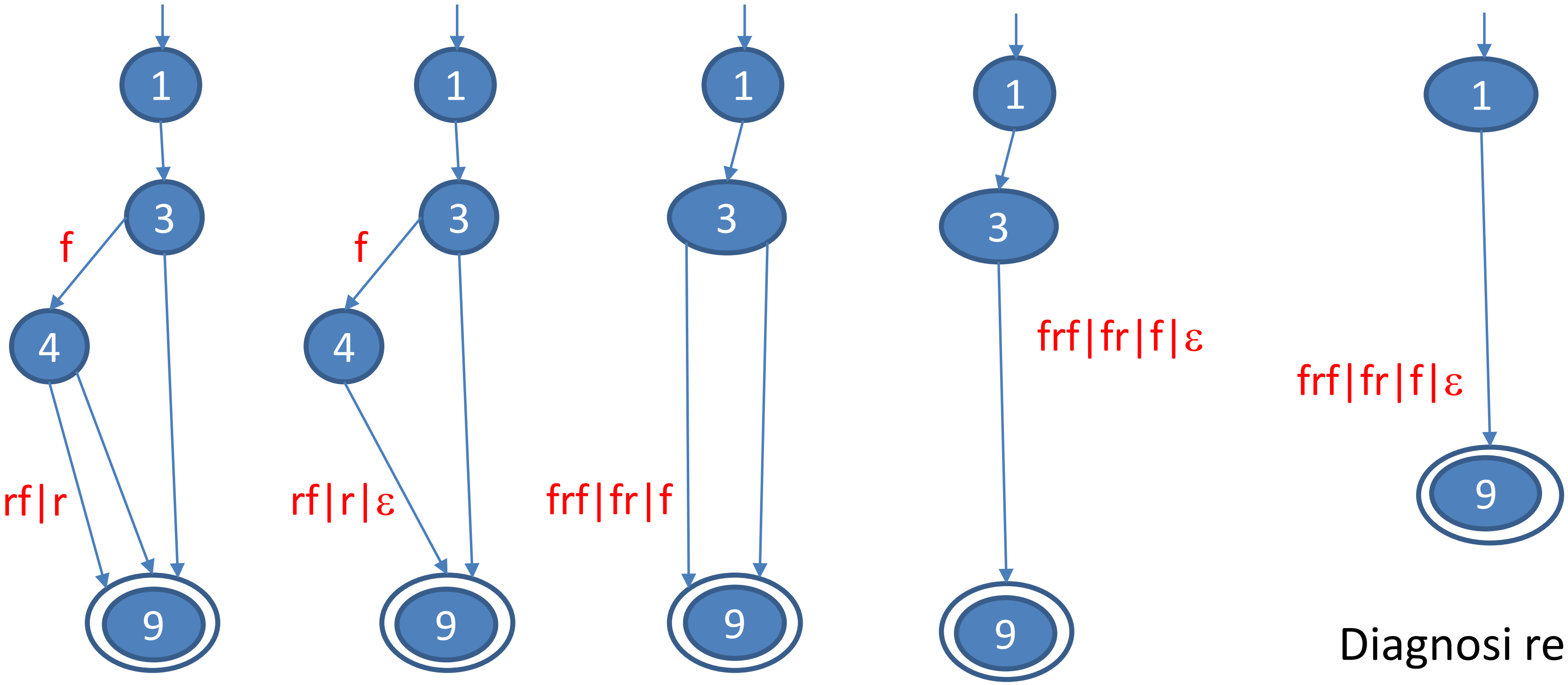
Applicazione dell'algoritmo EspressioneRegolare per il calcolo della diagnosi



Applicazione dell'algoritmo EspressioneRegolare per il calcolo della diagnosi



Applicazione dell'algoritmo EspressioneRegolare per il calcolo della diagnosi



Diagnosi relativa all'osservazione [o3, o2] = frf|fr|f|ε =(f(r|rf)?)?

Compito

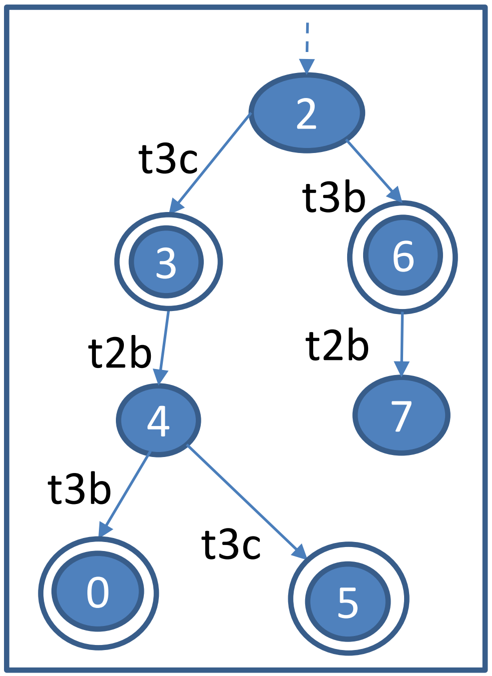
- Scrivere lo pseudocodice di un algoritmo che, noto lo spazio comportamentale relativo a un'osservazione lineare data di una rete di FA comportamentali, produca la diagnosi relativa
- Estendere l'applicazione di cui ai compiti precedenti in modo che, dietro richiesta dell'utente, fornisca la diagnosi relativa a un'osservazione lineare data sfruttando lo spazio comportamentale relativo alla stessa
- ATTENZIONE: perché l'elaborazione abbia successo, cioè produca una diagnosi, è necessario che in ingresso sia effettivamente fornita un'osservazione lineare inerente a una traiettoria della rete (altrimenti si dovrebbe avvertire l'utente che, purtroppo, non sempre conosce le osservazioni lineari della rete)

DIAGNOSTICATORE

Chiusura silenziosa

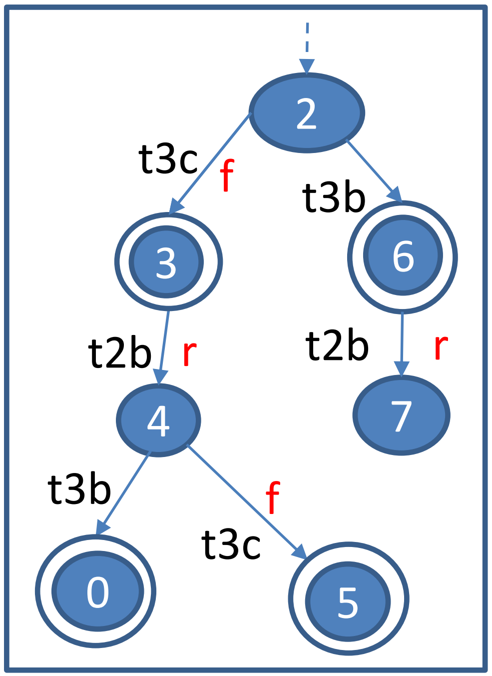
- A ogni stato s dello spazio comportamentale che sia o lo stato iniziale dello spazio stesso o uno stato avente almeno una transizione osservabile entrante corrisponde la sua cosiddetta «chiusura silenziosa», che è esclusiva di quello stato; s è lo stato d’ingresso della sua chiusura
- La chiusura silenziosa di s è costituita dal sottospazio dello spazio comportamentale che contiene tutti (e soli) gli stati raggiungibili a partire da s attraverso cammini contenenti esclusivamente transizioni non osservabili (dette anche silenziose)

Chiusura silenziosa (cont.)



- La figura mostra la chiusura silenziosa dello stato 2 dello spazio comportamentale illustrato a pag. 38
- Lo stato d'ingresso della chiusura è 2
- Gli stati finali in tale spazio sono evidenziati mediante un doppio bordo anche in questa figura

Chiusura silenziosa etichettata

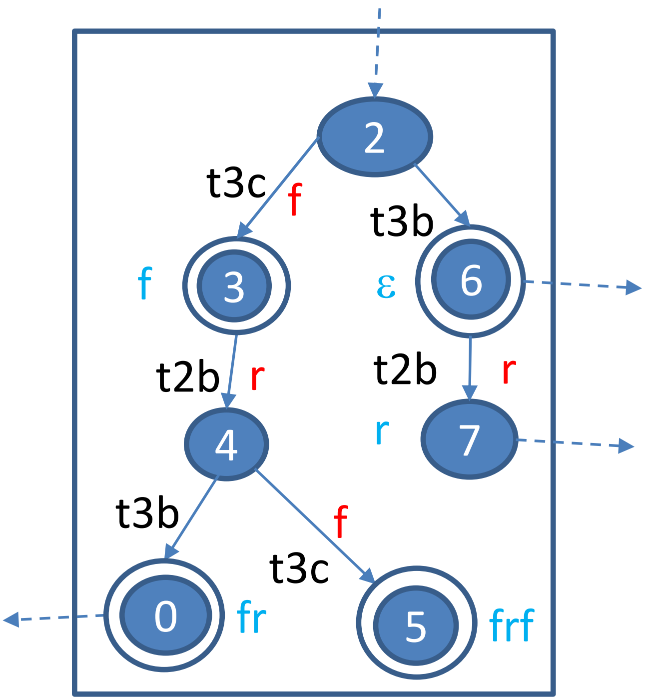


- La figura mostra la chiusura silenziosa della pagina precedente con la relativa etichettatura di rilevanza (in rosso)
- L’etichetta associata a ciascuna transizione a cui in figura non è affiancata alcuna etichetta è quella nulla (ε)

Chiusura silenziosa decorata

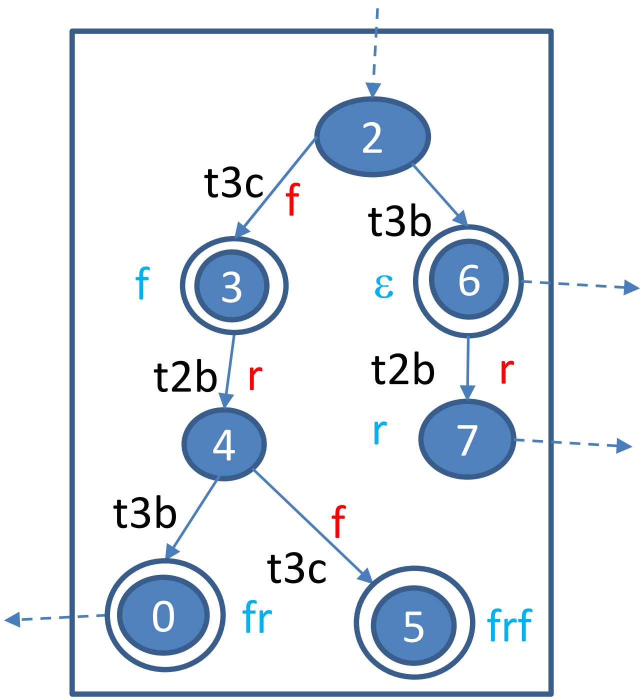
- Sia S l'insieme degli stati contenuti nella chiusura silenziosa di uno stato s. Per decorazione di ciascuno stato s'$\in$S, che sia finale o dotato di transizioni osservabili uscenti entro lo spazio comportamentale (uno stato cosiffatto si dice stato d'uscita della chiusura silenziosa), si intende l'espressione regolare di rilevanza relativa a tutti i cammini che portano da s a s' entro la chiusura silenziosa
- Il calcolo delle decorazioni può avvenire invocando l'algoritmo EspressioniRegolari, passando allo stesso la chiusura silenziosa vista come FA in cui lo stato d'ingresso è lo stato iniziale mentre gli stati di accettazione sono sia gli stati finali sia gli stati d'uscita (si noti che il medesimo stato può essere d'ingresso e/o d'uscita e/o finale)

