SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Dipartimento di Informatica

Linguaggi e Compilatori

Autore:
Simone Lidonnici

8 marzo 2025

Indice

E

Esercizi

E.1 Esercizi su automi e grammatiche

E.1.1 Trasformare un NFA in DFA

Dato un NFA $N = (Q_N, \Sigma, \delta_N, q_{0_N}, F_N)$ definiamo la ε-closure (o estensione) di uno stato $q \in Q_N$ come:

$$E(q) = \varepsilon\text{-closure}(q) = \left\{ s \in Q_N \,\middle|\, \begin{array}{c} s \text{ può essere raggiunto da } q \\ \text{tramite solamente } \varepsilon\text{-archi} \end{array} \right\}$$

La ε-closure di un insieme di stati $R \subseteq Q_N$ è definita come:

$$E(R) = \varepsilon\text{-closure}(R) = \bigcup_{q \in R} \varepsilon\text{-closure}(q)$$

La ε-closure di un insieme di stati R contiene sempre almeno R.
Per creare il DFA $D = (Q_D, \Sigma, \delta_D, q_{0_D}, F_D)$ equivalente all'NFA N, si esegue questo algoritmo:

Algoritmo: Subset Construction

```
def Subset_Construction(N):
    Q_D = {ε-closure(q_0N)}
    for R ∈ Q_D :  // anche quelli aggiunti durante il ciclo
        for a ∈ Σ :
            S = ε-closure(δ_N(R, a))
            if S ∉ Q_D :
                Aggiungo S a Q_D // Solitamente si numerano con A,B,...,Z
            Creo la transizione δ_D(R, a) = S
    return D
```

Quando uno stato $R \in Q_D$ è un insieme di stati in Q_N, la funzione δ_N viene calcolata come:

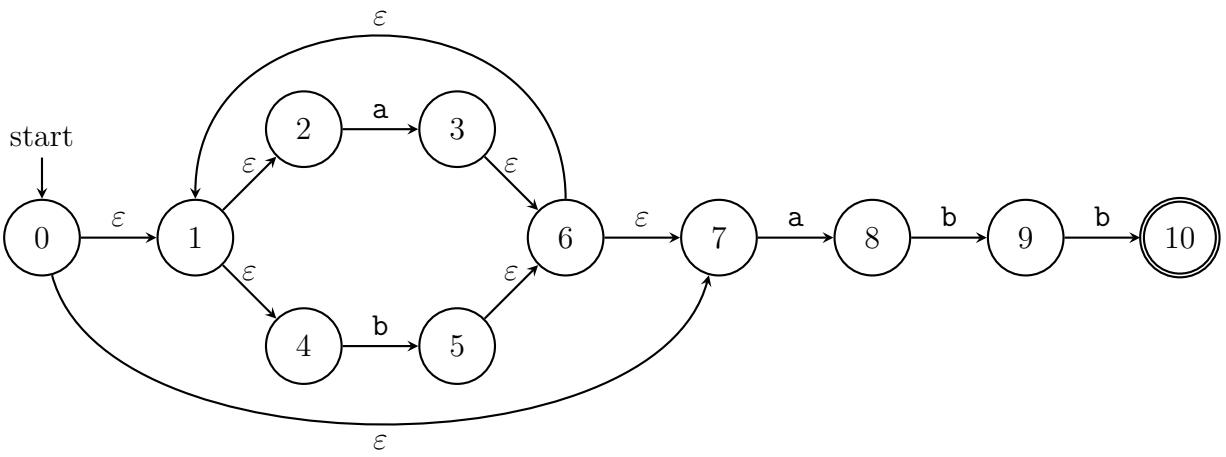
$$\delta_N(R, a) = \bigcup_{r \in R} \delta_N(r, a)$$

Gli stati accettanti in D sono tutti gli stati che contengono almeno uno stato accettante di N.
Il DFA risultante può essere disegnato oppure rappresentato come tabella con le intestazioni:

Stati NFA	Stato DFA	a	b
{0,1,2,3}	A	B	C
{1,2}	B	C	A
{3,4}	C	A	C

Con l'alfabeto $\Sigma = \{a, b\}$ la tabella avrebbe le intestazioni come sopra e la casella nella colonna a e riga A rappresenta la transizione $\delta_D(A, a)$. Nel caso l'alfabeto avesse altri simboli bisognerebbe aggiungere una colonna per ogni simbolo dell'alfabeto.

