Automi e Linguaggi Formali

Anno accademico 2005-2006

Docente: Francesca Rossi
E-mail: frossi@math.unipd.it

Orario:

- Lunedi' 15:30 – 17:30, LUM 250
- Venerdi' 13:30 – 16:30, LUM 250

Ricevimento: Giovedi',
16:00 - 18:00, studio

- Sito del corso:
www.math.unipd.it/~frossi/automi2006.html

Cambiamenti di orario

- non si fara' lezione Venerdi' 27 Gennaio

Altre informazioni utili

Libro di testo: J. E. Hopcroft, R. Motwani, and J. D. Ullman *Automi, linguaggi e calcolabilita'*, Addison-Wesley, 2003.
Si trova alla libreria Progetto (Via Marzolo o Via Portello).
Prezzo: circa 34,50 euro.

Compitini: Due compitini, uno a meta' del corso e uno alla fine. Possono sostituire lo scritto.

Esame: Scritto e, se richiesto dal docente, colloquio orale. Due appelli a fine corso. Due appelli di recupero: uno a Luglio e uno a Settembre.

Lucidi in inglese: sul sito
`http://www.cs.concordia.ca/~grahne/hmu_slides/`.
I lucidi che uso per i capitoli 1-7 sono basati sui lucidi in inglese di Gösta Grahne. Grazie anche David Ford per l'assistenza TEX.

Motivazione

- Automa = dispositivo astratto per fare delle computazioni
- Turing ha studiato e definito le "macchine di Turing" (= computer astratti) prima che esistessero veri calcolatori
- Studieremo anche dispositivi piu' semplici delle macchine di Turing (automi a stati finiti, automi a pila, . . .), e modi di definire linguaggi, come grammatiche e espressioni regolari.
- Problemi nella classe NP = che non possono essere risolti efficientemente

Automi a stati finiti

Gli automi a stati finiti sono usati come modello per

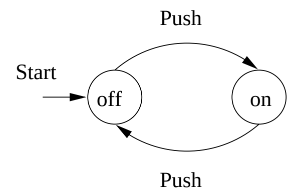
- Software per la progettazione di circuiti digitali.

- Analizzatori lessicali di un compilatore.

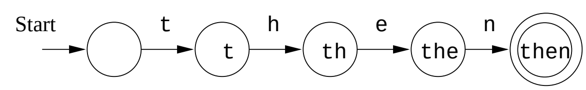
- Ricerca di parole chiave in un file o sul web.

- Software per verificare sistemi a stati finiti, come protocolli di comunicazione.

- Esempio: automa a stati finiti per un interruttore on/off



- Esempio: automa a stati finiti che riconosce la stringa `then`



Rappresentazioni strutturali

Ci sono vari modi di specificare una macchina

Grammatiche: Una regola come $E \Rightarrow E + E$ specifica un'spressione aritmetica

- $Coda \Rightarrow Persona.Coda$

dice che una coda e' costituita da una persona seguita da una coda.

Espressioni regolari: Denotano la struttura dei dati, per esempio:

`'[A-Z][a-z]*[][A-Z][A-Z]'`

e' compatibile con (matches) `Ithaca NY`

non e' compatibile con `Palo Alto CA`

Domanda: Quale espressione e' compatibile con

`Palo Alto CA`

Concetti di base

Alfabeto: Insieme fnito e non vuoto di simboli

Esempio: $\Sigma = \{0, 1\}$ alfabeto binario

Esempio: $\Sigma = \{a, b, c, \ldots, z\}$ insieme di tutte le lettere minuscole

Esempio: Insieme di tutti i caratteri ASCII

Stringa: Sequenza fnita di simboli da un alfabeto Σ, e.g. 0011001

Stringa vuota: La stringa con zero occorrenze di simboli da Σ

- La stringa vuota e' denotata con ϵ

Lunghezza di una stringa: Numero di posizioni per i simboli nella stringa.