Chiusura silenziosa decorata



- La figura mostra le decorazioni (in azzurro) relativa agli stati finali o d'uscita della chiusura silenziosa di cui alla pag. 59
- Le frecce tratteggiate evidenziano le transizioni osservabili entranti e uscenti dagli stati della chiusura (esse non appartengono alla chiusura stessa)

Diagnosi relativa a una chiusura silenziosa



È l’alternativa delle decorazioni relative agli stati finali contenuti nella chiusura; essa è indicata con Δ. In questo esempio è

$$\Delta = f\,|\,fr\,|\,frf\,|\,\varepsilon = (f(r(f)?)?)?$$

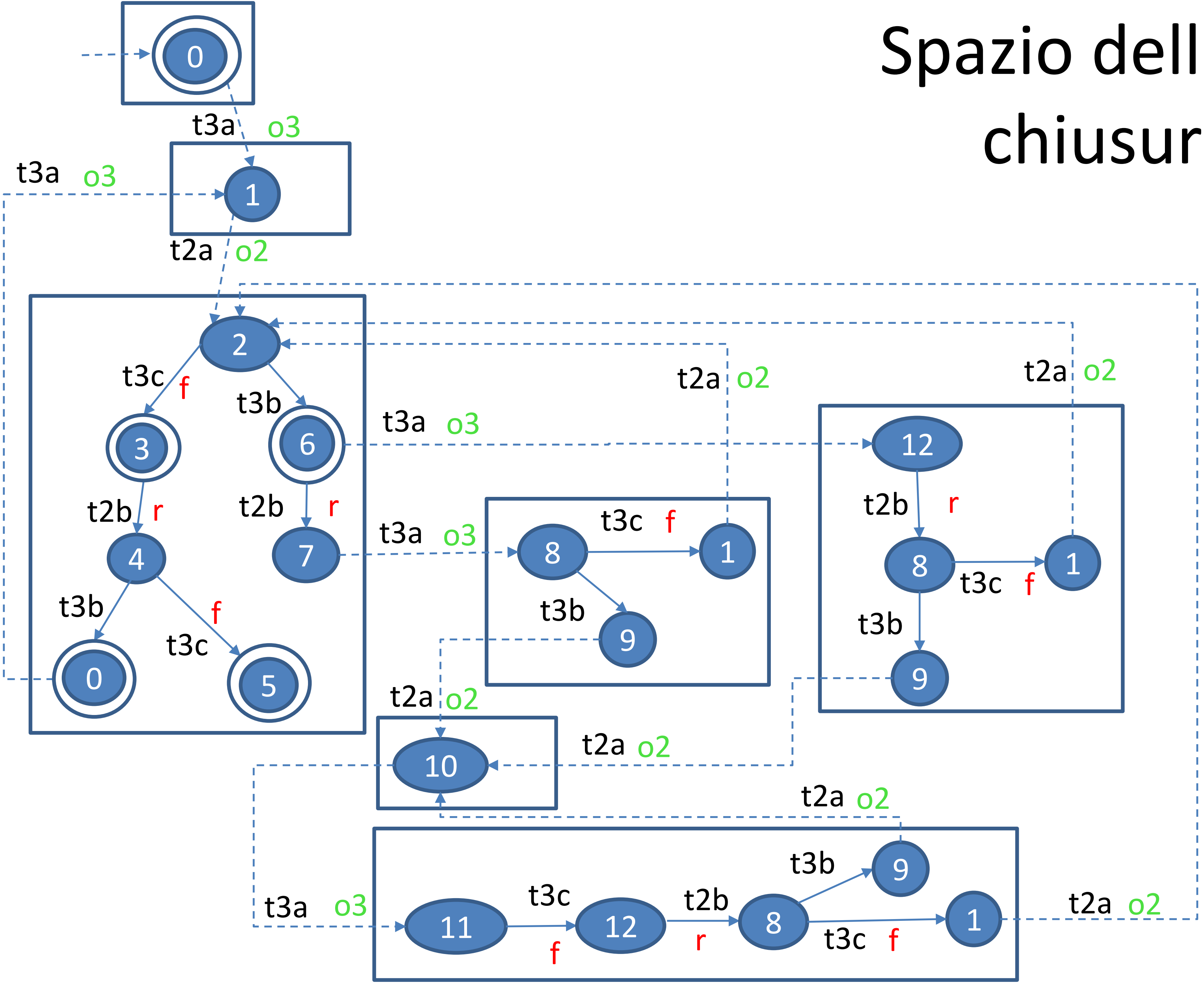
Chiusure silenziose

- Gli insiemi degli stati dello spazio comportamentale contenuti in chiusure silenziose distinte non sono necessariamente disgiunti (anche se non possono essere uguali fra loro); lo stesso vale per gli insiemi delle transizioni
- Ogni stato dello spazio comportamentale della rete appartiene ad almeno una chiusura silenziosa
- Ciascuna chiusura è univocamente identificata dal suo stato d'ingresso. Il numero delle chiusure distinte inerenti al medesimo spazio comportamentale è pari al numero di stati che, nello spazio comportamentale, sono dotati di transizioni osservabili entranti (anche auto-transizioni)

Interconnessione delle chiusure silenziose

- Le chiusure silenziose relative allo spazio comportamentale di una rete di FA comportamentali sono fra loro interconnesse mediante transizioni osservabili, uscenti da stati d'uscita della chiusura sorgente e dirette allo stato d'ingresso della chiusura destinazione (dove la chiusura sorgente può coincidere con la chiusura destinazione)
- Lo stato d'ingresso di una chiusura può appartenere anche ad altre chiusure, dove però non è uno stato d'ingresso. Nessuna interconnessione fra chiusure deve essere diretta a stati che non siano d'ingresso
- La figura successiva illustra tutte le chiusure silenziose relative allo spazio comportamentale della rete sinora considerata come esempio e tutte le reciproche interconnessioni, evidenziando anche l'etichettatura di osservabilità e quella di rilevanza
- Tale struttura prende il nome di spazio delle chiusure

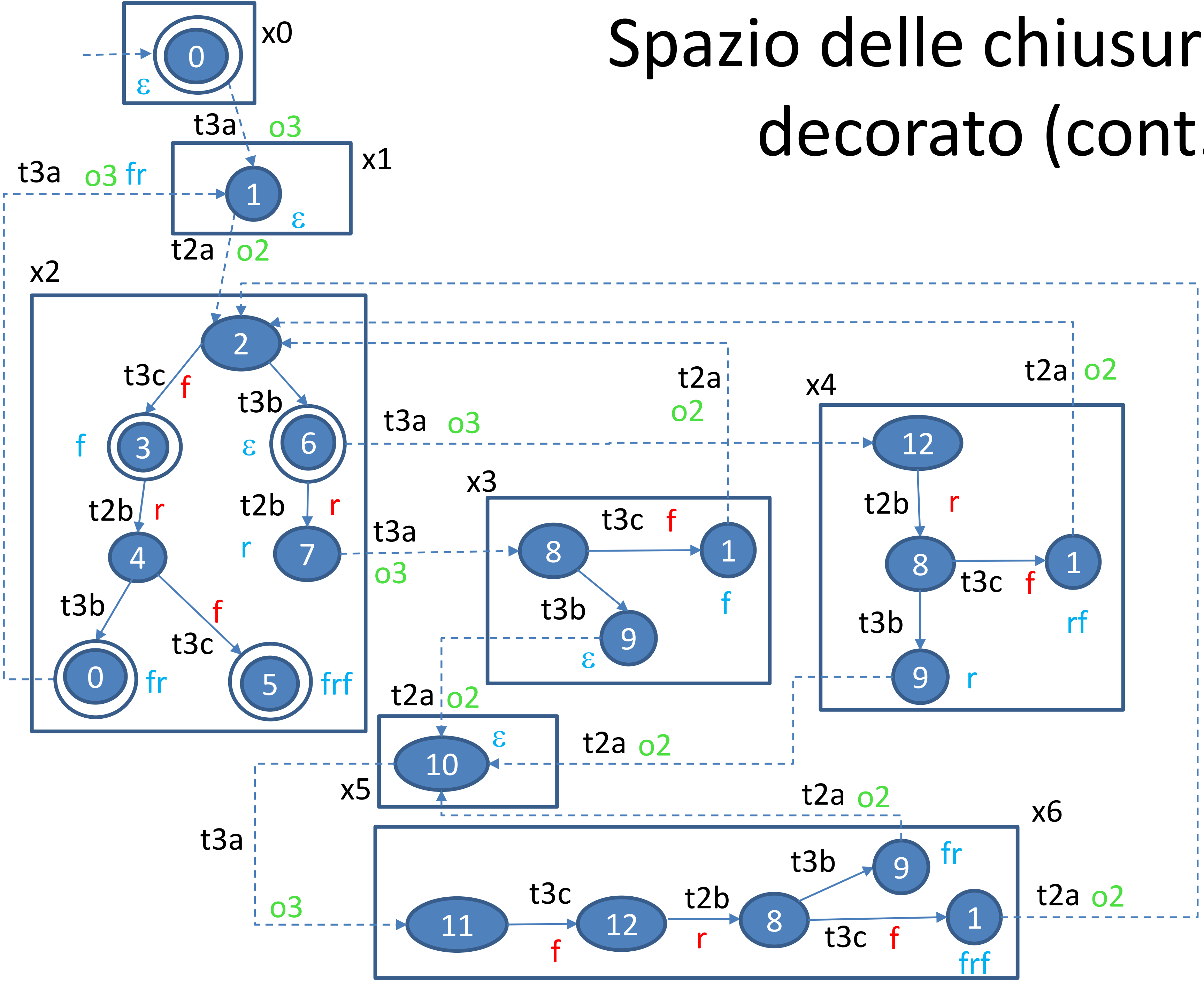
Spazio delle chiusure



Spazio delle chiusure decorato

- È semplicemente lo spazio delle chiusure in cui ciascuna chiusura è decorata
- Questo spazio è preliminare alla creazione del cosiddetto diagnosticatore

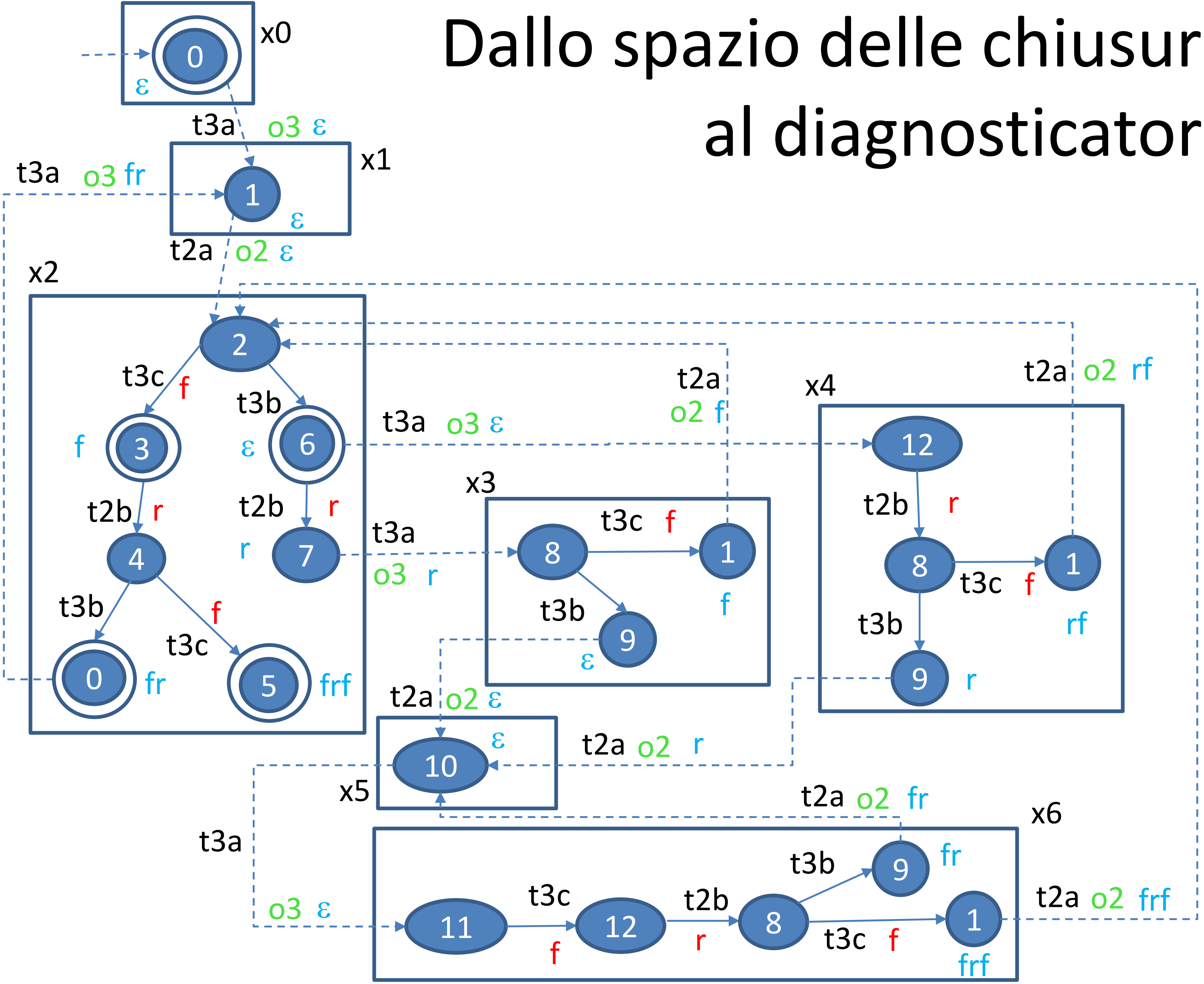
Spazio delle chiusure decorato (cont.)



Dallo spazio delle chiusure decorato al diagnosticatore

- Al fine di costruire il diagnosticatore, è necessario che la decorazione di ciascuno stato d’uscita entro le chiusure sia ‘ricopiata’ sulla transizione osservabile da esso uscente, concatenando alla decorazione l’eventuale etichetta di rilevanza relativa alla transizione osservabile stessa
- Questo passaggio è illustrato nella pagina seguente. Nell’esempio, a ciascuna decorazione ‘ricopiata’ è stata concatenata l’etichetta nulla perché, fortuitamente, tutte le transizione osservabili sono dotate di etichetta di rilevanza nulla

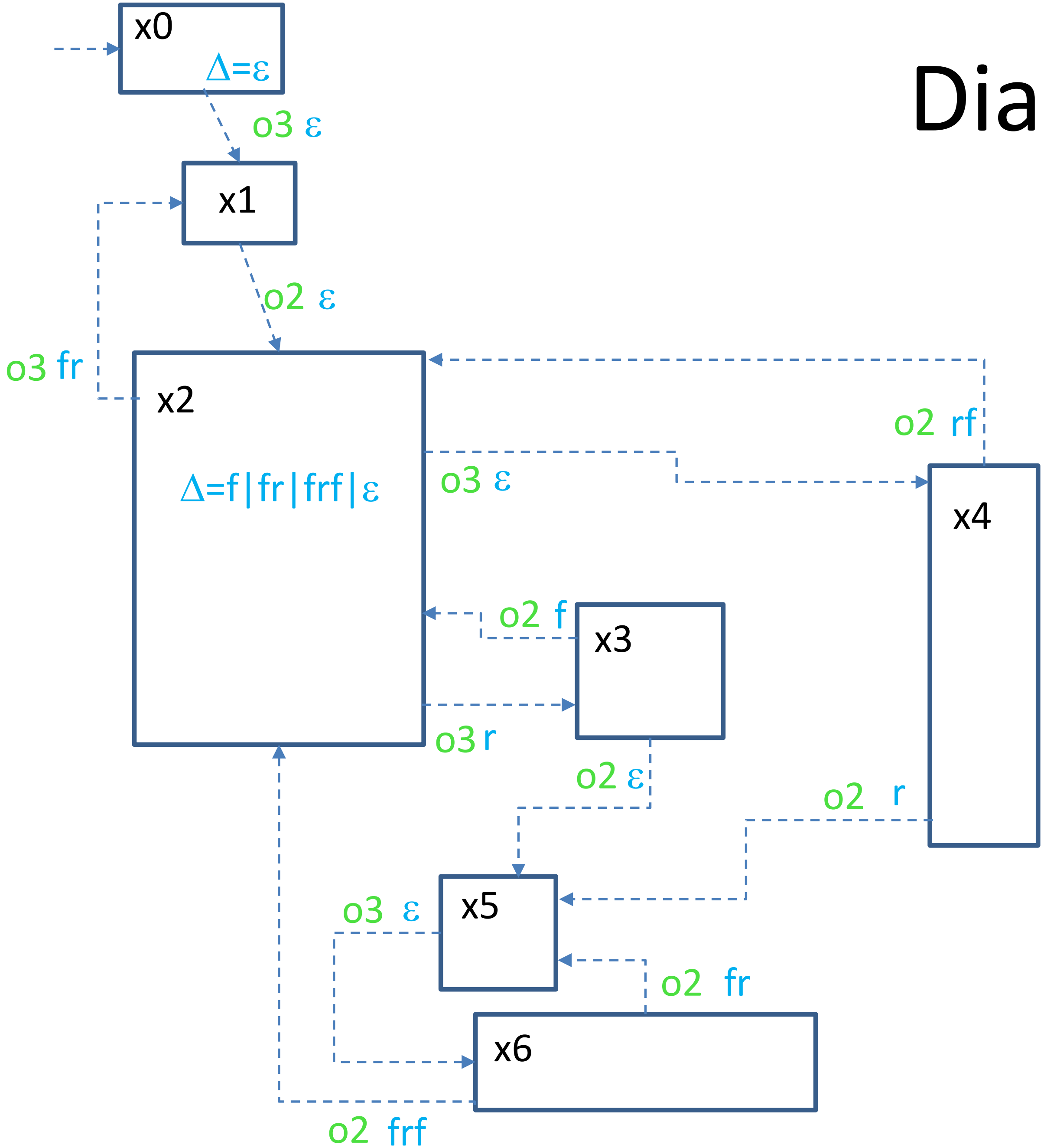
Dallo spazio delle chiusure al diagnosticatore



Diagnosticatore di una rete di FA comportamentali

- Per ottenere il diagnosticatore è infine necessario associare a ciascuna chiusura che contenga degli stati finali la sua diagnosi
- Ai fini del ragionamento diagnostico, le informazioni utili sono solo quelle indicate nella pagina seguente
- Nell'esempio, alle chiusure x1, x3, x4, x5 e x6 non è associata alcuna diagnosi perché esse non contengono stati finali

Diagnosticatore



Può essere visto come un FA dove ogni stato corrisponde a una chiusura silenziosa (di cui però non interessa conoscere il contenuto), a ciascuno stato di accettazione è associata una diagnosi Δ, ogni transizione è dotata di etichetta di osservabilità e di una espressione regolare

Diagnosticatore (cont.)

- Per traiettoria del diagnosticatore si intende una sequenza di transizioni (osservabili) dello stesso che porta dallo stato iniziale (cioè quello corrispondente alla chiusura relativa allo stato iniziale dello spazio comportamentale) a uno stato di accettazione (cioè a uno stato del diagnosticatore corrispondente a una chiusura che contiene qualche stato finale dello spazio comportamentale)
- Pertanto una traiettoria del diagnosticatore non comprende alcuna transizione interna alle chiusure dello spazio delle chiusure
- Come già detto, una traiettoria termina necessariamente in uno stato relativo a una chiusura dotata di stati finali dello spazio comportamentale; pertanto tale stato del diagnosticatore ha una diagnosi associata (quella della chiusura corrispondente)

Compito

- Scrivere lo pseudocodice di un algoritmo che, dato lo spazio comportamentale di una rete e uno stato d'ingresso (dove tale stato è o lo stato iniziale dello spazio comportamentale delle rete considerata o uno stato appartenente a tale spazio e dotato di almeno una transizione osservabile entrante), sia in grado di produrre la sua chiusura silenziosa
- Scrivere lo pseudocodice di un algoritmo che sfrutti l'invocazione del modulo di cui al punto precedente al fine di generare lo spazio delle chiusure
- Scrivere lo pseudocodice di un algoritmo che sia in grado di produrre il diagnosticatore della rete considerata
- Estendere l'applicazione di cui alle richieste già avanzate in modo che contempli anche la generazione del diagnosticatore

DIAGNOSI RELATIVA A OSSERVAZIONI LINEARI BASATA SUL DIAGNOSTICATORE

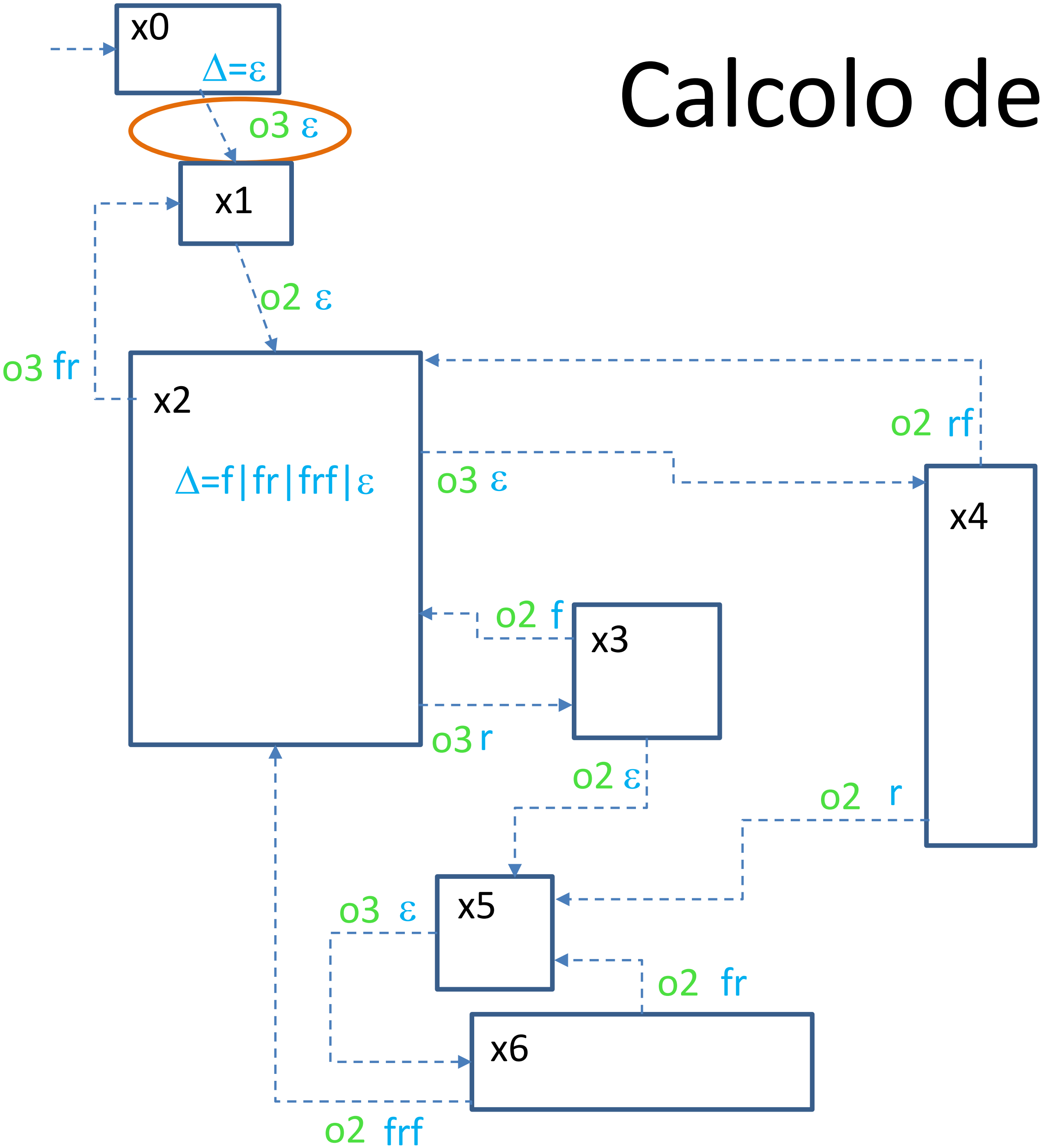
Diagnosi relativa a una osservazione lineare

Data una qualsivoglia osservazione lineare O relativa a una rete di FA comportamentali e il diagnosticatore di tale rete, la diagnosi corrispondente a O può essere calcolata come segue:
a partire dallo stato iniziale del diagnosticatore, si seguono tutte le traiettorie del diagnosticatore stesso che producono O.
Per ciascuna traiettoria si costruisce un’espressione regolare che è la concatenazione delle espressioni regolari che si trovano lungo le transizioni (osservabili) della stessa.
A tale concatenazione si fa seguire (sempre mediante concatenazione) la diagnosi relativa allo stato di accettazione (del diagnosticatore) raggiunto alla fine della traiettoria.
La diagnosi relativa a O è data dall’alternativa di tutte le espressioni così costruite

Calcolo della diagnosi

- Al contrario del metodo dedicato al calcolo della diagnosi relativa a una singola osservazione lineare (già illustrato nella sezione che inizia a pag. 47), lo sfruttamento del diagnosticatore consente di calcolare la diagnosi relativa a qualsiasi osservazione lineare della rete in uso
- Nelle pagine che seguono è illustrato un esempio di calcolo della diagnosi relativa a una osservazione lineare sfruttando il diagnosticatore
- L'osservazione considerata è O = [o3, o2, o3, o2]
- Nell'esempio, ogni coppia (xi,ρ) è costituita dall'identificatore di uno stato del diagnosticatore (corrispondente a una chiusura) e da una espressione regolare
- La coppia di partenza è (x0,ε), cioè lo stato iniziale del diagnosticatore e un'espressione regolare nulla

Calcolo della diagnosi



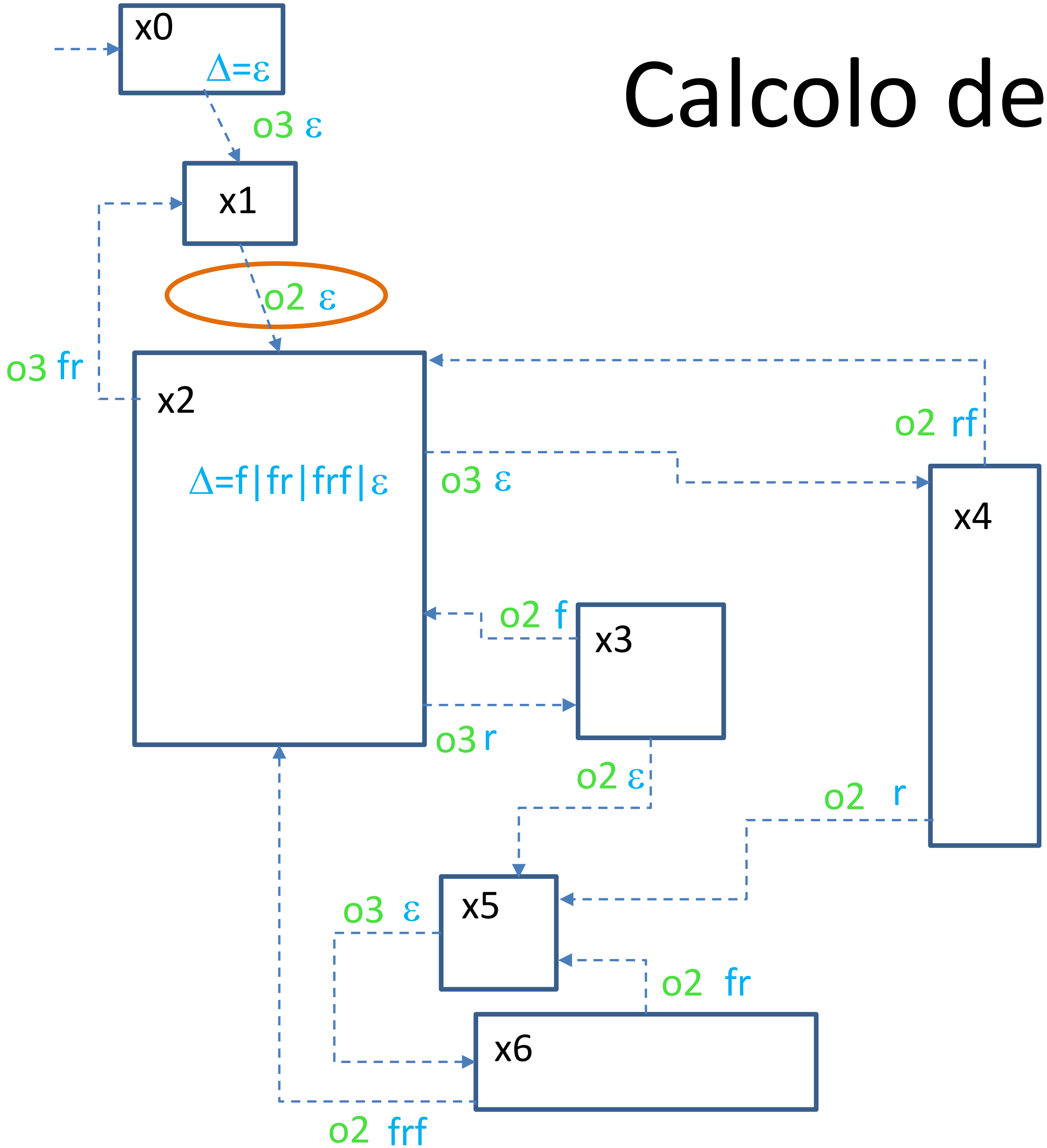
evento o3:

(x0,ε) o3 (x1,ε)

stato ed espressione a monte dell'evento

stato ed espressione a valle dell'evento

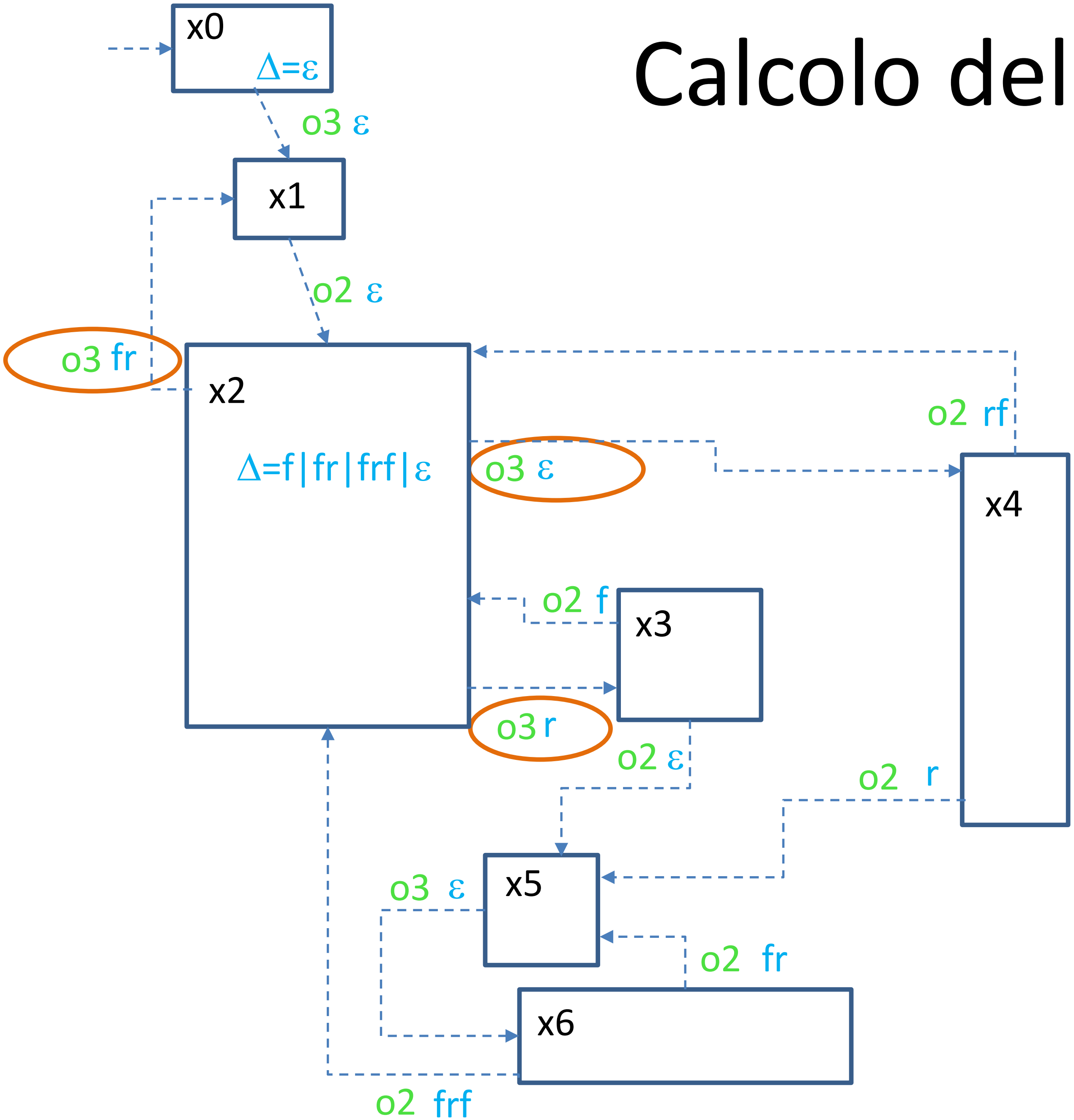
Calcolo della diagnosi



evento o2:

(x1,ε) o2 (x2,ε)

Calcolo della diagnosi



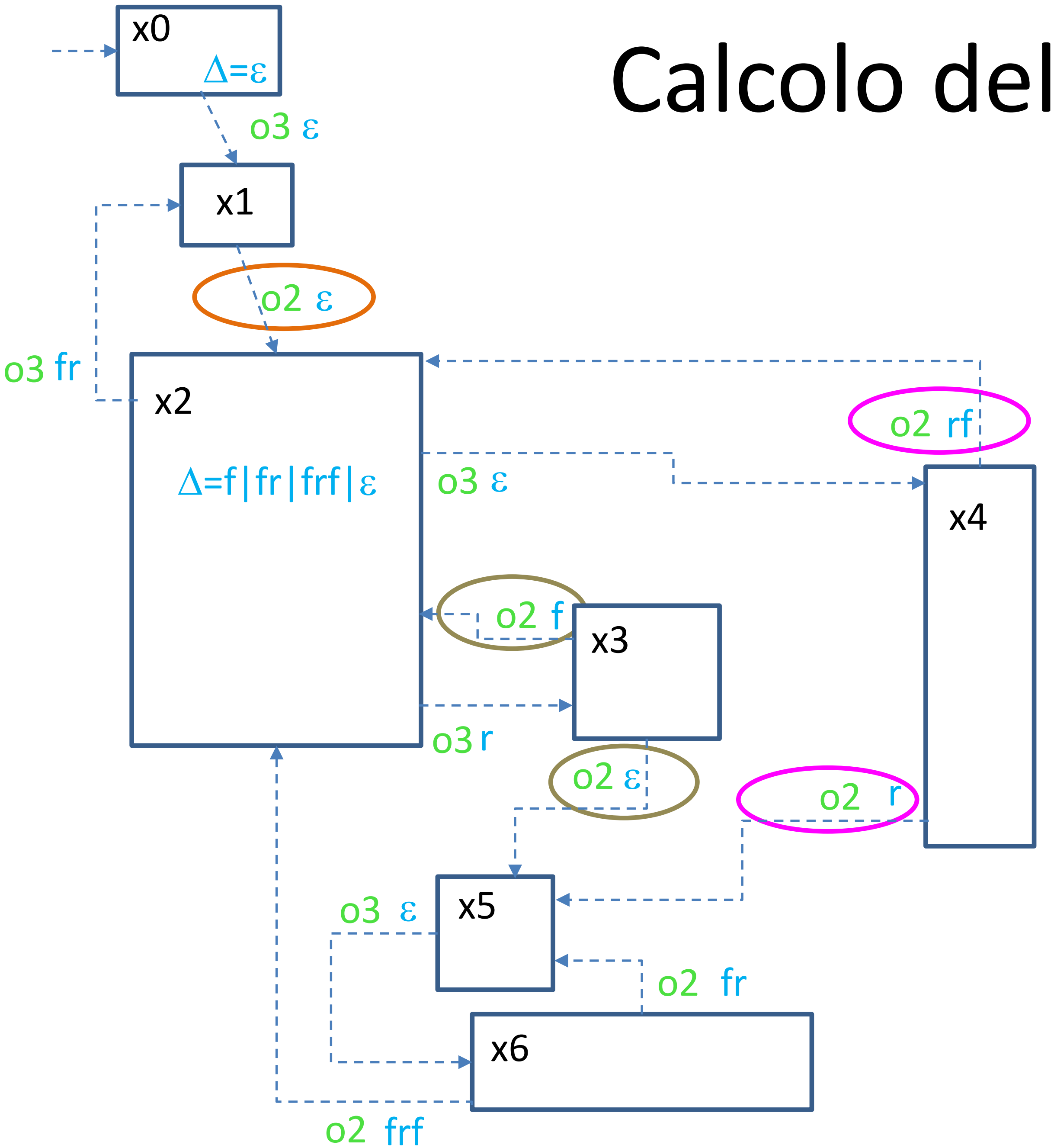
evento o3:

(x2,ε) o3 (x1,fr)

(x2,ε) o3 (x4,ε)

(x2,ε) o3 (x3,r)

Calcolo della diagnosi



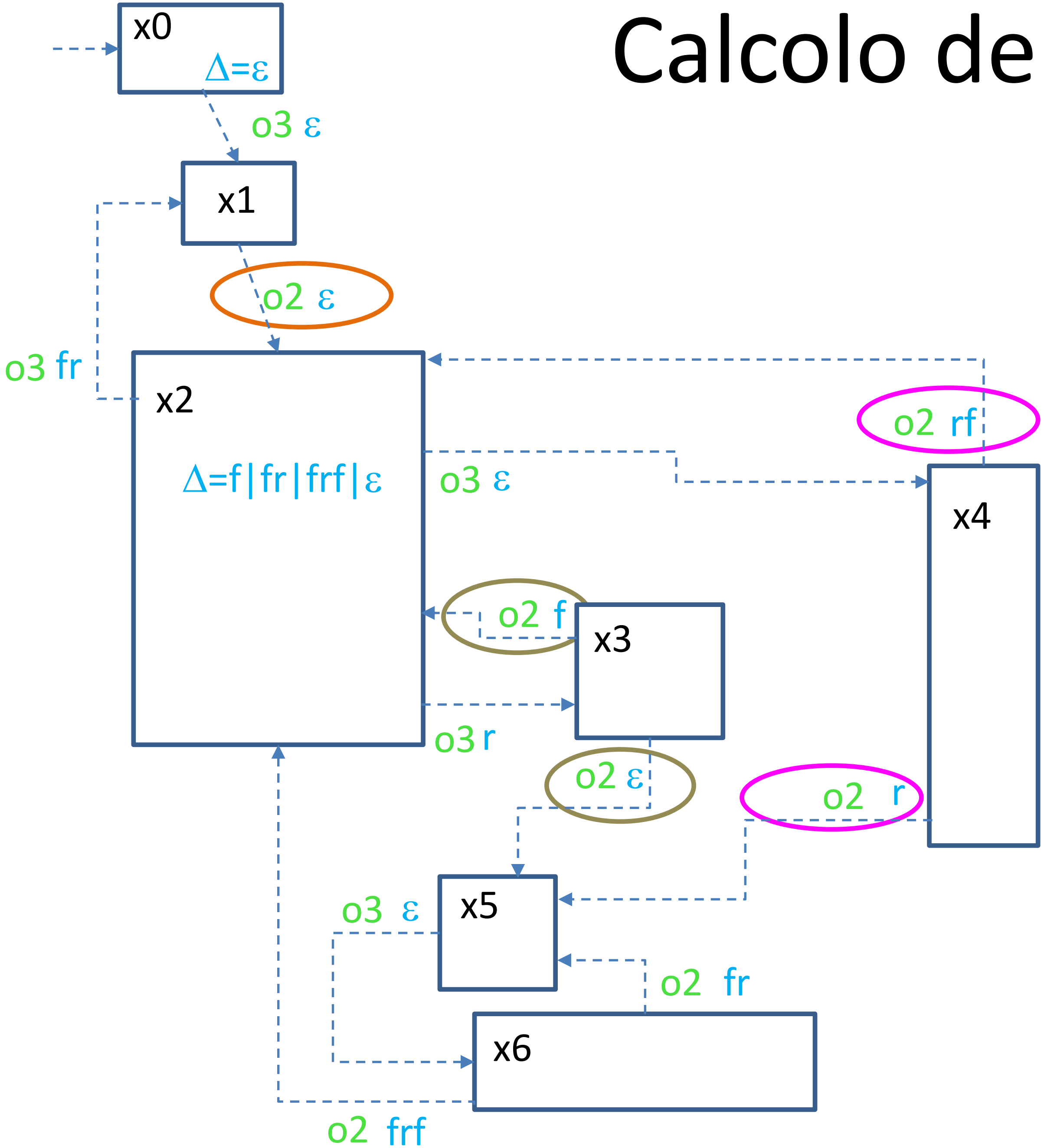
evento o2:

(x1,fr) o2 (x2,fr)

(x4,ε) o2 (x2,rf)
(x4,ε) o2 (x5,r)

(x3,r) o2 (x2,rf)
(x3,r) o2 (x5,r)

Calcolo della diagnosi



evento o2:

(x1,fr) o2 (x2,fr)

(x4,ε) o2 (x2,rf)

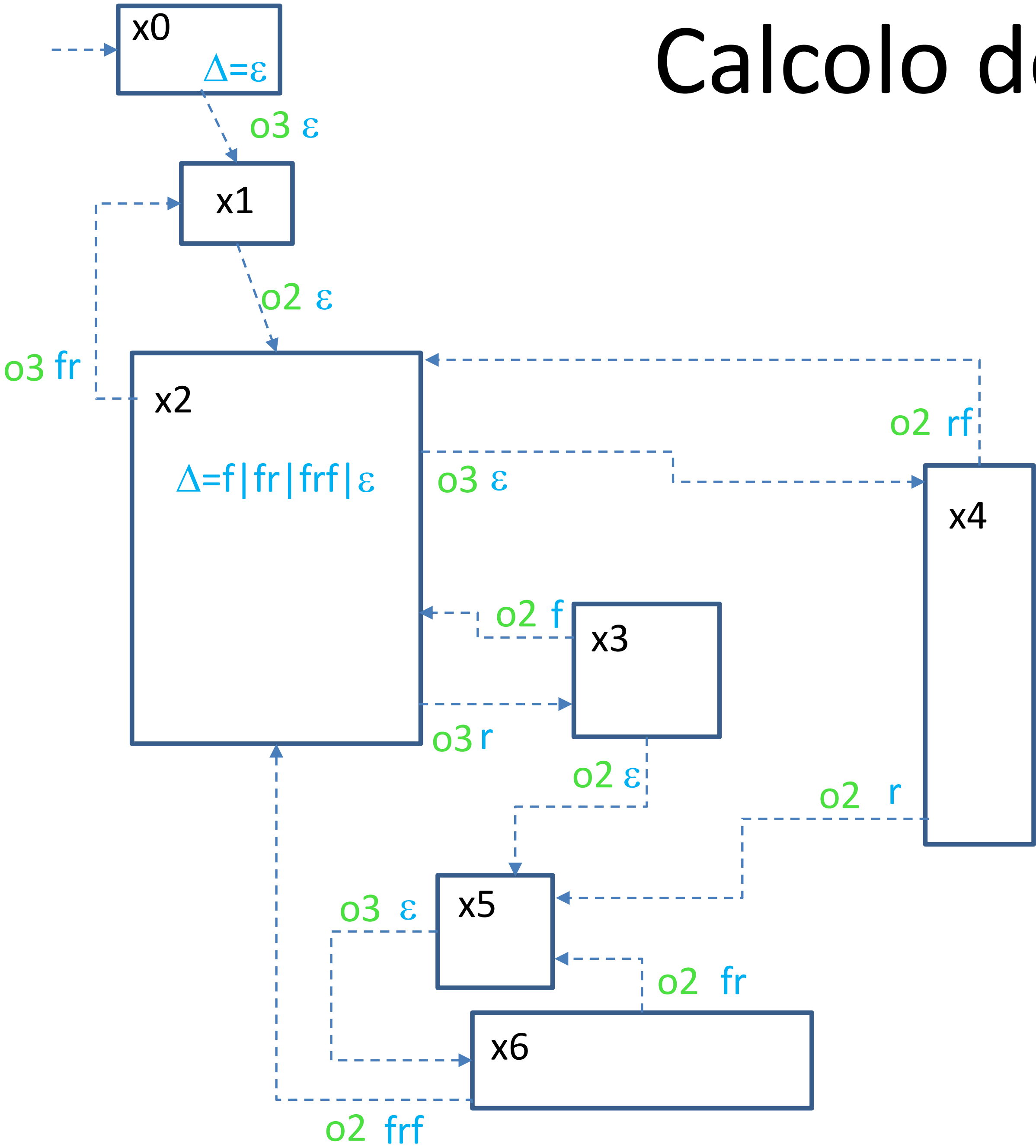
(x4,ε) o2 (x5,r)

(x3,r) o2 (x2,rf)

(x3,r) o2 (x5,r)

Queste sono le situazioni possibili al termine della generazione di O, si devono considerare solo le traiettorie (ovvero i cammini che finiscono in uno stato di accettazione - e pertanto in una chiusura finale)

Calcolo della diagnosi



(x2,fr)

(x2,rf)
~~(x5,r)~~

(x2,rf)
~~(x5,r)~~

Eliminiamo quindi le coppie relative a stati che non sono di accettazione. Poi, per ciascuna coppia (xi,ρ) rimasta (in questo esempio ne sono rimaste tre, di cui due identiche), concateniamo ρ con Δ(xi) :

(x2,fr)→fr(f|fr|frf|ε)
(x2,rf)→rf(f|fr|frf|ε)
La diagnosi relativa all'osservazione O è dunque (fr|rf)(f|fr|frf|ε)

Algoritmo DiagnosiLineare

- Per calcolare la diagnosi relativa a un’osservazione lineare data sfruttando il diagnosticatore è utilizzabile l’algoritmo DiagnosiLineare, che introduce qualche piccolo miglioramento nei passaggi illustrati sommariamente nelle pagine precedenti
- I parametri d’ingresso dell’algoritmo sono il diagnosticatore (Γ) e una osservazione lineare (O) della rete considerata
- L’uscita è un’espressione regolare di rilevanza che rappresenta la diagnosi relativa all’osservazione lineare data

Algoritmo DiagnosiLineare (cont.)

- Nello pseudocodice dell’algoritmo, fornito nella pagina seguente, ogni coppia (xi,ρ) è costituita dall’identificatore di uno stato del diagnosticatore e da una espressione regolare mentre un arco (cioè una transizione fra due stati del diagnosticatore, anche coincidenti) è indicato come $\langle x_1,(o,\rho),x_2\rangle$, dove x_1 e x_2 sono rispettivamente lo stato sorgente e destinazione dell’arco, o è l’evento osservabile dell’arco e ρ l’espressione regolare a esso associata

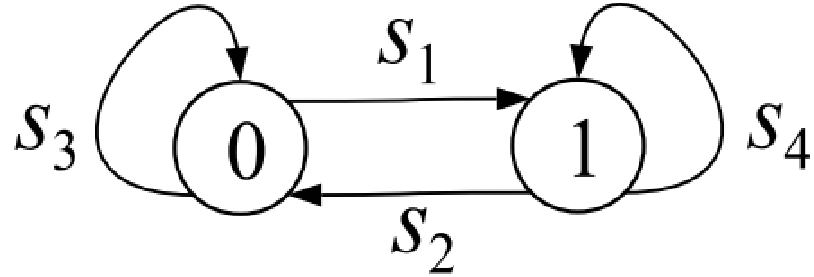
Algoritmo DiagnosiLineare(Γ,O)

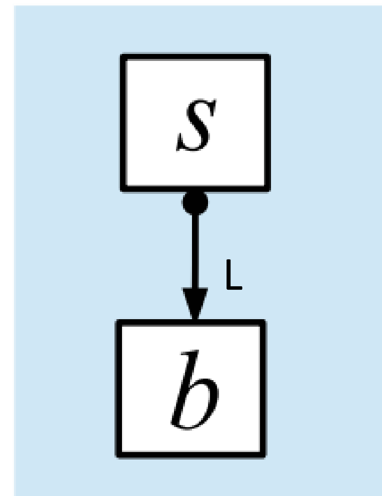
$\chi \leftarrow \{(x_0, \varepsilon)\}$
for all evento o entro l'osservazione O (nell'ordine di comparizione) **do**
    $\chi_{\text{new}} \leftarrow \emptyset$
    for all $(x', \rho') \in \chi$ **do**
        for all arco $\langle x', (o, \rho), x'_2 \rangle$ entro il diagnosticatore Γ **do**
            $\rho_2 \leftarrow \rho' \rho$
            if $(x'_2, \rho'_2) \in \chi_{\text{new}}$ **then**
                sostituire (x'_2, ρ'_2) con $(x'_2, (\rho'_2 | \rho_2))$ in χ_{new}
            else
                inserire (x'_2, ρ_2) in χ_{new}
            end if
        end for
    end for
    $\chi \leftarrow \chi_{\text{new}}$
end for
rimuovere da χ tutte le coppie (x_i, ρ_i) in cui x_i non è di accettazione
if $\chi = \{(x, \rho)\}$ **then**
    $R \leftarrow r\Delta(x)$
else if $\chi = \{(x_1, \rho_1), \ldots, (x_k, \rho_k)\}$ con $k > 1$ **then**
    $R \leftarrow (\rho_1(\Delta(x_1))) \mid \ldots \mid (\rho_k(\Delta(x_k)))$
end if
return R

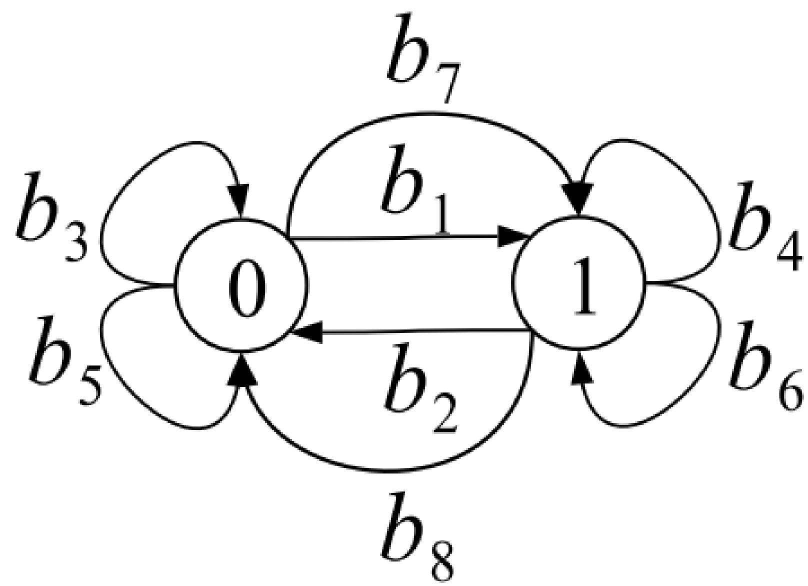
Compito

- Implementare l’algoritmo DiagnosiLineare
- Estendere l’applicazione di cui alle richieste già avanzate in modo che contempli anche il calcolo della diagnosi relativa a un’osservazione lineare data sfruttando il diagnosticatore della rete di FA comportamentali considerata

Altra rete







Transizioni

s

- s1: /{op(L)}
- s2: /{cl(L)}
- s3: /{cl(L)}
- s4: /{op(L)}

b

- b1: op(L)
- b2: cl(L)
- b3: op(L)
- b4: cl(L)
- b5: cl(L)
- b6: op(L)
- b7: cl(L)
- b8: op(L)

Osservabilità

s

- s_1: act
- s_2: sby
- s_3: ε
- s_4: ε

b

- b_1: opn
- b_2: cls
- b_3: ε
- b_4: ε
- b_5: nop
- b_6: nop
- b_7: opn
- b_8: cls

Rilevanza

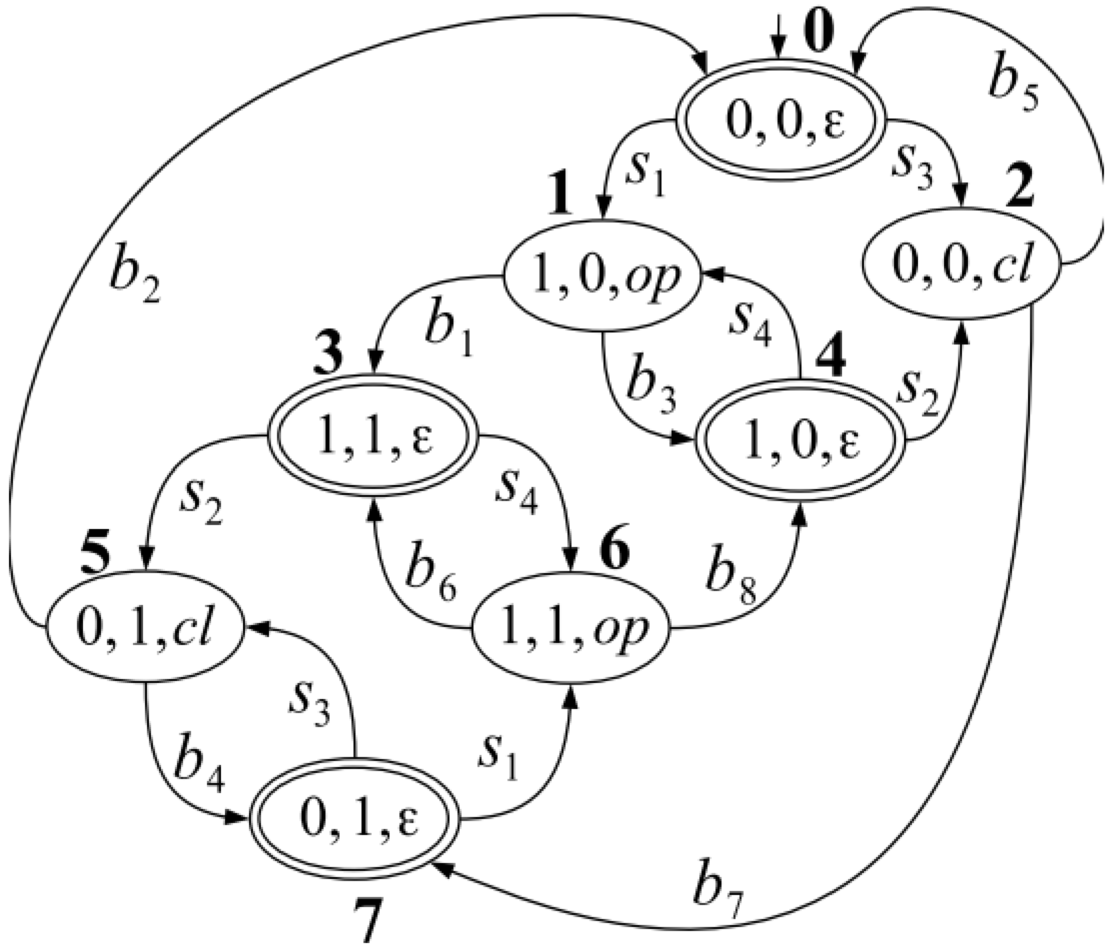
s

- s_1: ε
- s_2: ε
- s_3: f_1
- s_4: f_2

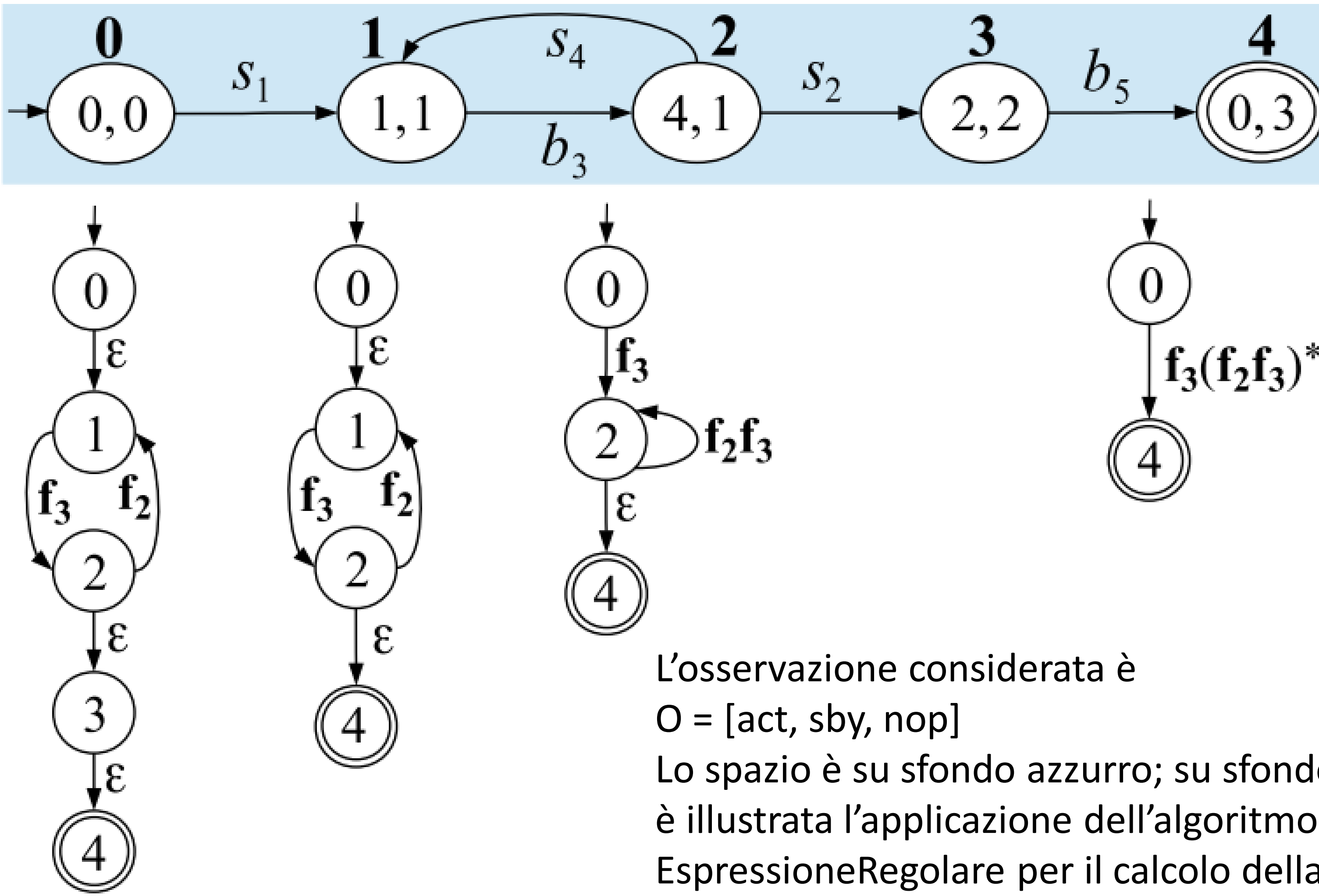
b

- b_1: ε
- b_2: ε
- b_3: f_3
- b_4: f_4
- b_5: ε
- b_6: ε
- b_7: f_5
- b_8: f_6

Spazio comportamentale

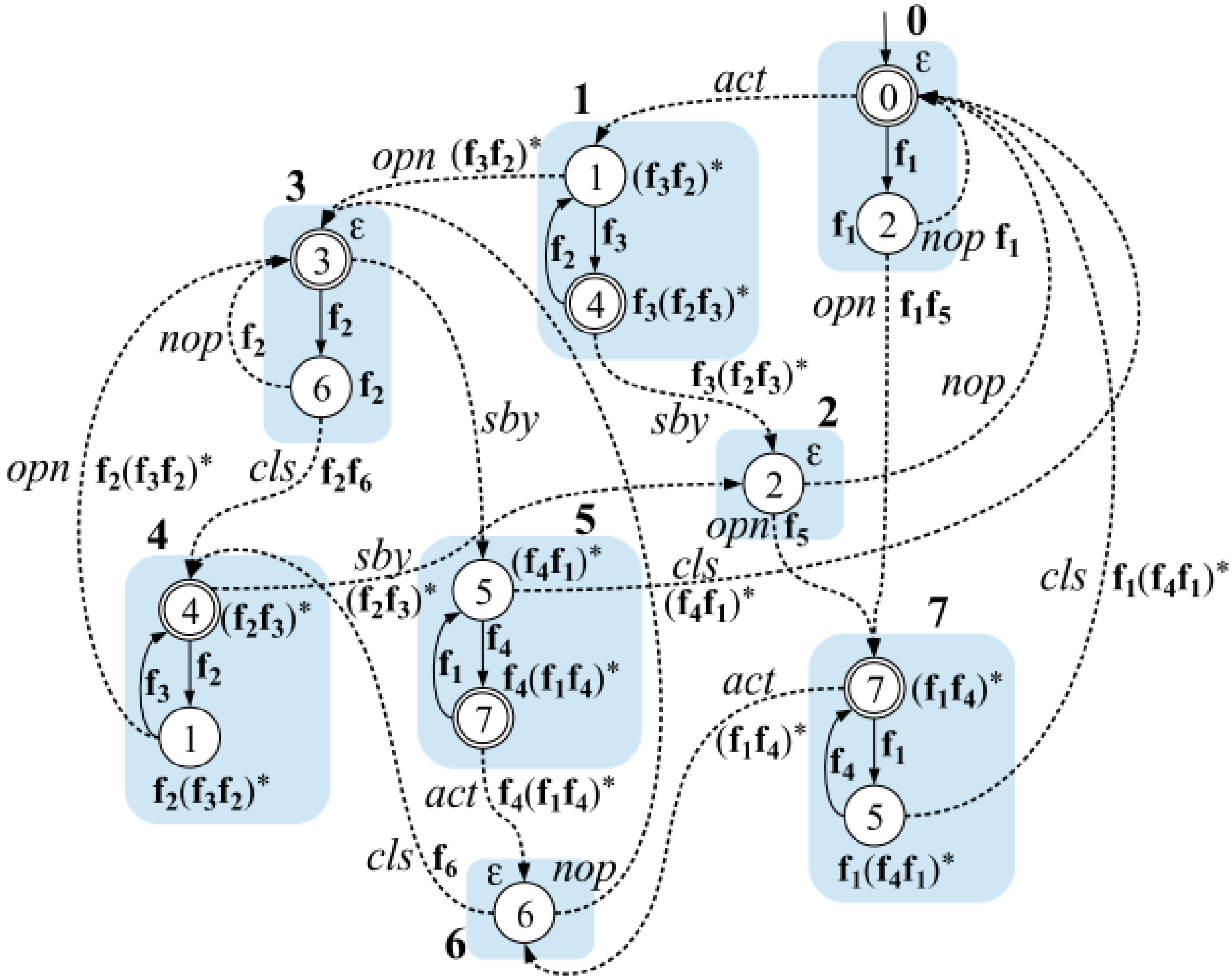


Spazio comportamentale relativo a una osservazione lineare e diagnosi



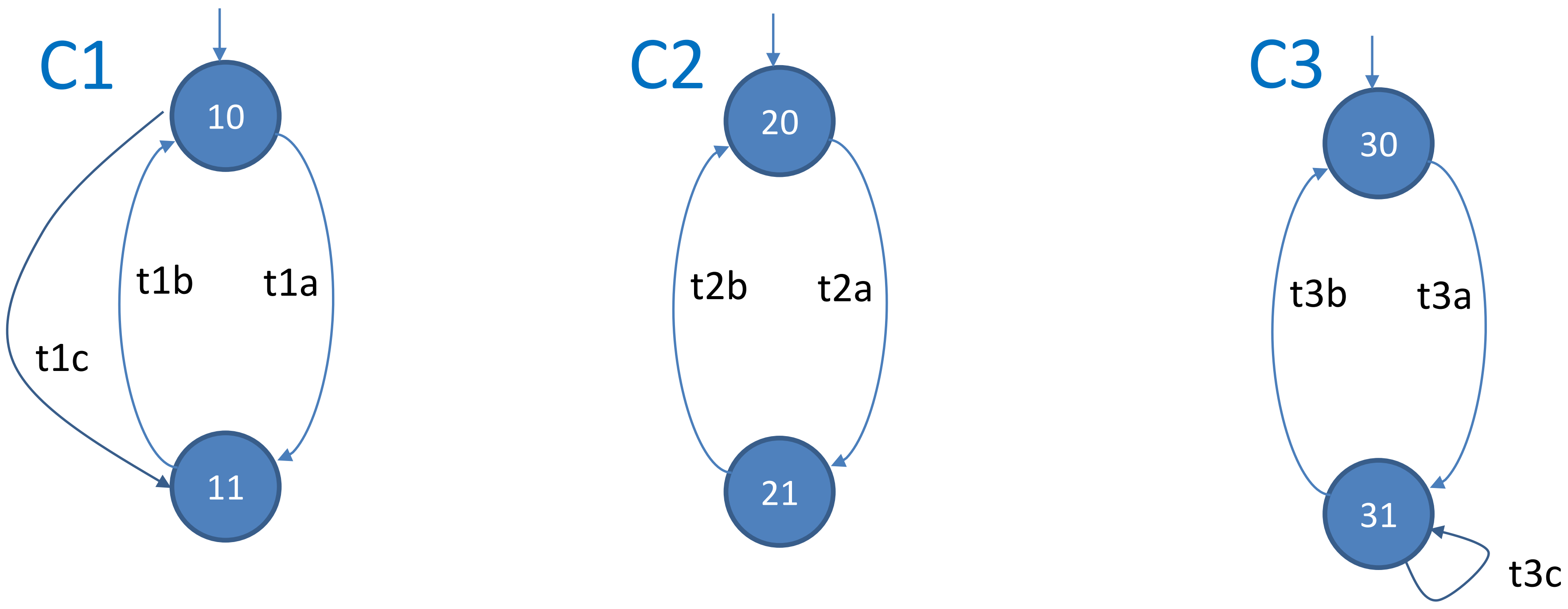
L’osservazione considerata è
O = [act, sby, nop]
Lo spazio è su sfondo azzurro; su sfondo bianco è illustrata l’applicazione dell’algoritmo EspressioneRegolare per il calcolo della diagnosi relativa a O

Diagnosticatore

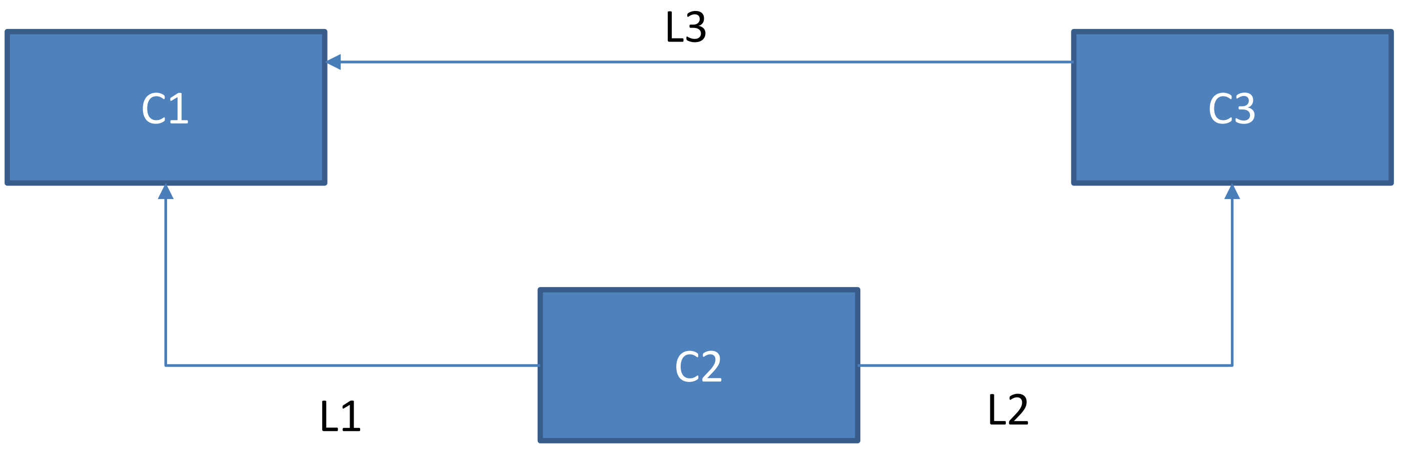


BENCHMARK

Tre automi a stati finiti



Topologia



Transizioni

C1

- t1a: e1(L1)
- t1b: e2(L3)
- t1c:

Evento in ingresso nullo e insieme vuoto di eventi in uscita

C2

- t2a: /{e1(L1),e3(L2)}
- t2b: /{e1(L1)}

C3

- t3a: /{e2(L3)}
- t3b: e3(L2)
- t3c: e3(L2)

Osservabilità

C1

- t1a: ε
- t1b: ε
- t1c: ε

C2

- t2a: o1
- t2b: o2

C3

- t3a: ε
- t3b: ε
- t3c: ε

Rilevanza

C1

- t1a: ε
- t1b: ε
- t1c: f1

C2

- t2a: ε
- t2b: ε

C3

- t3a: ε
- t3b: ε
- t3c: f3

SPERIMENTAZIONE

Interpretazione classica

- Una rete di FA può rappresentare un sistema (artificiale o naturale) o un processo
- Secondo l'interpretazione più classica, l'osservabilità indica cosa è visibile all'osservatore esterno (ad esempio, grazie alla presenza di sensori) dell'evoluzione dinamica (interna) del sistema/processo considerato
- Secondo l'interpretazione più classica, la rilevanza indica evoluzioni indesiderate (guasti, per questo si parla di diagnosi)

Consegne

- Si conduca una sperimentazione minima con tutti i risolutori realizzati utilizzando i casi di studio usati nell'arco della presentazione nonché quello fornito come benchmark
- Si conduca una sperimentazione su almeno un ulteriore esempio creato dal gruppo di lavoro, facendo variare le caratteristiche della rete (numero di componenti e di link, numero di stati per FA, ecc.), delle relazioni di osservabilità e rilevanza, nonché la lunghezza delle osservazioni lineari considerate
- Gli esempi possono tenere conto dell'interpretazione classica di cui alla pagina precedente ma anche fornire una interpretazione libera (secondo cui, della rete considerata, sono rilevanti le evoluzioni di interesse per l'attività che si vuole svolgere, che non deve essere necessariamente un'attività diagnostica)
- Saranno particolarmente apprezzati casi di studio reali(stici) e afferenti a domini insoliti

Consegne

- Un fine della sperimentazione è quello di registrare e valutare criticamente le prestazioni delle prove condotte, dal punto di vista sia temporale sia spaziale
- L’applicazione sviluppata potrebbe produrre anche informazioni intermedie (ad esempio, numero di stati e transizioni potati) reputate di interesse per la valutazione sperimentale
- La possibilità di determinare la diagnosi relativa a un’osservazione lineare secondo due metodi distinti (con e senza diagnosticatore) consente di <u>confrontare le prestazioni</u> (spaziali e temporali) e le uscite dei due metodi; se tali uscite differiscono, è stato compiuto un errore (logico o di programmazione). Tuttavia non è richiesto di effettuare un confronto automatico fra le uscite prodotte dai due metodi dal momento che ciò richiederebbe di valutare l’equivalenza di due espressioni regolari

RICHIESTE

Gruppi di lavoro

- Ogni gruppo, costituito da <u>due</u> studenti, deve portare a termine i compiti assegnati già descritti nelle sezioni precedenti e condurre una sperimentazione, di cui all’apposita sezione a essa dedicata, redigendo una relazione scritta che illustri il lavoro svolto, le scelte implementative compiute, le prove sperimentali eseguite e una valutazione critica delle stesse

Lavoro e relazione

- Particolare attenzione deve essere dedicata alla scelta di strutture dati volte a estendere le dimensioni degli spazi che possono essere elaborati nonché di altri accorgimenti aventi lo stesso fine. La relazione deve documentare tali scelte e accorgimenti
- Sono naturalmente apprezzati gli sforzi tesi a ridurre il costo temporale della computazione, che devono anch’essi essere documentati
- La relazione deve contenere ogni indicazione ritenuta utile al fine di consentire l’utilizzo dei programmi realizzati e la conduzione di ulteriori sperimentazioni
- La relazione deve evidenziare tutte le limitazioni riscontrate nelle prove di esecuzione effettuate

Requisiti funzionali

- L’applicazione software sviluppata deve consentire all’utente di:
  - considerare una nuova rete finita di FA comportamentali ogni volta che lo desideri
  - adottare più osservazioni lineari distinte per ciascuna singola rete
  - svolgere su ciascuna rete i compiti scelti fra quelli previsti
  - conservare in forma persistente ogni rete considerata, le osservazioni adottate per tale rete, i risultati prodotti dai compiti eseguiti (spazi, diagnosticatori e diagnosi), anche in sessioni di lavoro diverse
  - riusare i risultati salvati circa una rete per svolgere nuovi compiti (ad esempio, usare uno spazio comportamentale già memorizzato per produrre il diagnosticatore corrispondente; usare uno spazio comportamentale relativo a un’osservazione per calcolare la diagnosi inerente alla stessa)

Requisiti funzionali

- L’applicazione software sviluppata deve:
  – per ogni automa considerato, sia esso fornito in ingresso o prodotto in uscita (componente della rete, spazio, diagnosticatore, …), generare un riassunto che ne compendi le dimensioni (numero di stati e numero di transizioni)
  – consentire all’utente di interrompere il calcolo prima che esso sia concluso (o eventualmente di fissare una durata massima per l’elaborazione), fornendo in uscita i risultati parziali già calcolati, accompagnati anch’essi da un riassunto analogo a quello del punto precedente, esplicitando però per quali risultati il calcolo non è stato completato

Requisiti non funzionali

- Ai fini della presentazione all’utente di alcuni dei risultati dell’applicazione sviluppata (spazio comportamentale e diagnosticatore), è opportuno usare un linguaggio di rappresentazione di automi
- Tale linguaggio può essere adottato anche per gli FA in ingresso (componenti della rete)
- Se l’acquisizione in ingresso della descrizione di una rete di automi avviene da file (come sarebbe opportuno), il formato d’ingresso deve consentire anche la rappresentazione della topologia della rete e delle informazioni di osservabilità e rilevanza

Requisiti non funzionali

- Non è richiesta la realizzazione di GUI, l'elaborazione può felicemente essere batch (I/O solo da/per file)
- Non è imposto un linguaggio di programmazione, né un ambiente di sviluppo né un ambiente di destinazione

Consegna del materiale

- Ai fini del superamento della prova orale è necessario consegnare, entro i tempi indicati nelle note relative agli appelli, una cartella elettronica contenente codice sorgente, eventuale codice eseguibile, casi di studio, esiti delle prove sperimentali e relazione (in formato sia "sorgente", sia pfd)
- La consegna della cartella deve avvenire inviando via email il link alla stessa, creato usando un sistema di condivisione (ad es. dropbox), oppure attraverso la piattaforma Moodle