Esempio:
Dato l'NFA per il linguaggio $L = \{(a \cup b)^* abb\}$:



Eseguiamo i passaggi dell'algoritmo:

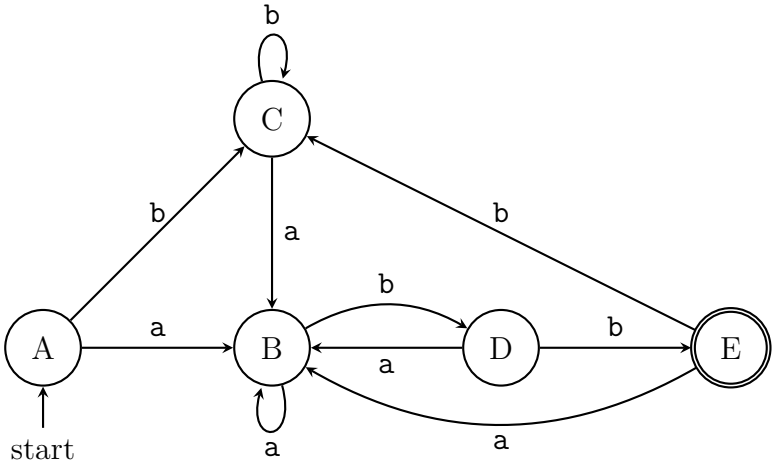
- Lo stato iniziale di D sarà ε-closure$(0) = \{0, 1, 2, 4, 7\} = A$
- Sullo stato A eseguiamo:
  1. ε-closure$(\delta_N(A, a)) = \varepsilon$-closure$(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\} = B$
     Avremo una transizione $\delta_D(A, a) = B$
  2. ε-closure$(\delta_N(A, b)) = \varepsilon$-closure$(\{5\}) = \{1, 2, 4, 5, 6, 7\} = C$
     Avremo una transizione $\delta_D(A, b) = C$
- Sullo stato B eseguiamo:
  1. ε-closure$(\delta_N(B, a)) = \varepsilon$-closure$(\{3, 8\}) = B$
     Avremo una transizione $\delta_D(B, a) = B$
  2. ε-closure$(\delta_N(B, b)) = \varepsilon$-closure$(\{5, 9\}) = \{1, 2, 4, 5, 6, 7, 9\} = D$
     Avremo una transizione $\delta_D(B, b) = D$
- Sullo stato C eseguiamo:
  1. ε-closure$(\delta_N(C, a)) = \varepsilon$-closure$(\{3, 8\}) = B$
     Avremo una transizione $\delta_D(C, a) = B$
  2. ε-closure$(\delta_N(C, b)) = \varepsilon$-closure$(\{5\}) = C$
     Avremo una transizione $\delta_D(C, b) = C$
- Sullo stato D eseguiamo:
  1. ε-closure$(\delta_N(D, a)) = \varepsilon$-closure$(\{3, 8\}) = B$
     Avremo una transizione $\delta_D(D, a) = B$
  2. ε-closure$(\delta_N(D, b)) = \varepsilon$-closure$(\{5, 10\}) = \{1, 2, 4, 5, 6, 7, 10\} = E$
     Avremo una transizione $\delta_D(D, b) = E$

- Sullo stato E eseguiamo:

  1. ε-closure$(\delta_N(E, a)) = \varepsilon$-closure$(\{3, 8\}) = B$
     Avremo una transizione $\delta_D(E, a) = B$
  2. ε-closure$(\delta_N(E, b)) = \varepsilon$-closure$(\{5\}) = C$
     Avremo una transizione $\delta_D(E, b) = C$

- Abbiamo finito gli stati da analizzare quindi l'algoritmo è terminato e lo stato accettante di D sarà E perchè è l'unico che contiene lo stato 10 di N.