$|w|$ denota la lunghezza della stringa w

$|0110| = 4, |\epsilon| = 0$

Potenze di un alpfabeto: Σ^k = insieme delle stringhe di lunghezza k con simboli da Σ

Example: $\Sigma = \{0, 1\}$

$\Sigma^1 = \{0, 1\}$

$\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^0 = \{\epsilon\}$

Domanda: Quante stringhe ci sono in Σ^3

L'insieme di tutte le stringhe su Σ e' denotato da Σ^*

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$

Anche:

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$

$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$

Concatenazione: Se x e y sono stringhe, allora xy e' la stringa ottenuta rimpiazzando una copia di y immediatamente dopo una copia di x

$x = a_1 a_2 \ldots a_i, y = b_1 b_2 \ldots b_j$

$xy = a_1 a_2 \ldots a_i b_1 b_2 \ldots b_j$

Esempio: $x = 01101, y = 110, xy = 01101110$

Nota: Per ogni stringa x

$$x\epsilon = \epsilon x = x$$

Linguaggi:

Se Σ e' un alfabeto, e $L \subseteq \Sigma^*$
allora L e' un linguaggio

Esempi di linguaggi:

- L'insieme delle parole italiane legali

- L'insieme dei programmi C legali

- L'insieme delle stringhe che consistono di n zeri seguiti da n uni

$$\{\epsilon, 01, 0011, 000111, \ldots\}$$

- L'insieme delle stringhe con un numero uguale di zeri e di uni

$$\{\epsilon, 01, 10, 0011, 0101, 1001, \ldots\}$$

- L_P = insieme dei numeri binari il cui valore e' primo

$$\{10, 11, 101, 111, 1011, \ldots\}$$

- Il linguaggio vuoto $\emptyset$

- Il linguaggio $\{\epsilon\}$ consiste della stringa vuota

Nota: $\emptyset \neq \{\epsilon\}$

Nota: L'alfabeto Σ e' sempre finito

Problema: La stringa w e' un elemento di un linguaggio L?

Esempio: Dato un numero binario, e' primo = e' un elemento di L_P?

E' 11101 $\in L_P$? Che risorse computazionali sono necessarie per rispondere a questa domanda?

Di solito non pensiamo ai problemi come delle decisioni si/no, ma come qualcosa che trasforma un input in un output.

Esempio: Fare il parsing di un programma C = controllare se il programma e' corretto, e se lo e', produrre un albero di parsing.

Introduzione informale agli automi

Protocollo per commercio elettronico usando soldi elettronici

Eventi permessi:

1. Il cliente puo' *pagare* il negozio (= spedire il file dei soldi al negozio)

2. Il cliente puo' *cancellare* i soldi (come fermare un assegno)

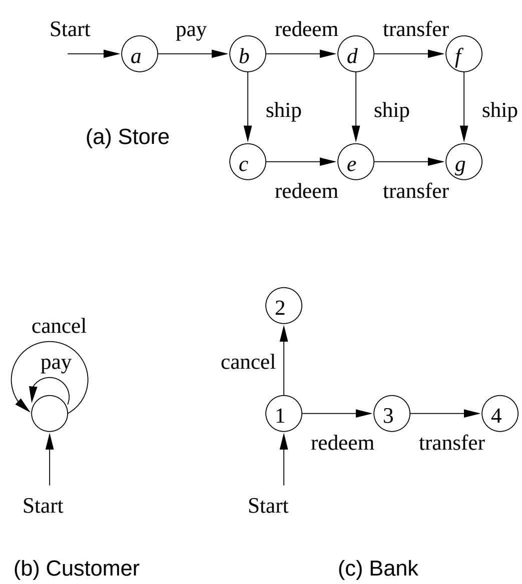
3. Il negozio puo' *spedire* la merce al cliente

4. Il negozio puo' *prendere* i soldi (= incassare un assegno)

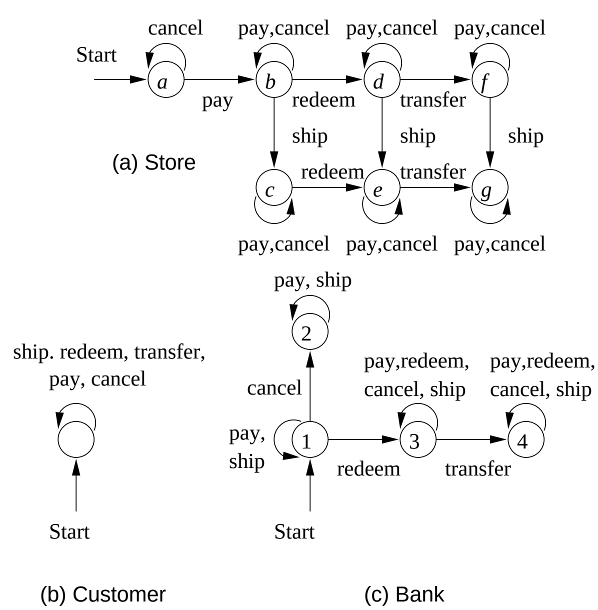
5. La banca puo' *trasferire* i soldi al negozio

Commercio elettronico

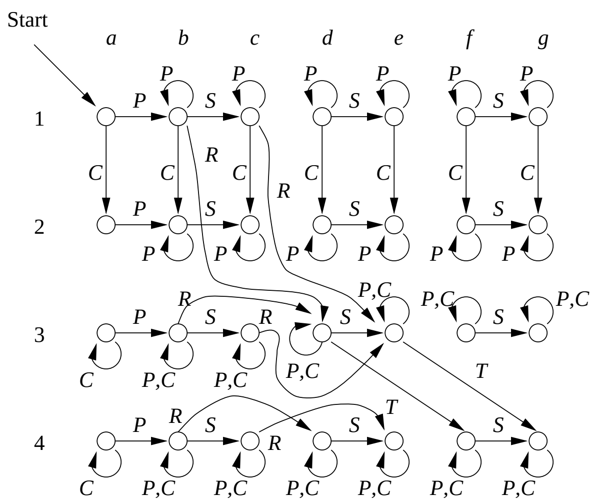
Il protocollo per ogni partecipante:



Protocolli completati:



L'intero sistema come automa:



Dimostrazioni deduttive

- Sequenza di enunciati la cui verita' porta da un enunciato iniziale (l'ipotesi) ad un enunciato finale (la conclusione)

- Forma del teorema: Se H, allora C

- H= ipotesi, C= conclusione

- Esempio: se $x \geq 4$, allora $2^x \geq x^2$

- x parametro quantificato universalmente (vale per tutti gli x)

- Modus ponens: regola logica che passare da un enuciato al successivo

  – Se H e' vera, e sappiamo che "se H allora C", allora possiamo concludere che anche C e' vera

- Teoremi della forma "C1 se e solo se C2": due direzioni di prova

- Dimostrazione per assurdo: H e non C implica il falso

Quantificatori

- Per ogni x ($\forall x$): vale per tutti i valori della variabile

- Esiste x ($\exists x$): vale per almeno un valore della variabile

- Esempio: un insieme s e' infinito se e solo se, per ogni intero n, esiste almeno un sottoinsieme T di S con n elementi

- Dobbiamo considerare un n arbitrario e poi trovare un insieme con quel numero n di elementi

- $\forall$ precede $\exists$

- Enunciato simile ma di significato diverso, e scorretto: Esiste un sottoinsieme T dell'insieme S tale che, per ogni n, T ha n elementi

Dimostrazioni per induzione

- Utili quando ci sono cose definite ricorsivamente
- Esempio: 0 e' un intero, e se n e' un intero allora n+1 e' un intero
- Induzione sugli interi: dobbiamo dimostrare un enunciato $S(n)$ su un intero n
  - Base: dimostriamo $S(i)$ per un intero particolare (0 o 1 di solito)
  - Passo induttivo: per $n \geq i$, dimostriamo che se vale $S(n)$ allora vale anche $S(n+1)$
- Possiamo concludere che $S(n)$ e' vero per ogni $n \geq i$

Esempio

- Se $x \geq 4$, allora $2^x \geq x^2$
- Base: $x = 4 \Rightarrow 2^x = 2^4 = 16$ e $x^2 = 4^2 = 16$
- Induzione: Supponiamo che $2^x \geq x^2$ per $x \geq 4$
- Dobbiamo dimostrare che $2^{x+1} \geq (x+1)^2$
- Abbiamo:
  - $2^{x+1} = 2^x \times 2 \geq x^2$ (dalla base)
  - Dimostriamo adesso che $2x^2 \geq (x+1)^2$
  - Semplificando: $x \geq 2 + 1/x$

  - Se $x \geq 4$, $1/x \leq 1/4 \Rightarrow 2 + 1/x \leq 2.25$

Induzione strutturale

- Molte strutture possono essere definite ricorsivamente

- Esempio (espressioni):
  - caso base: qualunque numero o lettera e' un'espressione
  - caso induttivo: se E e F sono espressioni, allora lo sono anche $E+F$, $E\times F$, e (E)
  - Esempi: 3+(4x2), (2x(5+7))x4

- Per dimostrare teoremi su un'espressione: si dimostra l'enunciato sul caso base, e poi si dimostra l'enunciato sulla struttura X a partire dalla validita' dell'enunciato sulle strutture di cui X e' composta secondo la definizione ricorsiva

Esempio

- Teorema: ogni espressione ha un numero uguale di parentesi aperte e chiuse

- Caso base: zero parentesi $\Rightarrow$ vero

- Induzione: Tre modi per costruire un'espressione induttivamente: $E+F$, $E\times F$, e (E)

- Per $E+F$ e $E\times F$: se vale per E e F, supponiamo che E abbia n parentesi aperte e chiuse e F ne abbia m $\Rightarrow$ $E+F$ ne ha $n+m$

- Per (E): se vale per E, supponiamo che E abbia n parentesi aperte e chiuse $\Rightarrow$ (E) ne ha $n+1$

Automi a stati finiti deterministici

Un DFA e' una quintupla

$$A = (Q, \Sigma, \delta, q_0, F)$$

- Q e' un insieme finito di *stati*

- Σ e' un *alfabeto finito* (= simboli in input)

- δ e' una *funzione di transizione* $(q,a) \mapsto p$

- $q_0 \in Q$ e' lo *stato iniziale*

- $F \subseteq Q$ e' un insieme di *stati finali*

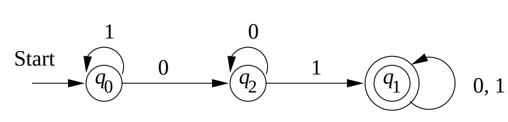
Esempio: Un automa A che accetta

$L = \{x01y : x, y \in \{0,1\}^*\}$

L'automa $A = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$ come una *tabella di transizione*:

	0	1
$\rightarrow q_0$	q_2	q_0
$\star q_1$	q_1	q_1
q_2	q_2	q_1

L'automa come un *diagramma di transizione*:



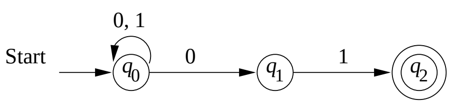
Un automa a stati finiti (FA) *accetta* una stringa $w = a_1 a_2 \cdots a_n$ se esiste un cammino nel diagramma di transizione che

1. Inizia nello stato iniziale

2. Finisce in uno stato finale (di accettazione)

3. Ha una sequanza di etichette $a_1 a_2 \cdots a_n$

Esempio: L'automa a stati finiti



accetta ad esempio la stringa 01101

- La funzione di transizione δ puo' essere estesa a $\hat{\delta}$ che opera su stati e stringhe (invece che su stati e simboli)

Base: $\hat{\delta}(q, \epsilon) = q$

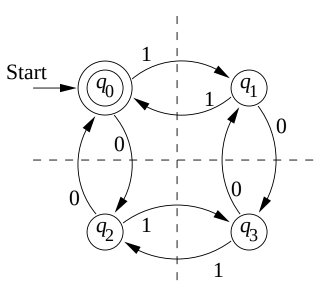
Induzione: $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$

- Formalmente, il *linguaggio accettato da* A e'

$$L(A) = \{w : \hat{\delta}(q_0, w) \in F\}$$

- I linguaggi accettati da automi a stati finiti sono detti *linguaggi regolari*

Esempio: DFA che accetta tutte e sole le stringhe con un numero pari di zeri e un numero pari di uni



Rappresentazione tabulare dell'automa

	0	1
$\star \rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

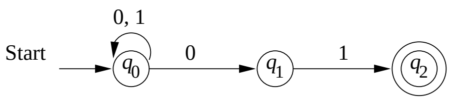
Esercizi

- DFA per i seguenti linguaggi sull'alfabeto $\{0, 1\}$:
  - Insieme di tutte le strighe che finiscono con 00
  - Insieme di tutte le stringhe con tre zeri consecutivi
  - Insieme delle stringhe con 011 come sottostringa
  - Insieme delle stringhe che cominciano o finiscono (o entrambe le cose) con 01

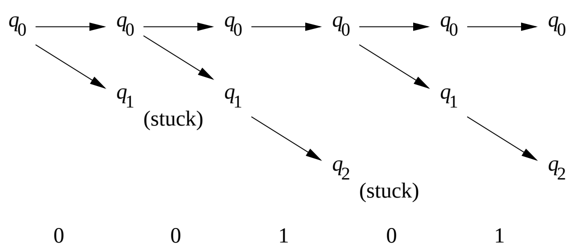
Automi a stati finiti non deterministici (NFA)

Un NFA puo' essere in vari stati nello stesso momento, oppure, visto in un altro modo, puo' ''scommettere'' su quale sara' il prossimo stato

Esempio: un automa che accetta tutte e solo le stringhe che finiscono in 01.



Ecco cosa succede quando l'automa elabora l'input 00101



Formalmente, un NFA e' una quintupla

$$A = (Q, \Sigma, \delta, q_0, F)$$

- Q e' un insieme finito di stati
- Σ e' un alfabeto finito
- δ e' una funzione di transizione da $Q \times \Sigma$ all'insieme dei sottoinsiemi di Q
- $q_0 \in Q$ e' lo *stato iniziale*
- $F \subseteq Q$ e' un insieme di *stati finali*

Esempio: L' NFA di due pagine fa e'

$$(\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$

dove δ e' la funzione di transizione

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$\star q_2$	$\emptyset$	$\emptyset$

Funzione di transizione estesa $\widehat{\delta}$.

Base: $\widehat{\delta}(q, \epsilon) = \{q\}$

Induzione:

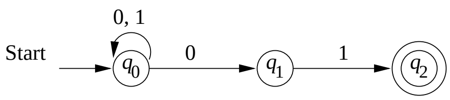
$$\widehat{\delta}(q, xa) = \bigcup_{p \in \widehat{\delta}(q,x)} \delta(p, a)$$

Esempio: Calcoliamo $\widehat{\delta}(q_0, 00101)$ sulla lavagna

- Formalmente, il *linguaggio accettato da* A e'

$$L(A) = \{w : \widehat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

Proviamo formalmente che l' NFA



accetta il linguaggio $\{x01 : x \in \Sigma^*\}$. Faremo una induzione mutua sui tre enunciati seguenti

0. $w \in \Sigma^* \Rightarrow q_0 \in \hat{\delta}(q_0, w)$

1. $q_1 \in \hat{\delta}(q_0, w) \Leftrightarrow w = x0$

2. $q_2 \in \hat{\delta}(q_0, w) \Leftrightarrow w = x01$

Base: Se $|w| = 0$ allora $w = \epsilon$. Allora l'enunciato (0) segue dalla defnizione Per (1) e (2) entrambi i lati sono falsi per ϵ

Induzione: Assumiamo che $w = xa$, dove $a \in \{0, 1\}$, $|x| = n$ e gli enunciati (0)–(2) valgono per x. Si mostra che gli enunciati valgono per xa.

Equivalenza di DFA e NFA

• Gli NFA sono di solito piu' facili da "programmare".

• Sorprendentemente, per ogni NFA N c'e' un DFA D, tale che $L(D) = L(N)$, e viceversa.

• Questo comporta una *construzione a sottonsiemi*, un esempio importante di come un automa B puo' essere costruito da un altro automa A.

• Dato un NFA

$$N = (Q_N, \Sigma, \delta_N, q_0, F_N)$$

costruiremo un DFA

$$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$$

tali che

$$L(D) = L(N)$$

.

I dettagli della costruzione a sottoinsiemi:

• $Q_D = \{S : S \subseteq Q_N\}$.

Nota: $|Q_D| = 2^{|Q_N|}$, anche se la maggior parte degli stati in Q_D sono "garbage", cioe' non raggiungibili dallo stato iniziale.

• $F_D = \{S \subseteq Q_N : S \cap F_N \neq \emptyset\}$

• Per ogni $S \subseteq Q_N$ e $a \in \Sigma$,

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$

Costruiamo δ_D dall' NFA gia' visto:

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$\star\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\star\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\star\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$\star\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Nota: Gli stati di D corrispondono a sottoinsiemi di stati di N, ma potevamo denotare gli stati di D in un altro modo, per esempio $A-F$.

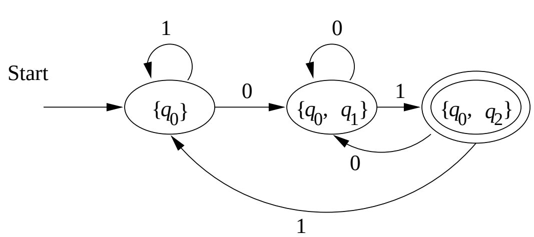
	0	1
A	A	A
$\rightarrow B$	E	B
C	A	D
$\star D$	A	A
E	E	F
$\star F$	E	B
$\star G$	A	D
$\star H$	E	F

Possiamo spesso evitare la crescita esponenziale degli stati costruendo la tabella di transizione per D solo per stati accessibili S come segue:

Base: $S = \{q_0\}$ e' accessibile in D

Induzione: Se lo stato S e' accessibile, lo sono anche gli stati in $\bigcup_{a \in \Sigma} \delta_D(S, a)$.

Esempio: Il "sottoinsieme" DFA con stati accessibili solamente.



Teorema 2.11: Sia D il DFA ottenuto da un NFA N con la costruzione a sottoinsiemi. Allora $L(D) = L(N)$.

Prova: Prima mostriamo per induzione su $|w|$ che

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

Base: $w = \epsilon$. L'enunciato segue dalla definizione.

Induzione:

$$
\begin{aligned}
\widehat{\delta}_D(\{q_0\},xa) &\stackrel{\text{def}}{=} \delta_D(\widehat{\delta}_D(\{q_0\},x),a) \\
&\stackrel{\text{i.h.}}{=} \delta_D(\widehat{\delta}_N(q_0,x),a) \\
&\stackrel{\text{cst}}{=} \bigcup_{p\in\widehat{\delta}_N(q_0,x)} \delta_N(p,a) \\
&\stackrel{\text{def}}{=} \widehat{\delta}_N(q_0,xa)
\end{aligned}
$$

Ora segue che $L(D) = L(N)$.

Teorema 2.12: Un linguaggio L e' accettato da un DFA se e solo se L e' accettato da un NFA.

Prova: La parte "se" e' il Teorema 2.11.

Per la parte "solo se" notiamo che un qualsiasi DFA puo' essere convertito in un NFA equivalente modificando la δ_D in δ_N secondo la regola seguente:

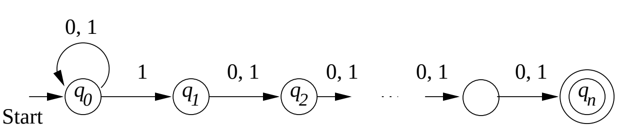
- Se $\delta_D(q,a) = p$, allora $\delta_N(q,a) = \{p\}$.

Per induzione su $|w|$ si puo' mostrare che se $\widehat{\delta}_D(q_0,w) = p$, allora $\widehat{\delta}_N(q_0,w) = \{p\}$.

L'enunciato del teorema segue.

Crescita esponenziale degli stati

Esiste un NFA N con $n+1$ stati che non ha nessun DFA equivalente con meno di 2^n stati



$$L(N) = \{x1c_2c_3\cdots c_n : x \in \{0,1\}^*, c_i \in \{0,1\}\}$$

Supponiamo che esista un DFA equivalente con meno di 2^n stati.

D deve ricordare gli ultimi n simboli che ha letto.

Ci sono 2^n sequenze di bit $a_1a_2\cdots a_n$

$$
\begin{aligned}
\exists\, q, a_1a_2\cdots a_n, b_1b_2\cdots b_n : q &\in \widehat{\delta}_N(q_0, a_1a_2\cdots a_n), \\
q &\in \widehat{\delta}_N(q_0, b_1b_2\cdots b_n), \\
a_1a_2\cdots a_n &\neq b_1b_2\cdots b_n
\end{aligned}
$$

Caso 1:

$1a_2\cdots a_n$
$0b_2\cdots b_n$

Allora q deve essere sia uno stato di accettazione che uno stato di non accettazione.

Caso 2:

$a_1\cdots a_{i-1}1a_{i+1}\cdots a_n$
$b_1\cdots b_{i-1}0b_{i+1}\cdots b_n$

Ora $\widehat{\delta}_N(q_0, a_1\cdots a_{i-1}1a_{i+1}\cdots a_n0^{i-1}) =$
$\widehat{\delta}_N(q_0, b_1\cdots b_{i-1}0b_{i+1}\cdots b_n0^{i-1})$

e $\widehat{\delta}_N(q_0, a_1\cdots a_{i-1}1a_{i+1}\cdots a_n0^{i-1}) \in F_D$

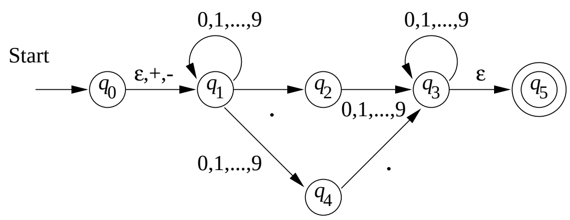
$\widehat{\delta}_N(q_0, b_1\cdots b_{i-1}0b_{i+1}\cdots b_n0^{i-1}) \notin F_D$

FA con transizioni epsilon

Un ϵ-NFA che accetta numeri decimali consiste di:

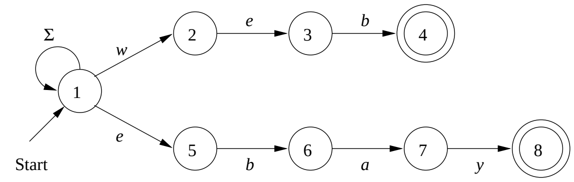
1. Un segno + o -, opzionale
2. Una stringa di cifre decimali
3. un punto decimale
4. un'altra stringa di cifre decimali

Una delle stringhe (2) e (4) sono opzionali



Esempio:

ϵ-NFA che accetta l'insieme di parole chiave $\{\texttt{ebay}, \texttt{web}\}$



Un ϵ-NFA e' una quintupla $(Q, \Sigma, \delta, q_0, F)$ dove δ e' una funzione da $Q \times \Sigma \cup \{\epsilon\}$ all'insieme dei sottoinsiemi di Q.

Esempio: L' ϵ-NFA della pagina precedente

$$E = (\{q_0, q_1, \ldots, q_5\}, \{., +, -, 0, 1, \ldots, 9\}\ \delta, q_0, \{q_5\})$$

dove la tabella delle transizioni per δ e'

	ϵ	+,-	.	$0, \ldots, 9$
$\rightarrow q_0$	$\{q_1\}$	$\{q_1\}$	$\emptyset$	$\emptyset$
q_1	$\emptyset$	$\emptyset$	$\{q_2\}$	$\{q_1, q_4\}$
q_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_3	$\{q_5\}$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_4	$\emptyset$	$\emptyset$	$\{q_3\}$	$\emptyset$
$\star q_5$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Epsilon-chiusura

Chiudiamo uno stato aggiungendo tutti gli stati raggiungibili da lui tramite una sequenza $\epsilon\epsilon\cdots\epsilon$

Definizione induttiva di ECLOSE(q)

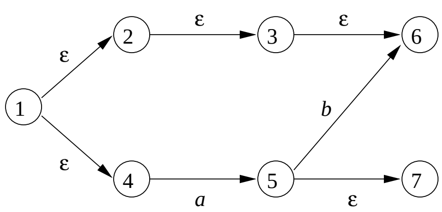
Base:

$q \in$ ECLOSE(q)

Induzione:

$p \in$ ECLOSE(q) and $r \in \delta(p, \epsilon) \Rightarrow r \in$ ECLOSE(q)

Esempio di ϵ-closure



Per esempio,

$$\text{ECLOSE}(1) = \{1, 2, 3, 4, 6\}$$

- Definizione induttiva di $\widehat{\delta}$ per automi ϵ-NFA

Base:

$$\widehat{\delta}(q, \epsilon) = \text{ECLOSE}(q)$$

Induzione:

$$\widehat{\delta}(q, xa) = \bigcup_{p \in \delta(\widehat{\delta}(q,x),a)} \text{ECLOSE}(p)$$

Calcoliamo
$\widehat{\delta}(q_0, 5.6)$ per l'NFA della pagina 40

Dato un ϵ-NFA

$$E = (Q_E, \Sigma, \delta_E, q_0, F_E)$$

costruiremo un DFA

$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

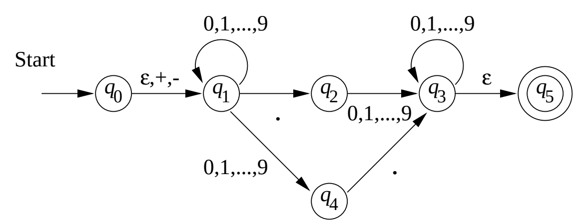
tale che

$$L(D) = L(E)$$

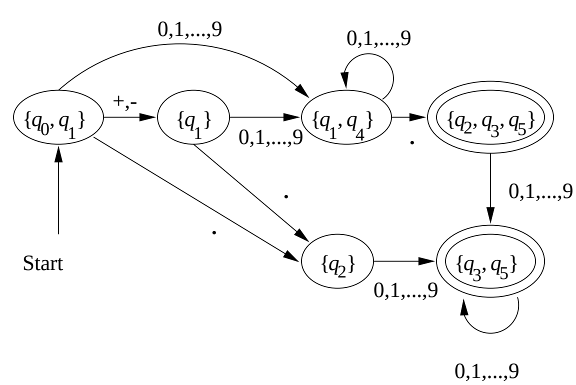
Dettagli della costruzione:

- $Q_D = \{S : S \subseteq Q_E$ e $S = \text{ECLOSE}(S)\}$
- $q_D = \text{ECLOSE}(q_0)$
- $F_D = \{S : S \in Q_D$ e $S \cap F_E \neq \emptyset\}$
- $\delta_D(S, a) =$
  $\bigcup \{\text{ECLOSE}(p) : p \in \delta(t, a)$ per alcuni $t \in S\}$

Esempio: ϵ-NFA E



DFA D corrispondente ad E



Teorema 2.22: Un linguaggio L e' accettato da un ϵ-NFA E se e solo se L e' accettato da un DFA.

Prova: Usiamo D costruito come sopra e mostriamo per induzione che $\widehat{\delta}_D(q_0, w) = \widehat{\delta}_E(q_D, w)$

Base: $\widehat{\delta}_E(q_0, \epsilon) = \text{ECLOSE}(q_0) = q_D = \widehat{\delta}(q_D, \epsilon)$

Induzione:

$$
\begin{aligned}
\widehat{\delta}_E(q_0, xa) &= \bigcup_{p \in \delta_E(\widehat{\delta}_E(q_0,x),a)} \text{ECLOSE}(p) \\
&= \bigcup_{p \in \delta_D(\widehat{\delta}_D(q_D,x),a)} \text{ECLOSE}(p) \\
&= \bigcup_{p \in \widehat{\delta}_D(q_D,xa)} \text{ECLOSE}(p) \\
&= \widehat{\delta}_D(q_D, xa)
\end{aligned}
$$