Il DFA risultante rappresentato sotto forma di automa sarà quindi:



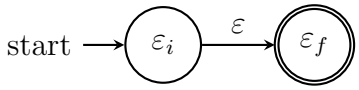
Sotto forma di tabella invece sarà:

Stati NFA	Stato DFA	a	b
{0,1,2,4,7}	A	B	C
{1,2,3,4,6,7,8}	B	B	D
{1,2,4,5,6,7}	C	B	C
{1,2,4,5,6,7,9}	D	B	E
{1,2,3,5,6,7,10}	E	B	C

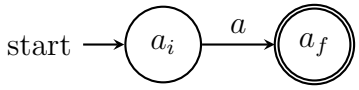
E.1.2 Trasformare un'espressione regolare in NFA

Data un'espressione regolare R e il suo Syntax Tree, possiamo trasformarlo in NFA eseguendo un visita in profondità del Syntax Tree e in base al nodo che visitiamo creaiamo un NFA parziale per il suo sottoalbero. Nella spiegazione dell'algoritmo i nodi s_i, s_f indicano il primo e l'ultimo nodo dell'NFA che riconosce s, negli esercizi solitamente si numerano sequanzialmente partendo da 0 in base all'ordine di visita. Eseguendo la visita in profondità l'NFA da creare cambia in base al simbolo contenuto nel nodo visitato:

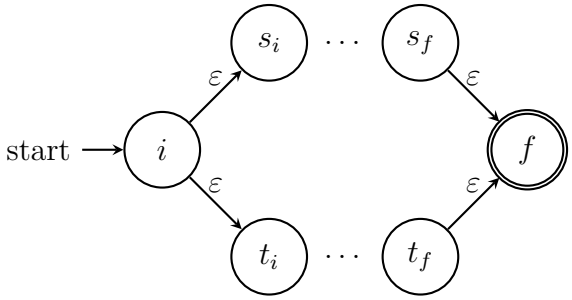
- Se stiamo visitando una foglia contente ε, costruiremo il seguente NFA:



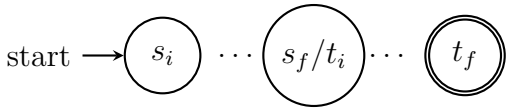
- Se stiamo visitando una foglia contente un simbolo $a \in \Sigma$, costruiremo il seguente NFA:



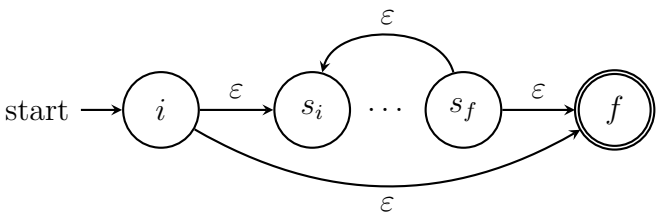
- Se stiamo visitando un nodo contente un'unione $s \cup t$, costruiremo il seguente NFA, aggiungendo gli stati i e f:



- Se stiamo visitando un nodo contente una concatenazione st, costruiremo il seguente NFA, unendo lo stato finale di s_f con lo stato iniziale di t_i:



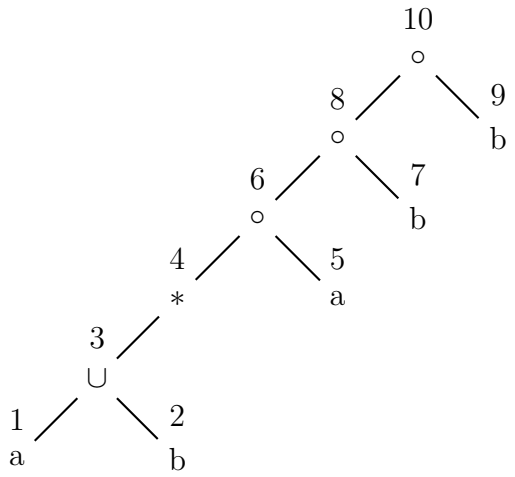
- Se stiamo visitando un nodo contente una star di Kleene s^*, costruiremo il seguente NFA, aggiungendo gli stati i e f:



L'NFA risultante avrà un solo stato accettante e ogni stato (ad eccezione di quello accettante) avrà le transizioni uscenti in uno di questi due modi:

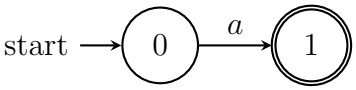
1. Una sola transizione con etichetta $a \in \Sigma$
2. Una o due transizioni con etichetta ε

Esempio:
Data l'espressione regolare $(a \cup b)^* abb$ con il seguente Abstract Tree ($\circ$ indica la concatenzazione):

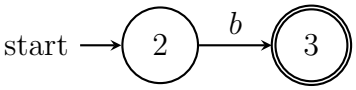


In questo esempio i nodi dell'albero sono numerati in base all'ordine in cui vanno visitati. Eseguiamo i passi dell'algoritmo, partendo dalla foglia in basso a sinistra:

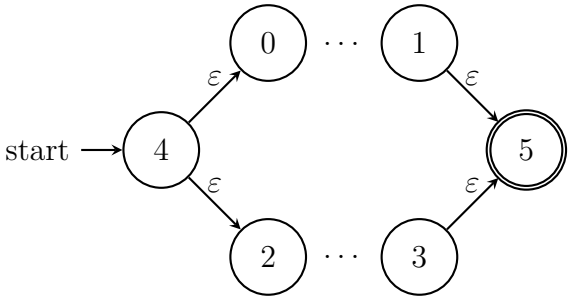
1. La foglia contiene a, quindi l'NFA corrispondente sarà:



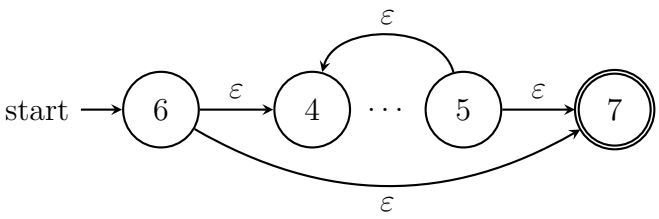
2. La foglia contiene b, quindi l'NFA corrispondente sarà:



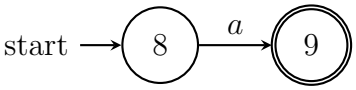
3. La foglia contiene un'unione, quindi l'NFA corrispondente a $a \cup b$ sarà:



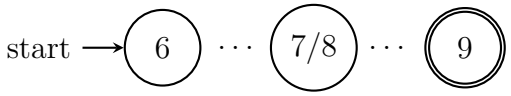
4. La foglia contiene una star di Kleene, quindi l'NFA corrispondente a $(a \cup b)^*$ sarà:



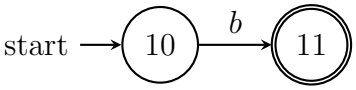
5. La foglia contiene a, quindi l'NFA corrispondente sarà:



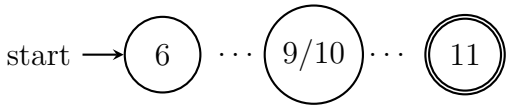
6. La foglia contiene una concatenazione, quindi l'NFA corrispondente a $(a \cup b)^*a$ sarà:



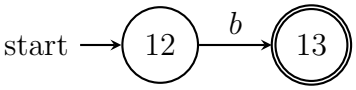
7. La foglia contiene b, quindi l'NFA corrispondente sarà:



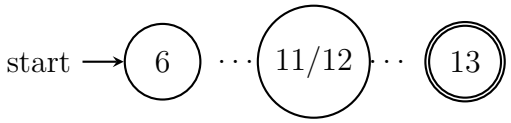
8. La foglia contiene una concatenazione, quindi l'NFA corrispondente a $(a \cup b)^*ab$ sarà:



9. La foglia contiene b, quindi l'NFA corrispondente sarà:



10. La foglia contiene una concatenazione, quindi l'NFA corrispondente a $(a \cup b)^*abb$ sarà:



L'automa finale disegnato completamente sarà: