Linguaggi formali, automi e logiche

Angelo Montanari

1 Automi a stati finiti su parole finite

In questo capitolo vengono richiamati gli elementi di base della teoria degli automi a stati finiti (*automi finiti*, per brevità) su parole finite. Sia data la nozione primitiva di simbolo. Una *parola* (o stringa) finita è una sequenza finita di simboli giustapposti. Ad esempio, *acba* è una parola ottenuta utilizzando i simboli a, b e c. Definiamo *lunghezza* di una parola w, notazione $|w|$, il numero di simboli che la compongono. La *parola vuota*, denotata da ϵ, è la stringa composta da 0 simboli. Dalla definizione di lunghezza di una parola, segue che $|\epsilon| = 0$.

La *concatenazione* di due parole u e v, notazione $u \cdot v$ (uv per brevità), è (definita come) la loro giustapposizione. Siano dati due insiemi A e B di parole. L'insieme concatenazione di A e B è $A \cdot B = \{uv : \ u \in A, \ v \in B\}$. Per ogni $n \geq 0$, definiamo ricorsivamente l'insieme A^n nel seguente modo: $A^0 = \{\epsilon\}$, $A^{n+1} = A \cdot A^n$. L'insieme chiusura (di Kleene) di A è $A^* = \bigcup_{n \geq 0} A^n$. Definiamo inoltre l'insieme $A^+ = A^* \setminus \{\epsilon\}$. Dato un insieme finito di simboli A, detto *alfabeto*, una parola su A è un elemento di A^* e un linguaggio (di parole finite) su A è un sottoinsieme di A^*.

1.1 Automi finiti deterministici e nondeterministici

Definizione 1.1 (*Automa finito deterministico*)

Un automa finito deterministico (DFA) $\mathcal{A}$ è una quintupla (Q, A, δ, q_0, F), dove Q è un insieme finito di stati, A è un alfabeto finito, q_0 è un elemento di Q, detto stato iniziale*, $\delta : \ Q \times A \to Q$ è una funzione, detta* funzione di transizione*, che riceve in ingresso uno stato e un simbolo dell'alfabeto e restituisce in uscita uno stato, e F è un sottoinsieme di Q, detto* insieme degli stati finali*.*

La funzione di transizione $\hat{\delta}$ estende δ sostituendo A con A^* ed è definita nel seguente modo: $\hat{\delta}(q, \epsilon) = q$ e, per ogni parola w e simbolo a, $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$.

Una parola w è accettata da $\mathcal{A}$ se (e solo se) $\hat{\delta}(q_0, w) \in F$. Il linguaggio $L(\mathcal{A})$ accettato da $\mathcal{A}$ è l'insieme di tutte e sole le parole su A accettate da $\mathcal{A}$. Un linguaggio è detto regolare *se è un insieme accettato da un automa finito.* □

Si noti che $\hat{\delta}(q, a) = \delta(q, a)$. D'ora in poi, per semplicità, ogni qualvolta ciò non comporti delle ambiguità, useremo δ al posto di $\hat{\delta}$.

Definizione 1.2 (*Automa finito nondeterministico*)

Un automa finito nondeterministico (NFA) $\mathcal{A}$ è una quintupla (Q, A, δ, q_0, F), dove Q, A, q_0 e F sono definiti come nel caso degli automi finiti deterministici, mentre $\delta : \ Q \times A \to 2^Q$ è una funzione di transizione che riceve in ingresso uno stato e un simbolo dell'alfabeto e restituisce in uscita un insieme di stati.

La funzione di transizione $\hat{\delta}$, che estende δ sostituendo A con A^*, è definita nel seguente modo: $\hat{\delta}(q, \epsilon) = \{q\}$ e, per ogni parola w e simbolo a, $\hat{\delta}(q, wa) = \bigcup_{p \in \hat{\delta}(q,w)} \delta(p, a)$.

Una parola w è accettata da $\mathcal{A}$ se (e solo se) $\hat{\delta}(q_0, w) \cap F \neq \emptyset$. Il linguaggio $L(\mathcal{A})$ accettato da $\mathcal{A}$ è l'insieme di tutte e sole le parole su A accettate da $\mathcal{A}$. □

La funzione di transizione δ può essere sostituita da un'equivalente relazione di transizione Δ. Nel seguito faremo riferimento ora all'una ora all'altra notazione.

Si noti che $\hat{\delta}(q,a) = \delta(q,a)$. Come fatto in precedenza, d'ora in poi, quando possibile, useremo δ al posto di $\hat{\delta}$. E' inoltre utile estendere δ ad argomenti in $2^Q \times A^*$ nel modo seguente: $\delta(P,w) = \bigcup_{p\in P} \delta(p,w)$.

Teorema 1.3 (*Equivalenza tra automi finiti deterministici e nondeterministici*)

Per ogni automa finito nondeterministico, esiste un automa finito deterministico che accetta lo stesso linguaggio, e viceversa.

Prova

Sia $\mathcal{A} = (Q, A, \delta, q_0, F)$ un NFA che accetta il linguaggio L. Il corrispondente DFA $\mathcal{A}' = (Q', A, \delta', q_0', F')$ può essere costruito nel seguente modo:

- $Q' = 2^Q$;
- $q_0' = \{q_0\}$;
- per ogni $P \in Q' (= 2^Q)$ e $a \in A$, $\delta'(P,a) = \delta(P,a)$;
- $F' = \{P \in Q' : \ P \cap F \neq \emptyset\}$.

Si noti che P in $\delta'(P,a)$ rappresenta uno stato (elemento di Q'), mentre in $\delta(P,a)$ rappresenta un insieme di stati (sottoinsieme di Q).

Dimostriamo, per induzione sulla lunghezza della parola w, che $\delta'(q_0', w) = \delta(q_0, w)$, ossia che lo stato restituito da $\delta'(q_0', w)$ "coincide" con l'insieme di stati restituito da $\delta(q_0, w)$. Se $|w| = 0$, ossia se $x = \epsilon$, allora $\delta'(q_0', \epsilon) = \{q_0\} = \delta(q_0, \epsilon)$. Supponiamo che la tesi valga per parole di lunghezza minore o uguale a n e consideriamo una parola wa di lunghezza $n+1$, con $a \in A$. Per definizione, $\delta'(q_0', wa) = \delta'(\delta'(q_0', w), a)$. Per ipotesi induttiva, $\delta'(q_0', w) = \delta(q_0, w)$. Dunque, $\delta'(q_0', wa) = \delta'(\delta(q_0, w), a) = \bigcup_{p\in\delta(q_0,w)} \delta'(\{p\}, a) = \bigcup_{p\in\delta(q_0,w)} \delta(p,a) = \delta(q_0, wa)$.

Per completare la prova basta osservare che $\mathcal{A}'$ accetta w se e solo se $\delta'(q_0', w) \in F'$ se e solo se $\delta(q_0, w) \in F'$ se e solo se $\delta(q_0, w) \cap F \neq \emptyset$ se e solo se $\mathcal{A}$ accetta w. Quindi $L(\mathcal{A}) = L(\mathcal{A}')$.

Il viceversa è immediato. ■

Si noti che la dimensione (numero di stati) dell'automa deterministico ottenuto nella precedente dimostrazione è *esponenziale* nella dimensione dell'automa nondeterministico di partenza.

Definizione 1.4 (*Automa finito nondeterministico con ϵ-mosse*)

Un automa finito nondeterministico con ϵ-mosse $\mathcal{A}$ è una quintupla (Q, A, δ, q_0, F), dove Q, A, q_0 e F sono definiti come nel caso degli automi finiti nondeterministici, mentre $\delta : \ Q \times (A \cup \{\epsilon\}) \to 2^Q$ è una funzione di transizione che riceve in ingresso uno stato e un simbolo dell'alfabeto, oppure la parola vuota ϵ, e restituisce in uscita un insieme di stati.

Siano dati $q \in Q$ e $P \in 2^Q$. Una ϵ-mossa è una transizione di tipo $\delta(q,\epsilon)$. L'insieme di stati raggiungibili a partire da q usando solo ϵ-mosse è denotato da ϵ-$CLOSURE(q)$. Definiamo inoltre, per ogni $P \in 2^Q$, ϵ-$CLOSURE(P) = \bigcup_{p\in P} \epsilon$-$CLOSURE(p)$. La funzione di transizione $\hat{\delta}$ è definita nel seguente modo: $\hat{\delta}(q,\epsilon) = \epsilon$-$CLOSURE(q)$ e, per ogni parola w e simbolo a, $\hat{\delta}(q, wa) = \epsilon$-$CLOSURE(\bigcup_{p\in\hat{\delta}(q,w)} \delta(p,a))$.

Una parola w è accettata da $\mathcal{A}$ se (e solo se) $\hat{\delta}(q_0, w) \cap F \neq \emptyset$. Il linguaggio $L(\mathcal{A})$ accettato da $\mathcal{A}$ è l'insieme di tutte e sole le parole su A accettate da $\mathcal{A}$. □

Si noti che, per ogni $a \in A$,

$$\hat{\delta}(q,a) = \epsilon\text{-}CLOSURE(\bigcup_{p \in \hat{\delta}(q,\epsilon)} \delta(p,a)) = \epsilon\text{-}CLOSURE(\bigcup_{p \in \epsilon\text{-}CLOSURE(q)} \delta(p,a)).$$

Tale valore, in generale, non coincide con $\delta(q,a)$. Similmente, $\hat{\delta}(q,\epsilon)$ può essere diverso da $\delta(q,\epsilon)$. Nel seguito, occorrerà perciò distinguere tra le funzioni δ e $\hat{\delta}$.

Teorema 1.5 (*Equivalenza tra automi finiti nondeterministici con o senza ϵ-mosse*)

Per ogni automa finito nondeterministico con ϵ-mosse, esiste un automa finito nondeterministico senza ϵ-mosse che accetta lo stesso linguaggio, e viceversa.

Prova

Sia $\mathcal{A} = (Q, A, \delta, q_0, F)$ un NFA con ϵ-mosse che accetta un linguaggio L. Definiamo un NFA $\mathcal{A}' = (Q, A, \delta', q_0, F')$ tale che, $\delta'(q,a) = \hat{\delta}(q,a)$ e $F' = F \cup \{q_0\}$, se $\epsilon\text{-}CLOSURE(q_0) \cap F \neq \emptyset$, altrimenti $F' = F$. E' facile mostrare che $L(\mathcal{A}) = L(\mathcal{A}')$.

Il viceversa è immediato. ■

1.2 Espressioni regolari

Esistono vari tipi di espressioni regolari su un un alfabeto A:

a) *Espressioni regolari ristrette*: sono costruite a partire da insiemi finiti di parole su A usando le operazioni $\cup$ (unione), $\cdot$ (concatenazione) e $*$ (chiusura di Kleene).

b) *Espressioni regolari generali*: sono costruite a partire da insiemi finiti di parole su A usando le operazioni $\cup$ (unione), $\cap$ (intersezione), $\neg$ (complementazione rispetto a A^*), $\cdot$ (concatenazione) e $*$ (chiusura di Kleene).

c) *Espressioni regolari star-free*: sono le espressioni regolari generali che non contengono occorrenze di $*$.

In seguito mostreremo che le espressioni regolari ristrette e generali hanno lo stesso potere espressivo e che le espressioni regolari star-free sono strettamente meno espressive delle espressioni regolari generali (o ristrette). Con il termine espressioni regolari faremo riferimento convenzionalmente ad espressioni regolari ristrette.

Teorema 1.6 (*Equivalenza tra automi finiti ed espressioni regolari*)

Per ogni automa finito, esiste una espressione regolare (ristretta) che definisce lo stesso linguaggio, e viceversa.

Prova

Proviamo innanzitutto che, dato un DFA, esiste una espressione regolare che genera lo stesso linguaggio. Sia $\mathcal{A} = (\{q_1, \ldots q_n\}, A, \delta, q_1, F)$ un DFA. Sia $R^k_{i,j}$ l'insieme delle parole x tali che $\delta(q_i, x) = q_j$ e, per ogni prefisso proprio $y \neq \epsilon$ di x, se $\delta(q_i, y) = q_l$, allora $l \leq k$. In altri termini, $R^k_{i,j}$ contiene le parole che portano da q_i a q_j senza passare (entrare ed uscire) per stati di indice maggiore di k. L'insieme $R^k_{i,j}$ è definibile ricorsivamente nel seguente modo:

$$R^0_{i,j} = \begin{cases} \{a : \ \delta(q_i,a) = q_j\} & i \neq j \\ \{a : \ \delta(q_i,a) = q_j\} \cup \{\epsilon\} & i = j \end{cases}$$

$$R^k_{i,j} = (R^{k-1}_{i,k} \cdot (R^{k-1}_{k,k})^* \cdot R^{k-1}_{k,j}) \cup R^{k-1}_{i,j}$$

Mostriamo che, per ogni k, i e j, esiste una espressione regolare per l'insieme $R^k_{i,j}$. Procediamo per induzione su k. Se $k = 0$, $R^k_{i,j}$ è un insieme finito di parole su A, caratterizzabile facilmente per mezzo di un'espressione regolare. Assumiamo ora $k > 0$. Per ipotesi induttiva, per ogni i, j, esiste una espressione regolare che cattura l'insieme $R^{k-1}_{i,j}$. Dalla definizione ricorsiva di $R^k_{i,j}$, è facile ricavare un'espressione regolare per $R^k_{i,j}$, componendo opportunemente le espressioni regolari per gli insiemi $R^{k-1}_{i,j}$. Dato che

$$L(\mathcal{A}) = \bigcup_{q_j \in F} R^n_{1,j},$$

è immediato ricavare un'espressione regolare per $L(\mathcal{A})$.

Proviamo ora che per ogni espressione regolare, esiste un NFA con ϵ-mosse con (al più) uno stato finale dal quale non esce alcuna transizione che riconosce lo stesso linguaggio. Siano A un alfabeto e r un'espressione regolare su A. Procediamo per induzione sulla struttura di r. Se $r = \epsilon$, $r = \emptyset$ o $r = a$, con $a \in A$, è immediato costruire un automa finito equivalente (un automa per $r = \epsilon$ è un automa con un solo stato, privo di transizioni, che è sia stato inziale sia stato finale dell'automa; un automa per $r = \emptyset$ è un automa con un solo stato, che funge da stato iniziale, privo di transizioni e di stati finali; un automa per $r = a$ è un automa con due stati e un'unica transizione, etichettata con a, che porta dallo stato iniziale allo stato finale).

Se $r = r_1 \cup r_2$, per ipotesi induttiva esistono due NFA con ϵ-mosse $\mathcal{A}_1 = (Q_1, A_1, \delta_1, q_1, \{f_1\})$ e $\mathcal{A}_2 = (Q_2, A_2, \delta_2, q_2, \{f_2\})$ privi di transizioni che escono dai rispettivi stati finali che riconoscono, rispettivamente, i linguaggi r_1 e r_2. Senza perdita di generalità possiamo assumere che Q_1 e Q_2 siano disgiunti.

E' facile provare che l'automa $\mathcal{A} = (Q_1 \cup Q_2 \cup \{q_0, f_0\}, A_1 \cup A_2, \delta, q_0, \{f_0\})$, dove δ è definita come segue:

- $\delta(q_0, \epsilon) = \{q_1, q_2\}$,
- $\delta(q, a) = \delta_1(q, a)$ per $q \in Q_1 \setminus \{f_1\}$, $a \in A_1 \cup \{\epsilon\}$,
- $\delta(q, a) = \delta_2(q, a)$ per $q \in Q_2 \setminus \{f_2\}$, $a \in A_2 \cup \{\epsilon\}$,
- $\delta(f_1, \epsilon) = \delta(f_2, \epsilon) = \{f_0\}$,

riconosce il linguaggio $L(\mathcal{A}) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$.

Se $r = r_1 \cdot r_2$, per ipotesi induttiva esistono due NFA con ϵ-mosse $\mathcal{A}_1$ e $\mathcal{A}_2$, definiti come nel caso precedente, che riconoscono, rispettivamente, i linguaggi r_1 e r_2. Non è difficile provare che l'automa $\mathcal{A} = (Q_1 \cup Q_2, A_1 \cup A_2, \delta, q_1, \{f_2\})$, dove δ è definita come segue:

- $\delta(q, a) = \delta_1(q, a)$ per $q \in Q_1 \setminus \{f_1\}$, $a \in A_1 \cup \{\epsilon\}$,
- $\delta(f_1, \epsilon) = \{q_2\}$,
- $\delta(q, a) = \delta_2(q, a)$ per $q \in Q_2$, $a \in A_2 \cup \{\epsilon\}$,

riconosce il linguaggio $L(\mathcal{A}) = L(\mathcal{A}_1) \cdot L(\mathcal{A}_2)$.

Infine, se $r = r_1^*$, per ipotesi induttiva esiste un NFA con ϵ-mosse $\mathcal{A}_1 = (Q_1, A_1, \delta_1, q_1, \{f_1\})$, privo di transizioni uscenti da f_1, che riconosce il linguaggio r_1. Sia $\mathcal{A} = (Q_1 \cup \{q_0, f_0\}, A_1, \delta, q_0, \{f_0\})$, dove δ è definita come segue:

- $\delta(q_0, \epsilon) = \delta(f_1, \epsilon) = \{q_1, f_0\}$,
- $\delta(q, a) = \delta_1(q, a)$ per $q \in Q_1 \setminus \{f_1\}$, $a \in A_1 \cup \{\epsilon\}$,

E' facile provare che $L(\mathcal{A}) = (L(\mathcal{A}_1))^*$. ■

1.3 Proprietà dei linguaggi regolari

Teorema 1.7 (*Proprietà di chiusura*)

I linguaggi regolari sono chiusi rispetto alle operazioni booleane, alla concatenazione e alla chiusura di Kleene. Inoltre, la costruzione degli automi che riconoscono i linguaggi che si ottengono attraverso l'esecuzione di tali operazioni è effettiva.

Prova

Per le operazioni di unione, concatenazione e chiusura di Kleene, la tesi segue immediatamente dalla dimostrazione del Teorema 1.6.

Consideriamo l'operazione di complementazione. Sia dato un DFA $\mathcal{A} = (Q, A, \delta, q_0, F)$. Assumiamo che per ogni stato $q \in Q$ ed ogni simbolo $a \in A$ esista una transizione uscente da q etichettata con a (definiamo *completo* un automa che soddisfa tale condizione).[1] A partire da $\mathcal{A}$, definiamo un DFA $\mathcal{A}' = (Q, A, \delta, q_0, Q \setminus F)$. Per costruzione, $\mathcal{A}'$ accetta una parola $w \in A^*$ se e solo se $\mathcal{A}$ non accetta w. Da cui segue che $L(\mathcal{A}') = A^* \setminus L(\mathcal{A})$.

La proprietà di chiusura rispetto all'operazione di intersezione segue immediatamente dalle proprietà di chiusura rispetto alle operazioni di unione e complementazione (l'intersezione di due linguaggi può essere definita come il complemento dell'unione dei complementi dei due linguaggi dati). L'automa che riconosce il linguaggio intersezione può essere costruito nel modo seguente. Siano $\mathcal{A}_1 = (Q_1, A, \delta_1, q_1, F_1)$ e $\mathcal{A}_2 = (Q_2, A, \delta_2, q_2, F_2)$ due DFA. Definiamo $\mathcal{A} = (Q_1 \times Q_2, A, \delta, (q_1, q_2), F_1 \times F_2)$, ove $\delta((p_1, p_2), a) = (\delta_1(p_1, a), \delta_2(p_2, a))$. E' facile vedere che $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$. ■

E' importante osservare come nella dimostrazione della chiusura rispetto alla complementazione determinismo e completezza dell'automa $\mathcal{A}$ giochino un ruolo fondamentale nella costruzione dell'automa "complementare" $\mathcal{A}'$. In particolare, se $\mathcal{A}$ fosse non deterministico, vi potrebbero essere delle parole x riconosciute sia da $\mathcal{A}$ sia da $\mathcal{A}'$ (dato un automa $\mathcal{A}$ non deterministico, tali sono tutte le parole per le quali esistono sia una computazione di $\mathcal{A}$ che termina in uno stato di F sia una computazione di $\mathcal{A}$ che termina in uno stato di $Q \setminus F$). Nel caso in cui $\mathcal{A}$ non fosse completo, vi potrebbero essere delle parole che non appartengono né ad $\mathcal{A}$ né ad $\mathcal{A}'$ (dato un automa $\mathcal{A}$ incompleto, tali sono tutte le parole per le quali non esistono computazioni in grado di scandirle nella loro interezza).

Un importante corollario del Teorema 1.7 è il seguente.

Corollario 1.8 *Le espressioni regolari ristrette sono tanto espressive quanto le espressioni regolari generali.*

Prova

Dato il DFA che riconosce $A^* \setminus L(\mathcal{A})$ o quello che riconosce $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$, applicando le tecniche usate per la dimostrazione del Teorema 1.7, è possibile ottenere un'espressione regolare ristretta che definisce il medesimo linguaggio.

D'ora in avanti, dato un linguaggio L, useremo la scrittura $\overline{L}$ per denotare il complementare di L.

Teorema 1.9 (*Decidibilità e algoritmi di decisione*)

Dati due automi finiti $\mathcal{A}$ e $\mathcal{A}'$, i problemi di stabilire se $L(\mathcal{A}) = \emptyset$ (problema del vuoto), $L(\mathcal{A}) = A^*$ (problema dell'universalità), $L(\mathcal{A}) \subseteq L(\mathcal{A}')$ (problema dell'inclusione) e $L(\mathcal{A}) = L(\mathcal{A}')$ (problema dell'equivalenza) sono decidibili.

[1] Si noti che ogni automa incompleto si può facilmente completare, ossia per ogni automa incompleto, esiste un automa completo ad esso equivalente. Tale automa si ottiene nel seguente modo: si aggiunge un nuovo stato $\overline{q}$, non finale, e, per ogni simbolo $a \in A$, si aggiunge una transizione da $\overline{q}$ a $\overline{q}$ etichettata con a; inoltre, per ogni stato $q \in Q$ e ogni simbolo $a \in A$ per i quali non esista una transizione uscente da q etichettata con a, si aggiunge una transizione da q a $\overline{q}$ etichettata con a.

Prova

Sia $\mathcal{A} = (\{q_1, \ldots q_n\}, A, \delta, q_1, F)$. Mostriamo che $L(\mathcal{A}) \neq \emptyset$ se e solo se esiste $w \in L(\mathcal{A})$ tale che $|w| < n$, da cui segue immediatamente la decidibilità del problema di stabilire se $L(\mathcal{A}) = \emptyset$. L'implicazione da destra a sinistra è ovvia. Dimostriamo per assurdo l'implicazione da sinistra a destra. Supponiamo che $L(\mathcal{A}) \neq \emptyset$ e per ogni $w \in L(\mathcal{A})$ valga $|w| \geq n$. Sia $z \in L(\mathcal{A})$ tale che $|z| \leq |w|$, per ogni $w \in L(\mathcal{A})$. Scriviamo $z = a_1 \ldots a_m$, dove $a_i \in A$, per $i = 1, \ldots, m$ e $m \geq n$. Sia $p_i = \delta(q_1, a_1 \ldots a_i)$, per $i = 0, 1, \ldots, m$. Dato che $m \geq n$ e gli stati di $\mathcal{A}$ sono n, almeno uno dei p_i è ripetuto. Ne segue che esistono i, j, con $i < j$, tali che $p_i = p_j$. Ciò significa che z si può scrivere come $z = uvw$, dove $u = a_1 \ldots a_i$, $v = a_{i+1} \ldots a_j$, $w = a_{j+1} \ldots a_m$. Dato che $p_i = p_j$ e $z = uvw \in L(\mathcal{A})$, allora anche $uw \in L(\mathcal{A})$. Ma $|uw| < |z|$, perchè $|v| > 0$. Assurdo, in quanto z è la parola più corta in $L(\mathcal{A})$.

I problemi dell'universalità, dell'inclusione e dell'equivalenza sono decidibili in quanto si ha che $L(\mathcal{A}) = A^*$ se e solo se $\overline{L(\mathcal{A})} = \emptyset$; $L(\mathcal{A}) \subseteq L(\mathcal{A}')$ se e solo se $L(\mathcal{A}) \cap \overline{L(\mathcal{A}')} = \emptyset$, e $L(\mathcal{A}) = L(\mathcal{A}')$ se e solo se $(L(\mathcal{A}') \cap \overline{L(\mathcal{A})}) \cup (L(\mathcal{A}) \cap \overline{L(\mathcal{A}')}) = \emptyset$. Si noti che, in virtù del Teorema 1.7, le costruzioni degli automi sono effettive. ■

Un algoritmo polinomiale (nel numero di stati dell'automa) che verifica se $L(\mathcal{A}) = \emptyset$ è il seguente: elimino tutti gli stati di $\mathcal{A}$ che non sono raggiungibili dallo stato iniziale. L'automa $\mathcal{A}$ riconosce qualche parola se è rimasto almeno uno stato finale.

Definizione 1.10 (*Relazione di equivalenza invariante destra*)

Sia A un generico insieme chiuso rispetto all'operazione di concatenazione. Una relazione di equivalenza $\sim$ su A si dice invariante destra (rispetto all'operazione di concatenazione) se, per ogni $x, y \in A$, $x \sim y$ implica che, per ogni $z \in A$, $xz \sim yz$. □

Definizione 1.11 (*Relazione $\sim_L$*)

Dato un linguaggio $L \subseteq A^*$, definiamo la relazione $\sim_L$ su A^* tale che, per ogni $x, y \in A^*$, $x \sim_L y$ se, per ogni $z \in A^*$, $xz \in L$ se e solo se $yz \in L$. □

E' immediato verificare che $\sim_L$ è una relazione di equivalenza. Inoltre $\sim_L$ è invariante destra. Siano dati $x \sim_L y$ e $v \in A^*$. Occorre provare che $xv \sim_L yv$, ossia che, per ogni $z \in A^*$, $xvz \in L$ se e solo se $yvz \in L$. Ciò segue dal fatto che $x \sim_L y$ (e $vz \in A^*$).

Definizione 1.12 (*Relazione $\sim_{\mathcal{A}}$*)

Dato un DFA $\mathcal{A} = (Q, A, \delta, q_0, F)$, definiamo la relazione $\sim_{\mathcal{A}}$ su A^* tale che, per ogni $x, y \in A^*$, $x \sim_{\mathcal{A}} y$ se $\delta(q_0, x) = \delta(q_0, y)$. □

Come in precedenza, è immediato verificare che la relazione $\sim_{\mathcal{A}}$ è una relazione di equivalenza. Dimostriamo che è una relazione di equivalenza invariante destra. Siano dati $x \sim_{\mathcal{A}} y$ e $v \in A^*$. Occorre provare che $xv \sim_{\mathcal{A}} yv$, ossia che $\delta(q_0, xv) = \delta(q_0, yv)$. Dalla definizione di δ segue facilmente che $\delta(q_0, xv) = \delta(\delta(q_0, x), v)$ e $\delta(q_0, yv) = \delta(\delta(q_0, y), v)$. Da $x \sim_{\mathcal{A}} y$ segue che $\delta(q_0, x) = \delta(q_0, y)$ e quindi la tesi. E' possibile inoltre dimostrare che il numero di classi determinate da $\sim_{\mathcal{A}}$ (*indice* di $\sim_{\mathcal{A}}$) è finito. A tal fine, è sufficiente osservare che ogni classe di equivalenza di $\sim_{\mathcal{A}}$ corrisponde ad uno stato di $\mathcal{A}$ raggiungibile a partire da q_0; poiché il numero di stati di $\mathcal{A}$ è finito, $\sim_{\mathcal{A}}$ ha indice finito.

Teorema 1.13 (*Myhill-Nerode*)

Sia $L \subseteq A^*$ un linguaggio di parole finite. Le seguenti proposizioni sono equivalenti:

(1) L è regolare;

(2) L è esprimibile come unione di classi di equivalenza di una relazione di equivalenza invariante destra di indice finito;

(3) la relazione di equivalenza $\sim_L$ ha indice finito.

Prova

(1) $\rightarrow$ (2). Supponiamo che L sia accettato dal DFA $\mathcal{A} = (Q, A, \delta, q_0, F)$. Consideriamo la corrispondente relazione $\sim_{\mathcal{A}}$ (cf. Definizione 1.12). Essa è una relazione di equivalenza invariante destra di indice finito. Inoltre, L è l'unione delle classi di equivalenza di $\sim_{\mathcal{A}}$ che contengono una parola x tale che $\delta(q_0, x) \in F$.

(2) $\rightarrow$ (3). Mostriamo che ogni relazione $\sim$ che soddisfa la (2) è un raffinamento di $\sim_L$ ($\sim$ è un raffinamento di $\sim_L$ se, per ogni x, y, $x \sim y$ implica $x \sim_L y$, ossia che ogni classe di equivalenza di $\sim$ è interamente contenuta in una classe di equivalenza di $\sim_L$). Da ciò segue che l'indice di $\sim_L$ non è maggiore dell'indice di $\sim$ e quindi è finito. Sia $x \sim y$. Poichè $\sim$ è invariante destra, per ogni $z \in A^*$, $xz \sim yz$. Dato che L è l'unione di classi di equivalenza di $\sim$, abbiamo che $xz \in L$ se e solo se $yz \in L$. Dunque $x \sim_L y$.

(3) $\rightarrow$ (1). Sia $\mathcal{A}' = (Q', A, \delta', q_0', F')$, dove $Q' = A^*/\sim_L$, $\delta'([x]_{\sim_L}, a) = [xa]_{\sim_L}$, $q_0' = [\epsilon]_{\sim_L}$ e $F' = \{[x]_{\sim_L}.\ x \in L\}$. La definizione di δ' è consistente: se $y \in [x]_{\sim_L}$, allora $[xa]_{\sim_L} = [ya]_{\sim_L}$ in quanto $\sim_L$ è invariante destra. Infine, $\mathcal{A}'$ accetta x se e solo se $\delta'(q_0', x) = [x]_{\sim_L} \in F'$ se e solo se $x \in L$. Dunque $\mathcal{A}'$ accetta L. ■

Dalla prova del Teorema 1.13 segue che se $L \subseteq A^*$ è regolare, esiste una corrispondenza tra automi finiti che riconoscono L e relazioni di equivalenza di indice finito invarianti destre tali che L è l'unione di alcune loro classi di equivalenza (proposizione (2)). Dato un automa $\mathcal{A}$ per L, è infatti possibile definire la relazione $\sim_{\mathcal{A}}$ che soddisfa (2) (si veda la prova dell'implicazione (1) $\rightarrow$ (2)). Tale relazione ha indice finito minore o uguale al numero di stati dell'automa $\mathcal{A}$ (alcuni stati potrebbero non essere accessibili da q_0). Viceversa, data una relazione $\sim$ che soddisfa (2), è possibile definire un automa che riconosce L, seguendo il procedimento utilizzato nella prova dell'implicazione (3) $\rightarrow$ (1). Il numero di stati dell'automa ottenuto è pari all'indice della relazione di equivalenza. Inoltre, dalla prova dell'implicazione (2) $\rightarrow$ (3) segue che $\sim_L$ è la relazione di equivalenza più grossolana, ossia con il minimo indice, che soddisfa la proposizione (2) (si noti che $L = \bigcup\{[x]_{\sim_L} : x \in L\}$). Dunque, $\sim_L$ corrisponde all'automa minimo, ossia con il minimo numero di stati, che riconosce L. E' possibile dimostrare che l'automa minimo è unico a meno di isomorfismi.

2 Automi a stati finiti su parole infinite

2.1 Automi di Büchi e linguaggi ω-regolari

Come in precedenza, sia A un alfabeto finito di simboli. Una parola infinita, o ω-parola, su A è una sequenza infinita di simboli di A giustapposti. Denotiamo con A^ω l'insieme delle ω-parole su A. Un linguaggio (di parole infinite) su A è un sottoinsieme di A^ω. Definiamo, inoltre, l'insieme $A^\infty = A^* \cup A^\omega$. Una ω-parola verrà scritta nella forma $\alpha = \alpha(0)\alpha(1), \ldots$, con $\alpha(i) \in A$ per ogni $i \geq 0$. Se $n \leq m$, $\alpha(n, m) = \alpha(n) \ldots \alpha(m-1)$ e $\alpha(n, \omega) = \alpha(n)\alpha(n+1) \ldots$. Useremo le abbreviazioni $\exists^\omega n$ per "esistono infiniti n" e $\exists^{<\omega} n$ per "esistono finiti n". Dato un insieme $W \subseteq A^*$, definiamo gli insiemi $W^\omega = \{\alpha \in A^\omega.\ \alpha = w_0 w_1 \ldots, con\ w_i \in W\}$ e $\overrightarrow{W} = \{\alpha \in A^\omega.\ \exists^\omega n\ \alpha(0, n) \in W\}$. Infine, per ogni $\alpha \in A^\omega$, definiamo $In(\alpha) = \{a \in A.\ \exists^\omega n\ \alpha(n) = a\}$.

Definizione 2.1 (*Automa di Büchi*)

Un automa di Büchi *è una quintupla $\mathcal{A} = (Q, A, \Delta, q_0, F)$, dove Q è un insieme finito di stati, A è un alfabeto finito di simboli, $q_0 \in Q$ è lo stato iniziale, $F \subseteq Q$ è l'insieme degli stati finali e $\Delta \subseteq Q \times A \times Q$ è la relazione di transizione.*

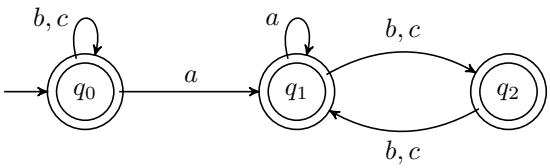

Figure 2.1: L'automa dell'Esempio 2.2.

Una computazione *di $\mathcal{A}$ su una ω-parola α è una ω-parola σ su Q tale che $\sigma(0) = q_0$ e, per ogni $i \geq 0$, $(\sigma(i), \alpha(i), \sigma(i+1)) \in \Delta$.*

Una computazione σ ha successo *se $In(\sigma) \cap F \neq \emptyset$. L'automa $\mathcal{A}$ accetta una ω-parola α se esiste una computazione di successo di $\mathcal{A}$ su α. Il linguaggio $L(\mathcal{A})$ accettato da $\mathcal{A}$ è l'insieme delle ω-parole accettate da $\mathcal{A}$. Un linguaggio è detto* ω-regolare *se (e solo se) è accettato da una automa di Büchi.* □

Nel caso in cui non sia sorgente di ambiguità, useremo l'espressione linguaggi regolari anche per i linguaggi ω-regolari.

Esempio 2.2 (*Linguaggio ω-regolare*)

Consideriamo il linguaggio L che contiene tutte e sole le ω-parole su $A = \{a, b, c\}$ tali che tra ogni coppia di occorrenze "consecutive" di a (ossia, tali che tra di esse non vi è alcuna altra occorrenza di a) esiste un numero pari di occorrenze di simboli diversi da a (b e/o c). Il linguaggio L è ω-regolare in quanto è riconosciuto dall'automa di Büchi $\mathcal{A} = (\{q_0, q_1, q_2\}, A, q_0, \Delta, \{q_0, q_1, q_2\})$, la cui relazione di transizione Δ è descritta in Figura 2.1. Si osservi come tutti e tre gli stati dell'automa siano stati finali. In particolare, si è scelto di considerare finale anche lo stato più a destra in Figura 2.1. Si osservi anche come non sia possibile accertare in un numero finito di passi se una data ω-parola $\alpha \in L$, mentre è sufficiente un numero finito di passi per stabilire se una data ω-parola $\alpha \notin L$.

Esercizio 2.3 *Sia $\mathcal{A}$ l'automa dell'Esempio 2.2. Si consideri l'automa $\mathcal{A}'$ ottenuto da $\mathcal{A}$ rimuovendo lo stato q_0, e le transizioni in esso entranti e da esso uscenti, e facendo diventare q_1 il nuovo stato iniziale. Si stabilisca se $\mathcal{A}$ e $\mathcal{A}'$ riconoscono o meno lo stesso linguaggio.*

Esercizio 2.4 *Si costruisca l'automa $\mathcal{A}'$ che riconosce la variante finita (linguaggio di parole finite) dell'Esempio 2.2.*

Esercizio 2.5 *Sia W il linguaggio riconosciuto dall'automa $\mathcal{A}'$ dell'Esercizio 2.4. Si caratterizzi il linguaggio $\overrightarrow{W}$.*

Teorema 2.6 (*Proprietà di chiusura*)

1. *Se $V \subseteq A^*$ è regolare, allora V^ω è ω-regolare;*
2. *se $V \subseteq A^*$ è regolare e $L \subseteq A^\omega$ è ω-regolare, allora $V \cdot L$ è ω-regolare;*
3. *se $L_1, L_2 \subseteq A^\omega$ sono ω-regolari, allora $L_1 \cup L_2$ e $L_1 \cap L_2$ sono ω-regolari.*

Inoltre, la costruzione degli automi è effettiva.

Prova

(1) Sia $\mathcal{A} = (Q, A, \Delta, q_0, F)$ l'automa che riconosce V. Dal momento che $V^\omega = (V \setminus \{\epsilon\})^\omega$, assumiamo, senza perdita di generalità, che V non contenga la parola vuota ϵ. Assumiamo, inoltre, che non vi siano transizioni entranti nello stato iniziale q_0. Un automa di Büchi che riconosce V^ω si ottiene aggiungendo all'automa $\mathcal{A}$ una transizione (s, a, q_0) per ogni transizione $(s, a, s') \in \Delta$, con s' stato finale, e assumendo q_0 quale unico stato finale dell'automa così ottenuto.

(2, 3) Le prove fornite nel caso degli automi su parole finite (cf. Teorema 1.7) non possono essere trasferite in modo immediato. La prova di tali proprietà è lasciata come esercizio. ■

Esercizio 2.7 *Dimostrare le proprietà (2) e (3) del Teorema 2.6.*

Definizione 2.8 (*Espressioni ω-regolari*)

Un'espressione ω-regolare ha la forma $\bigcup_{i=1}^{n} U_i \cdot V_i^\omega$, dove, per ogni $1 \leq i \leq n$, U_i e V_i sono espressioni regolari. □

Siano dati un automa di Büchi $\mathcal{A} = (Q, A, \Delta, q_0, F)$, $w \in A^*$ e $s, s' \in Q$. Con la notazione $s \rightarrow_w s'$ modelliamo l'esistenza di una computazione di $\mathcal{A}$ su w che conduce dallo stato s allo s'. Per ogni coppia di stati $s, s' \in Q$, definiamo $W_{ss'} = \{w \in A^*.\ s \rightarrow_w s'\}$. È immediato vedere che, per ogni $s, s' \in Q$, $W_{ss'}$ è un linguaggio regolare: un automa che riconosce $W_{ss'}$ si ottiene da $\mathcal{A}$ ponendo s come stato iniziale e $\{s'\}$ come insieme di stati finali.

Teorema 2.9 (*Equivalenza tra automi di Büchi ed espressioni ω-regolari - Büchi 1960*)

Per ogni automa di Büchi esiste una espressione ω-regolare che definisce lo stesso linguaggio e viceversa.

Prova

Sia $\mathcal{A} = (Q, A, \Delta, q_0, F)$ un automa di Büchi. Mostriamo che esiste una espressione ω-regolare che definisce $L(\mathcal{A})$. Dalla definizione della condizione di accettazione per un automa di Büchi, segue che $L(\mathcal{A}) = \bigcup_{s \in F} W_{q_0,s} \cdot (W_{ss})^\omega$. La tesi segue immediatamente dalla regolarità degli insiemi $W_{q_0,s}$ e W_{ss}, per ogni $s \in F$ (ogni insieme dell'insieme finito di insiemi della forma $W_{ss'}$, con $s, s' \in Q$, è regolare). Il viceversa si ricava direttamente dal Teorema 2.6. ■

Definizione 2.10 (*ω-parole definitivamente periodiche*)

Una ω-parola $\alpha \in A^\omega$ si dice definitivamente periodica *se $\alpha = uv^\omega$, per qualche $u, v \in A^*$.* □

Corollario 2.11 *Ogni linguaggio ω-regolare non vuoto L contiene una ω-parola definitivamente periodica (ossia una ω-parola del tipo $uvvv\ldots$).*

Prova

Dato un linguaggio ω-regolare L, dal Teorema 2.9 segue che $L = \bigcup_{i=1}^{n} U_i \cdot V_i^\omega$, con $U_i, V_i \subseteq A^*$ regolari. Da $L \neq \emptyset$ segue che $\exists\ uv_1v_2\ldots \in L$, con $u \in U_i$ e, per tutti $j \geq 1$, $v_j \in V_i$, per qualche $1 \leq i \leq n$. Da $uv_1v_2v_3\ldots \in L$ segue che $uv_1v_1v_1\ldots \in L$. ■

Esempio 2.12 (*Linguaggio (infinito) non ω-regolare*)

Sia $\beta \in A^\omega$ una parola non definitivamente periodica. Sia $L(\beta)$ il linguaggio delle ω-parole su A che hanno un suffisso in comune con β. Mostriamo che $L(\beta)$ non è ω-regolare. Per assurdo, assumiamo che $L(\beta)$ sia ω-regolare. E' immediato osservare che $L(\beta) \neq \emptyset$ ($\beta \in L(\beta)$). Per il Corollario 2.11, $L(\beta)$ deve contenere almeno una parola definitivamente periodica, ma, per costruzione, $L(\beta)$ non contiene alcuna parola definitivamente periodica (contraddizione). Dunque $L(\beta)$ non è ω-regolare.

Esercizio 2.13 *Fornire un esempio di parola non definitivamente periodica.*

Teorema 2.14 (*Decidibilità del problema del vuoto*)

Il problema del vuoto per gli automi di Büchi è decidibile.

Prova

Dalla definizione di condizione di accettazione per un automa di Büchi segue che un automa di Büchi $\mathcal{A}$ accetta almeno una parola se e solo se il suo grafo di transizione contiene un ciclo che contiene uno stato finale raggiungibile a partire dallo stato iniziale. ■

2.2 Complementazione

In questa sezione proveremo la chiusura dei linguaggi ω-regolari rispetto all'operazione di complementazione. Inoltre, mostreremo come un linguaggio ω-regolare e il suo complemento siano entrambi esprimibili come unioni finite di insiemi del tipo $U \cdot V^\omega$, dove sia U che V sono classi di una congruenza di indice finito su A^*.

Definizione 2.15 (*Congruenza*)

Una relazione di equivalenza $\sim$ su un generico insieme A si dice congruenza *(rispetto all'operazione di concatenazione) se, per ogni $x, y, x', y' \in A$, se $x \sim y$ e $x' \sim y'$, allora $xx' \sim yy'$.* □

Esercizio 2.16 *Dimostrare che una congruenza è una relazione di equivalenza invariante destra.*

Definizione 2.17 (*Saturazione*)

Una congruenza $\sim$ satura *un ω-linguaggio $L \subseteq A^\omega$ se, per ogni coppia U e V di $\sim$-classi, $U \cdot V^\omega \cap L \neq \emptyset$ implica $U \cdot V^\omega \subseteq L$.* □

Proposizione 2.18 *Siano $\sim$ una congruenza e $L \subseteq A^\omega$ un ω-linguaggio. Se $\sim$ satura L, allora $\sim$ satura $\overline{L}$.*

Prova

Supponiamo per assurdo che $\sim$ non saturi $\overline{L}$. Ne segue che esistono due $\sim$-classi U, V tali che $U \cdot V^\omega \cap \overline{L} \neq \emptyset$ e $U \cdot V^\omega \not\subseteq \overline{L}$. Da $U \cdot V^\omega \not\subseteq \overline{L}$ segue che esiste una ω-parola α tale che $\alpha \in U \cdot V^\omega$ e $\alpha \notin \overline{L}$ (e, quindi, $\alpha \in L$). Dato che $\sim$ satura L, da $U \cdot V^\omega \cap L \neq \emptyset$ segue che $U \cdot V^\omega \subseteq L$ (contraddizione). ■

Siano dati un automa di Büchi $\mathcal{A} = (Q, A, \Delta, q_0, F)$, $w \in A^*$ e $s, s' \in Q$. Scriviamo $s \rightarrow_w^F s'$ se esiste una computazione di $\mathcal{A}$ su w dallo stato s a s' ed almeno uno degli stati della computazione, inclusi s e s', appartiene a F. Definiamo $W_{ss'}^F = \{w \in A^*.\ s \rightarrow_w^F s'\}$.

Esercizio 2.19 *Dato un automa di Büchi $\mathcal{A} = (Q, A, \Delta, q_0, F)$, dimostrare che, per ogni $s, s' \in Q$, $W_{ss'}^F$ è regolare.*

Definizione 2.20 (*Relazione $\approx_\mathcal{A}$*)

Sia $\mathcal{A} = (Q, A, \Delta, q_0, F)$ un automa di Büchi. Definiamo una relazione $\approx_\mathcal{A}$ su A^* tale che $u \approx_\mathcal{A} v$ se, per ogni $s, s' \in Q$, $s \rightarrow_u s'$ se e solo se $s \rightarrow_v s'$ e $s \rightarrow_u^F s'$ se e solo se $s \rightarrow_v^F s'$. □

Lemma 2.21 (*Proprietà della relazione $\approx_\mathcal{A}$ - 1*)

Dato un automa di Büchi $\mathcal{A}$, la relazione $\approx_\mathcal{A}$ è una congruenza di indice finito che satura $L(\mathcal{A})$.

Prova

E' facile vedere che la relazione $\approx_{\mathcal{A}}$ è una congruenza. Inoltre, $\approx_{\mathcal{A}}$ ha indice finito in quanto $\mathcal{A}$ ha un numero finito di stati. Dimostriamo che $\approx_{\mathcal{A}}$ satura $L(\mathcal{A})$. Sia $\mathcal{A} = (Q, A, q_0, \Delta, F)$ un automa di Büchi e siano U e V due generiche classi di $\approx_{\mathcal{A}}$. Si consideri $\alpha \in U \cdot V^\omega \cap L(\mathcal{A})$. Da $\alpha \in L(\mathcal{A})$ segue che esiste una computazione di successo di $\mathcal{A}$ su α. Da $\alpha \in U \cdot V^\omega$, si ha che esistono $u \in U, v_1 \in V, v_2 \in V, \ldots$ tali che $\alpha = uv_1v_2 \ldots$. Ne segue che dalla computazione di successo di $\mathcal{A}$ su α è possibile estrarre una sottosequenza infinita di stati $q_0 s_1 s_2 \ldots$ tale che $q_0 \rightarrow_u s_1 \rightarrow_{v_1} s_2 \rightarrow_{v_2} s_3 \ldots$ e, per un numero infinito di i, $s_i \rightarrow^F_{v_i} s_{i+1}$. Sia $\beta = u'v'_1v'_2 \ldots \in U \cdot V^\omega$. Mostriamo che $\beta \in L(\mathcal{A})$. Poichè $u \approx_{\mathcal{A}} u'$ e, per ogni $i > 0$, $v_i \approx_{\mathcal{A}} v'_i$, vale che $q_0 \rightarrow_{u'} s_1 \rightarrow_{v'_1} s_2 \rightarrow_{v'_2} s_3 \ldots$ e, per un numero infinito di i, $s_i \rightarrow^F_{v'_i} s_{i+1}$. Ne segue che esiste una computazione di $\mathcal{A}$ su β in cui almeno uno stato finale si ripete un numero infinito di volte. Dunque $\beta \in L(\mathcal{A})$. ■

Corollario 2.22 (*Proprietà della relazione $\approx_{\mathcal{A}}$ - 2*)

Dato un automa di Büchi $\mathcal{A}$, la relazione $\approx_{\mathcal{A}}$ è una congruenza di indice finito che satura $\overline{L(\mathcal{A})}$.

Prova

Segue immediatamente dal Lemma 2.21 e dalla Proposizione 2.18. ■

Esercizio 2.23 *Dimostrare che la relazione $\approx_{\mathcal{A}}$ è una congruenza di indice finito.*

Si osservi che dal Teorema 1.13 segue che ogni classe di una relazione di equivalenza invariante destra di indice finito è regolare (caso particolare di unione finita di classi di equivalenza). Ogni classe della congruenza $\approx_{\mathcal{A}}$ è dunque regolare. Inoltre, per ogni parola w, la classe $[w]_{\approx_{\mathcal{A}}}$ che la contiene è l'intersezione degli insiemi $W_{s,s'}$, $W^F_{s,s'}$, $A^* \setminus W_{s,s'}$ e $A^* \setminus W^F_{s,s'}$ contenenti w (ovviamente, $w \in W_{s,s'}$ se e solo se $w \notin A^* \setminus W_{s,s'}$, $w \in W^F_{s,s'}$ se e solo se $w \notin A^* \setminus W^F_{s,s'}$ e $w \in W^F_{s,s'}$ solo se $w \in W_{s,s'}$).

Lemma 2.24 *Sia $\sim$ una congruenza su A^* di indice finito. Per ogni ω-parola $\alpha \in A^\omega$ esistono U e V, classi della congruenza $\sim$, tali che $\alpha \in U \cdot V^\omega$, con $V \cdot V \subseteq V$.*

Prova

Sia $\sim$ una congruenza su A^* di indice finito. Data una ω-parola $\alpha \in A^\omega$, diciamo che due posizioni k, k' si *riuniscono* in $m > k, k'$ se $\alpha(k, m) \sim \alpha(k', m)$. In tal caso scriviamo $k \cong^m_\alpha k'$. Dato che $\sim$ è una congruenza, $k \cong^m_\alpha k'$ implica $k \cong^{m'}_\alpha k'$ per ogni $m' > m$. Scriviamo $k \cong_\alpha k'$ se esiste m per cui $k \cong^m_\alpha k'$. La relazione $\cong_\alpha$ è una relazione di equivalenza sui naturali di indice finito in quanto $\sim$ ha indice finito. Dunque esiste una sequenza infinita $\sigma = k_0, k_1, \ldots$ di posizioni che appartengono alla stessa classe di equivalenza di $\cong_\alpha$. Dato che $\sim$ ha indice finito, esistono infiniti i tali che i segmenti $\alpha(k_0, k_i)$ appartengono tutti alla stessa classe di equivalenza di $\sim$. Passando eventualmente ad una sottosequenza infinita della sequenza data, possiamo assumere che $k_0 > 0$ e che, per ogni $i > 0$, $\alpha(k_0, k_i) \in V$ e $k_0 \cong_\alpha k_i$:

$$\exists k_0(\alpha(0, k_0) \in U \wedge \exists^\omega k(\alpha(k_0, k) \in V \wedge k_0 \cong_\alpha k)),$$

dove U è la classe della congruenza $\sim$ che contiene $\alpha(0, k_0)$. Mostriamo che $\alpha \in U \cdot V^\omega$. Dal fatto che, per ogni $i \geq 0$, $k_0 \cong_\alpha k_i$, segue che, per ogni $i \geq 0$, esiste $m > k_0, \ldots, k_i$ tale che le posizioni $k_0, \ldots, k_i$ si riuniscono in m, ossia tale che $\alpha(k_0, m) \sim \alpha(k_1, m) \ldots \sim \alpha(k_i, m)$. Passando eventualmente ad una sottosequenza infinita della sequenza data, possiamo assumere che le posizioni $k_0, \ldots, k_i$ si riuniscano in qualche $m < k_{i+1}$, e dunque in k_{i+1}. Concludiamo la prova mostrando che $\alpha(k_i, k_{i+1}) \in V$ per ogni $i \geq 0$. Per costruzione, si ha che, per ogni $i \geq 0$, $\alpha(k_0, k_i) \in V$. Da ciò immediatamente segue che $\alpha(k_0, k_1) \in V$. Per ogni $i > 0$, da $\alpha(k_0, k_{i+1}) \in V$ e dal fatto che le posizioni k_0 e k_i si riuniscono

in k_{i+1} segue che $\alpha(k_i, k_{i+1}) \in V$. Ciò consente di concludere che, per ogni $i \geq 0$, $\alpha(k_i, k_{i+1}) \in V$ e quindi $\alpha \in U \cdot V^\omega$.

Per poter concludere che $V \cdot V \subseteq V$ è sufficiente mostrare che $V \cdot V \cap V \neq \emptyset$ (in quanto V è una classe di una congruenza). Ciò segue immediatamente dal fatto che $\alpha(k_0, k_i)$, $\alpha(k_i, k_{i+1})$ e $\alpha(k_0, k_{i+1})$ appartengono a V, per ogni $i > 0$. ■

Esercizio 2.25 *Si dimostri che la relazione $\cong_\alpha$ è una relazione di equivalenza sui naturali di indice finito.*

Corollario 2.26 *Siano $L \subseteq A^\omega$ un ω-linguaggio e $\sim$ una congruenza di indice finito che satura L. Si ha che $L = \bigcup U \cdot V^\omega$, con U, V $\sim$-classi tali che $U \cdot V^\omega \cap L \neq \emptyset$.*

Prova

L'inclusione $\subseteq$ segue dal Lemma 2.24, mentre l'inclusione $\supseteq$ vale in quanto $\sim$ satura L. ■

Teorema 2.27 (*Chiusura rispetto alla complementazione - Büchi 1960*)

Se $L \subseteq A^\omega$ è ω-regolare, allora $\overline{L}(= A^\omega \setminus L)$ è ω-regolare. Inoltre, a partire da una automa di Büchi che riconosce L, è possibile costruirne uno che riconosce $\overline{L}$.

Prova

Sia $\mathcal{A}$ un automa di Büchi che riconosce L. Per il Corollario 2.22, $\approx_\mathcal{A}$ è una relazione di congruenza di indice finito che satura $\overline{L}$. Ne segue che, per il Corollario 2.26, $\overline{L} = \bigcup U \cdot V^\omega$, con U, V $\approx_\mathcal{A}$-classi tali che $U \cdot V^\omega \cap \overline{L} \neq \emptyset$. La relazione $\approx_\mathcal{A}$ è una congruenza (e quindi anche una relazione di equivalenza invariante destra) di indice finito e, di conseguenza, l'unione è finita. Dato che ogni classe di $\approx_\mathcal{A}$ è regolare, sfruttando il Teorema 2.6, possiamo concludere che $\overline{L}$ è ω-regolare.

Mostriamo ora che la costruzione dell'automa per il complementare è effettiva. Si noti che $U \cdot V^\omega \cap L = \emptyset$ se e solo se $U \cdot V^\omega \cap \overline{L} \neq \emptyset$, in quanto $\approx_\mathcal{A}$ satura L. La costruzione di un automa per $U \cdot V^\omega \cap L$ è effettiva per il Teorema 2.6 e il test $U \cdot V^\omega \cap L = \emptyset$ è decidibile per il Teorema 2.14. Infine, l'unione è finita, in quanto $\approx_\mathcal{A}$ ha indice finito, e la costruzione di un automa per l'unione è effettiva per il Teorema 2.6. ■

È interessante notare come, in virtù del Corollario 2.11 e delle proprietà chiusura dei linguaggi ω-regolari, ogni linguaggio ω-regolare risulti univocamente determinato dall'insieme delle sue parole definitivamente periodiche, ossia che se due linguaggi ω-regolari hanno lo stesso insieme di parole definitivamente periodiche, allora i due linguaggi sono uguali (è sufficiente considerare la differenza insiemistica dei due linguaggi che, per le proprietà di chiusura, è a sua volta un linguaggio ω-regolare e deve quindi contenere, per il Corollario 2.11, una parola definitivamente periodica).

Teorema 2.28 (*Decidibilità dei problemi dell'universalità, dell'inclusione e dell'equivalenza*)

Dati due automi di Büchi $\mathcal{A}$ e $\mathcal{A}'$, i seguenti problemi sono decidibili:

1. *$L(\mathcal{A}) = A^\omega$ (problema dell'universalità);*
2. *$L(\mathcal{A}) \subseteq L(\mathcal{A}')$ (problema dell'inclusione);*
3. *$L(\mathcal{A}) = L(\mathcal{A}')$ (problema dell'equivalenza).*

Prova

Si procede come nel caso del Teorema 1.9. ■

Concludiamo la sezione presentando la controparte della relazione $\sim_L$, definita per i linguaggi su parole finite, nel caso infinito.

Definizione 2.29 (*Relazione* $\approx_L$)

Dato un linguaggio $L \subseteq A^\omega$*, definiamo la relazione* $\approx_L$ *su* A^* *tale che, per ogni* $u, v \in A^*$*,* $u \approx_L v$ *se, per ogni* $x, y, z \in A^*$*, (i)* $xuyz^\omega \in L$ *se e solo se* $xvyz^\omega \in L$ *e (ii)* $x(yuz)^\omega \in L$ *se e solo se* $x(yvz)^\omega \in L$*.* □

È facile mostrare che la relazione $\approx_L$ è una congruenza. Se $u \approx_L v$ e $u' \approx_L v'$, allora, per ogni $x, y, z \in A^*$, $xuu'yz^\omega \in L$ se e solo se $xvu'yz^\omega \in L$ se e solo se $xvv'yz^\omega \in L$. Similmente, $x(yuu'z)^\omega \in L$ se e solo se $x(yvv'z)^\omega \in L$. Ne segue che $uu' \approx_L vv'$.

Lemma 2.30 *Dato un linguaggio* $L = L(\mathcal{A})$*, per un qualche automa di Büchi* $\mathcal{A}$*, la congruenza* $\approx_\mathcal{A}$ *è un raffinamento della congruenza* $\approx_L$*.*

Prova

Dobbiamo dimostrare che se $u \approx_\mathcal{A} v$, allora $u \approx_L v$. Poiché $\approx_\mathcal{A}$ satura L, si ha che $xuyz^\omega \in L$ se e solo se $[xuy]_{\approx_\mathcal{A}} \cdot [z]^\omega_{\approx_\mathcal{A}} \subseteq L$. Dato che $u \approx_\mathcal{A} v$ e $\approx_\mathcal{A}$ è una congruenza, si ha che $[xuy]_{\approx_\mathcal{A}} = [xvy]_{\approx_\mathcal{A}}$, da cui $[xuy]_{\approx_\mathcal{A}} \cdot [z]^\omega_{\approx_\mathcal{A}} \subseteq L$ se e solo se $[xvy]_{\approx_\mathcal{A}} \cdot [z]^\omega_{\approx_\mathcal{A}} \subseteq L$. Ciò consente di concludere che $xuyz^\omega \in L$ se e solo se $xvyz^\omega \in L$. Similmente, è possibile provare che $x(yuz)^\omega \in L$ se e solo se $x(yvz)^\omega \in L$. Da cui la tesi $u \approx_L v$. ■

Il risultato del Lemma 2.30 si può generalizzare ad ogni congruenza $\sim$ che satura L.

Proposizione 2.31 *Dato un linguaggio* $L \subseteq A^\omega$*, ogni congruenza* $\sim$ *che satura* L *è un raffinamento della congruenza* $\approx_L$*.*

Prova

Si può ripercorrere passo passo la dimostrazione del Lemma 2.30. ■

Teorema 2.32 (*Arnold 1985*)

Sia $L \subseteq A^\omega$ *un linguaggio di parole infinite.* L *è* ω*-regolare se e solo se la congruenza* $\approx_L$ *ha indice finito e satura* L*. Inoltre,* $\approx_L$ *è la congruenza più grossolana che satura* L*.*

Prova

Se la congruenza $\approx_L$ ha indice finito e satura L, allora, per il Corollario 2.26, $L = \bigcup U \cdot V^\omega$, con U, V $\approx_L$-classi tali che $U \cdot V^\omega \cap L \neq \emptyset$. Dato che $\approx_L$ è una congruenza di indice finito, dal Teorema 1.13 segue che le $\approx_L$-classi U, V sono regolari. La ω-regolarità di L segue dalle proprietà di chiusura dei linguaggi ω-regolari (Teorema 2.6). Viceversa, sia L ω-regolare. Ciò significa che esiste un automa di Büchi $\mathcal{A}$ tale che $L = L(\mathcal{A})$. Dal fatto che $\approx_\mathcal{A}$ è un raffinamento di $\approx_L$ (cf. Lemma 2.30) di indice finito, segue che $\approx_L$ ha indice finito. Proviamo ora che $\approx_L$ satura L, ossia che, per ogni coppia di classi U e V di $\approx_L$, se $U \cdot V^\omega \cap L \neq \emptyset$, allora $U \cdot V^\omega \subseteq L$. Data la regolarità di U e V, la ω-regolarità di $U \cdot V^\omega \cap L$ segue dal Teorema 2.6. Il Corollario 2.11 garantisce l'esistenza di una ω-parola definitivamente periodica $\alpha = xy^\omega \in U \cdot V^\omega \cap L$. Dato che $\alpha = xy^\omega \in U \cdot V^\omega$, α può essere "riscritta" in termini di un U-segmento seguito da una sequenza infinita di V-segmenti. Inoltre, dato che vi sono infiniti V-segmenti che partizionano ogni suffisso di α, esistono (almeno) due segmenti che iniziano dopo lo stesso prefisso y_1 del periodo y (vale a dire $y = y_1y_2$). Ne segue che possiamo introdurre $w = xy^m y_1 \in U \cdot V^r$, per opportuni m e r, e $z = y_2y^n y_1 \in V^s$, per opportuni n e s, tali che $\alpha = xy^\omega = wz^\omega$. Da $[w]_{\approx_L} \cap U \cdot V^r \neq \emptyset$, si ricava $U \cdot V^r \subseteq [w]_{\approx_L}$ ($U \cdot V^r$ è una classe della congruenza $\approx_L$, costruita a partire dalle $\approx_L$-classi U e V). Similmente, $V^s \subseteq [z]_{\approx_L}$. Ne segue che $U \cdot V^\omega \subseteq [w]_{\approx_L} \cdot [z]^\omega_{\approx_L}$. Per completare la prova, è sufficiente mostrare che $[w]_{\approx_L} \cdot [z]^\omega_{\approx_L} \subseteq L$. Per assurdo, supponiamo che esista $\alpha \in [w]_{\approx_L} \cdot [z]^\omega_{\approx_L} \setminus L$, con $\alpha = w_0 z_1 z_2 \ldots$, dove $w_0 \approx_L w$ e $z_i \approx_L z$, per $i \geq 1$. Dato che $[w]_{\approx_L} \cdot [z]^\omega_{\approx_L} \setminus L$ è ω-regolare, possiamo assumere che α sia definitivamente periodica e, conseguentemente, che esistano

p e q tali che $\alpha = w_0 z_1 \dots z_p (z_{p+1} \dots z_{p+q})^\omega$. Poiché $wz^p(z^q)^\omega = wz^\omega = xy^\omega \in L$, per definizione di $\approx_L$, $\alpha = w_0 z_1 \dots z_p (z_{p+1} \dots z_{p+q})^\omega \in L$ (contraddizione).

Provato che $\approx_L$ satura L, la proprietà di $\approx_L$ di essere la congruenza più grossolana che satura L segue dalla Proposizione 2.31. ■

È opportuno evidenziare come, nel teorema precedente, l'ipotesi di saturazione risulti essere indispensabile. Esistono, infatti, linguaggi $L \subseteq A^\omega$ non ω-regolari per i quali $\approx_L$ ha indice finito.

Esempio 2.33 (*Trakhtenbrot 1966*)

Sia β una ω-parola su A non definitivamente periodica e $L(\beta)$ il linguaggio di tutte e sole le ω-parole che hanno un suffisso in comune con β. Nell'Esempio 2.12 abbiamo mostrato come un tale linguaggio non sia ω-regolare. È facile vedere che, per ogni coppia di parole $u, v \in A^*$, $u \approx_{L(\beta)} v$. Infatti, $xuyz^\omega \in L(\beta)$ se e solo se $xvyz^\omega \in L(\beta)$ e $x(yuz)^\omega \in L(\beta)$ se e solo se $x(yvz)^\omega \in L(\beta)$, in quanto $L(\beta)$ non contiene parole definitivamente periodiche. Ne segue che $\approx_{L(\beta)}$ definisce un'unica classe di equivalenza (che coincide con A^*) ed ha quindi indice finito (pari a 1). E' immediato verificare che $\approx_{L(\beta)}$ non satura $L(\beta)$.

2.3 Automi di Büchi e calcolo delle sequenze

In questa sezione studieremo il legame tra gli automi di Büchi e la teoria monadica al second'ordine di un successore ($S1S$ in breve). Il risultato fondamentale che dimosteremo è l'equivalenza tra definibilità in $S1S$ e ω-regolarità.

Definizione 2.34 (*Il linguaggio della teoria monadica al second'ordine di un successore sull'alfabeto A - $S1S_A$*)

Sia A un alfabeto finito di simboli. Il linguaggio della teoria monadica al second'ordine di un successore sull'alfabeto A, $S1S_A$, è definito nel seguente modo: i termini si ottengono a partire dalla costante 0 e dalle variabili (al prim'ordine) $x, y, \dots$ mediante applicazioni dalla funzione $+1$. Le formule atomiche hanno una delle seguenti forme: $t_1 = t_2$, $t_1 < t_2$, $t_1 \in X$, $t_1 \in Q_a$ (con $a \in A$), dove t_1, t_2 sono termini, X è una variabile al second'ordine e Q_a è un simbolo di predicato unario (uno per ogni $a \in A$). Le formule di $S1S_A$ si ottengono a partire dalle formule atomiche applicando gli operatori $\neg$, $\vee$, $\wedge$ e i quantificatori $\forall$ ed $\exists$ ad entrambi i tipi di variabili (individuali e insiemistiche). □

Una formula chiusa (o enunciato) di $S1S_A$ è una formula priva di variabili libere. La semantica di $S1S_A$ e la nozione di modello sono definite in modo standard.

Osservazione 2.35 *Si noti che la costante 0 e il predicato $<$ sono definibili (e, quindi, ridondanti) in $S1S_A$. Per quanto riguarda la costante 0, possiamo definire $x = 0$ come $\forall y(x < y \vee x = y)$, facendo uso del predicato $<$ e della quantificazione al prim'ordine, o, in alternativa, come $\neg\exists y(y + 1 = x)$, facendo uso della quantificazione al prim'ordine, ma non del predicato $<$. Per quanto riguarda il predicato $<$, possiamo definire $x < y$ come $\forall X(x + 1 \in X \wedge \forall z(z \in X \rightarrow z + 1 \in X) \rightarrow y \in X)$, usando la quantificazione al second'ordine (si rilevi il ruolo fondamentale della quantificazione* universale *su X, la quale impone che, fra le interpretazioni di X considerate, vi sia anche l'insieme che contiene tutti e soli gli elementi $\{x + 1, x + 2, \dots\}$). In alternativa, possiamo definire $x < y$ come $\exists X(x \notin X \wedge x + 1 \in X \wedge \forall z(z \in X \rightarrow z + 1 \in X) \wedge y \in X)$, dove $x \notin X$ è un'abbreviazione di $\neg(x \in X)$. Si noti che, data la definibilità al second'ordine di $<$, la prima delle due definizioni di 0 usa implicitamente la quantificazione al second'ordine. Non così la seconda, che garantisce l'effettiva definibilità al prim'ordine di 0.*

Dal punto di vista logico, rappresenteremo ogni ω-parola $\alpha \in A^\omega$ come una struttura model-theoretic della forma $\underline{\alpha} = (\omega, 0, +1, <, \{Q_a\}_{a\in A})$ (interpretazione per $S1S_A$), dove il dominio ω è l'insieme dei naturali, la costante 0 è (interpretata come) il naturale 0, $+1$ è la funzione successore sui naturali, $<$ è l'usuale relazione di ordinamento dei naturali e, per ogni $a \in A$, $Q_a = \{i \in \omega.\ \alpha(i) = a\}$. Dato un enunciato φ, il linguaggio definito da φ è $L(\varphi) = \{\alpha \in A^\omega.\ \underline{\alpha} \models \varphi\}$.

Esempio 2.36 *Sia L il linguaggio sull'alfabeto $\{a, b, c\}$ che contiene tutte e sole le ω-parole tali che ogni occorrenza del simbolo a sia seguita da un'occorrenza del simbolo b. L è catturato dal seguente enunciato al prim'ordine:*

$$\forall x(x \in Q_a \to \exists y(x < y \wedge y \in Q_b)).$$

Esempio 2.37 *Sia L il linguaggio sull'alfabeto $\{a, b, c\}$ che contiene tutte e sole le ω-parole tali che tra ogni coppia di occurrenze successive del simbolo a vi sia un numero pari di occorrenze dei simboli b e c. L è catturato dal seguente enunciato al second'ordine:*

$$\forall x \forall y(x \in Q_a \wedge y \in Q_a \wedge x < y \wedge \neg\exists z(x < z \wedge z < y \wedge z \in Q_a) \to$$

$$\exists X(x \in X \wedge \forall z(z \in X \leftrightarrow z + 1 \notin X) \wedge y \notin X)).$$

La dipendenza di $S1S_A$ dal particolare alfabeto A può essere eliminata nel seguente modo. A partire da $S1S_A$, costruiamo la teoria monadica al second'ordine di un successore $S1S$ sostituendo l'alfabeto A con $\{0,1\}^n$, con $n = \lceil \log_2 |A| \rceil$, fornendo una codifica dei simboli di A in $\{0,1\}^n$, e rimpiazzando i predicati Q_a, per ogni $a \in A$, con nuove variabili insiemistiche libere $X_1, \ldots, X_n$. Ogni ω-parola $\alpha \in A^\omega$ verrà rappresentata come una struttura model-theoretic della forma $\underline{\alpha} = (\omega, 0, +1, <, \{P_i\}_{i=1\ldots n})$, con $P_k = \{i \in \omega.\ (\alpha(i))_k = 1\}$, dove $(\alpha(i))_k$ denota il k-esimo bit dell'i-esimo simbolo di α (interpretazione per $S1S$). È facile verificare che, per ogni enunciato di $S1S_A$, esiste un'equivalente formula $\varphi(X_1, \ldots X_n)$, con variabili libere $X_1, \ldots X_n$, di $S1S$. Data una formula $\varphi(X_1, \ldots X_n)$, il linguaggio definito da $\varphi(X_1, \ldots X_n)$ è $L(\varphi) = \{\alpha \in (\{0,1\}^n)^\omega.\ \underline{\alpha} \models \varphi(X_1, \ldots X_n)\}$. Un ω-linguaggio si dice definibile in $S1S$ se e solo se $L = L(\varphi)$ per qualche formula φ di $S1S$.

Lemma 2.38 (*Chiusura per proiezione*)

Sia $\psi(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$ la S1S-formula $\exists X_i \varphi(X_1, \ldots, X_n)$. Se $L(\varphi)$ è ω-regolare, allora $L(\psi)$ è ω-regolare.

Prova

Sia $\mathcal{A}$ un automa di Büchi sull'alfabeto $\{0,1\}^n$ tale che $L(\mathcal{A}) = L(\varphi)$. Costruiamo un automa di Büchi $\mathcal{A}'$ sull'alfabeto $\{0,1\}^{n-1}$ che riconosce il linguaggio $L(\psi)$. L'automa $\mathcal{A}'$ si ottiene a partire da $\mathcal{A}$ sopprimendo la i-esima componente dei simboli usati nelle transizioni di $\mathcal{A}$, ossia ogni simbolo $(j_1 \ldots j_i \ldots j_n)$ usato nelle transizioni di $\mathcal{A}$ viene sostituito dal simbolo $(j_1 \ldots j_{i-1} j_{i+1} \ldots j_n)$. Una ω-parola $\alpha = (\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n)$, con $\alpha_j \in (\{0,1\})^\omega$, per $j = 1, \ldots, i-1, i+1, \ldots, n$, è accettata da $\mathcal{A}'$ se e solo se esiste una componente $\beta \in \{0,1\}^\omega$ tale che $\alpha' = (\alpha_1, \ldots, \alpha_{i-1}, \beta, \alpha_{i+1}, \ldots, \alpha_n)$ è accettata da $\mathcal{A}$. Dunque l'automa $\mathcal{A}'$ riconosce $L(\psi)$. ■

Teorema 2.39 (*Equivalenza tra ω-regolarità e definibilità in $S1S$ su parole infinite - Büchi 1960*)

Un ω-linguaggio $L \subseteq A^\omega$ è ω-regolare se e solo se è definibile in $S1S$.

Prova

Sia $\mathcal{A} = (Q, A, \Delta, q_0, F)$ un automa di Büchi che riconosce un ω-linguaggio L su A. Assumiamo che $Q = \{0, 1, \ldots, m\}$ e $q_0 = 0$. Scriviamo un enunciato $\exists Y_0 \ldots Y_m \varphi(Y_0, \ldots, Y_m)$ di $S1S_A$ tale che, per ogni $\alpha \in A^\omega$, $\underline{\alpha}$ è un modello di $\exists Y_0 \ldots Y_m \varphi(Y_0, \ldots, Y_m)$ se e solo se α è accettata da $\mathcal{A}$ (ossia $\alpha \in L(\mathcal{A})$). Gli insiemi Y_i rappresentano le posizioni in cui la computazione assume lo stato i. Occorre

innanzitutto imporre che la computazione inizi dallo stato 0 e in ogni istante si trovi in al più uno stato (la condizione che la computazione si trovi in ogni istante in almeno uno stato è garantita dall'appartenenza iniziale allo stato 0 e dalla condizione di consequenzialità espressa dalla successiva formula φ_2). Ciò è espresso dalla formula:

$$\varphi_1(Y_0, \ldots, Y_m) = 0 \in Y_0 \wedge \bigwedge_{i \neq j} \neg\exists y(y \in Y_i \wedge y \in Y_j).$$

Inoltre, in ogni istante, la computazione deve procedere sulla base della relazione di transizione Δ di $\mathcal{A}$. Ciò è catturato dalla formula:

$$\varphi_2(Y_0, \ldots, Y_m) = \forall x \bigvee_{(i,a,j) \in \Delta} (x \in Y_i \wedge x \in Q_a \wedge x + 1 \in Y_j).$$

Infine, occorre esprimere la condizione di accettazione in base alla quale una computazione di successo deve attraversare infinite volte almeno uno stato finale. Tale condizione è espressa dalla formula:

$$\varphi_3(Y_0, \ldots, Y_m) = \bigvee_{i \in F} \forall x \exists y(x < y \wedge y \in Y_i).$$

Combinando insieme tali condizioni, si ricava che $L(\mathcal{A})$ è definito dal seguente enunciato di $S1S_A$:

$$\exists Y_0 \ldots Y_m \ (\varphi_1(Y_0, \ldots, Y_m) \wedge \varphi_2(Y_0, \ldots, Y_m) \wedge \varphi_3(Y_0, \ldots, Y_m)).$$

Per dimostrare l'implicazione opposta, procederemo in due passi: prima trasformeremo le formule di $S1S$ in formule di $S1S_0$, una variante di $S1S$ che ammette solo variabili al second'ordine e formule atomiche delle forme $X \subseteq Y$ ("X è un sottoinsieme di Y") e $Succ(X, Y)$ ("$X = \{x\}$, $Y = \{y\}$ e $x + 1 = y$"). Successivamente, mostreremo che, per ogni formula ψ di $S1S_0$, il linguaggio $L(\psi)$ è ω-regolare.

Mostriamo innanzitutto che $S1S_0$ e $S1S$ sono equivalenti dal punto di vista espressivo: per ogni formula di $S1S_0$ esiste una formula di $S1S$ ad essa equivalente e viceversa. In un verso, è sufficiente osservare come la $S1S_0$-formula $X \subseteq Y$ sia equivalente alla $S1S$-formula $\forall x(x \in X \rightarrow x \in Y)$ e la $S1S_0$-formula $Succ(X, Y)$ alla $S1S$-formula $\exists x \exists y(x \in X \wedge \neg\exists x'(x' \in X \wedge x' \neq x) \wedge y \in Y \wedge \neg\exists y'(y' \in Y \wedge y' \neq y) \wedge x + 1 = y)$. Per quanto riguarda il verso opposto, per prima cosa definiamo le seguenti abbreviazioni: scriviamo $X = Y$ per $X \subseteq Y \wedge Y \subseteq X$, $X \neq Y$ per $\neg(X = Y)$, $Sing(X)$ ("X è un singoletto") per $\exists Y(Y \subseteq X \wedge Y \neq X \wedge \neg\exists Z(Z \subseteq X \wedge Z \neq X \wedge Z \neq Y))$ e $\varphi \rightarrow \psi$ per $\neg\varphi \vee \psi$. Al fine di trasformare le formule di $S1S$ in formule di $S1S_0$ operiamo ora le seguenti trasformazioni:

1. riscriviamo termini e formule in modo che la funzione $+1$ compaia solo in formule atomiche del tipo $x+1 = y$. Ad esempio, riscriviamo $x+1 \in X$ come $\exists y(x+1 = y \wedge y \in X)$ e $(x+1)+1 \in X$ come $\exists y \exists z(x + 1 = y \wedge y + 1 = z \wedge z \in X)$;

2. eliminiamo la costante 0 e $<$ sfruttando rispettivamente la quantificazione al primo e al second'ordine (cf. Osservazione 2.35)

In tal modo otteniamo formule che contengono solo formule atomiche delle seguenti forme: $x = y$, $x + 1 = y$ oppure $x \in X$. Per ottenere formule di $S1S_0$, eliminiamo le variabili al prim'ordine utilizzando le formule $Succ(X, Y)$ e $X \subseteq Y$. Formule del tipo $x = y$ possono essere riscritte nella forma $Sing(X) \wedge Sing(Y) \wedge X = Y$. Per quanto riguarda le occorrenze di formule del tipo $x+1 = y$ e $y \in X$, esse possono essere riscritte rispettivamente come $Succ(X, Y)$ (dalla definizione di $Succ(X, Y)$ seguono sia $Sing(X)$ sia $Sing(Y)$) e $Sing(Y) \wedge Y \subseteq X$. Infine, trasformiamo le quantificazioni esistenziali (risp. universali) di variabili al prim'ordine in quantificazioni esistenziali (risp. universali) di variabili al second'ordine ristrette a insiemi singoletti: $\exists x(\ldots)$ diventa $\exists X(SING(X) \wedge \ldots)$ e $\forall x(\ldots)$

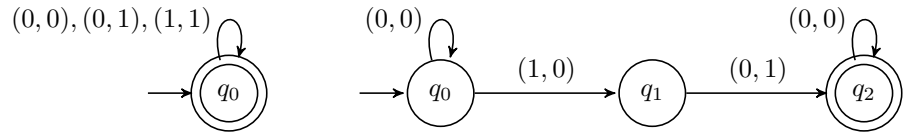

Figure 2.2: I casi base $X \subseteq Y$ e $Succ(X, Y)$.

diventa $\forall X(SING(X) \rightarrow \ldots)$. Si consideri, ad esempio, la formula $\forall x \exists y(x + 1 = y \wedge y \in Z)$. Essa può essere riscritta come $\forall X(Sing(X) \rightarrow \exists Y(Sing(Y) \wedge Succ(X, Y) \wedge Y \subseteq Z))$. È facile mostrare, per induzione sulla struttura delle formule di $S1S_0$, che, per ogni formula ψ di $S1S_0$, il linguaggio $L(\psi)$ è ω-regolare. I casi delle formule atomiche $X \subseteq Y$ e $Succ(X, Y)$ sono descritti in Figura 2.2. Per il passo induttivo basta ricordare che i linguaggi ω-regolari sono chiusi rispetto alle operazioni Booleane e alla proiezione. ■

Il risultato del Teorema 2.39 può essere interpretato come una normalizzazione delle formule di $S1S$ attraverso gli automi di Büchi: ad ogni formula $\phi \in S1S$ corrisponde un automa di Büchi $\mathcal{A}$ tale che $L(\mathcal{A}) = L(\phi)$ (implicazione da destra a sinistra); ad ogni automa di Büchi corrisponde una formula $\overline{\phi} \in S1S$ della forma $\exists Y_0 \ldots Y_m \ (\varphi_1(Y_0, \ldots, Y_m) \wedge \varphi_2(Y_0, \ldots, Y_m) \wedge \varphi_3(Y_0, \ldots, Y_m))$ tale che $L(\mathcal{A}) = L(\overline{\phi})$ (implicazione da destra a sinistra); da cui $L(\phi) = L(\overline{\phi})$, ossia $\overline{\phi}$ può essere interpretata come una forma normale per ϕ.

Il risultato di Büchi si applica, con modifiche minime della prova, al caso finito. L'interpretazione di $S1S$ su parole finite è definita nel seguente modo. Data $\alpha \in A^*$ di lunghezza $k+1$, la corrispondente interpretazione per $S1S$ è $\underline{\alpha} = (\{0 \ldots k\}, 0, +1, <, \{P_i\}_{i=1\ldots n})$, con $P_l = \{i \in \{0 \ldots k\}. \ (\alpha(i))_l = 1\}$. La funzione $+1$ deve essere ridefinita sul massimo k, ad esempio ponendo $k + 1 = k$. A differenza di quanto accade nel caso infinito, occorre inoltre considerare l'interpretazione $\underline{\epsilon}$, che corrisponde alla parola vuota ϵ, per la quale vanno adottate appropriate convenzioni (si impone che tale interpretazione soddisfi le formule universali, ma non quelle esistenziali). Si noti, infine, che in questa interpretazione di $S1S$ le variabili al second'ordine, quantificate o meno, vengono interpretate come insiemi finiti (dato che il dominio è finito).

Teorema 2.40 (*Equivalenza tra regolarità e definibilità in $S1S$ su parole finite*)
Un linguaggio $L \subseteq A^*$ è regolare se e solo se è definibile in $S1S$ al finito. ■

La teoria $S1S$ è l'insieme di tutti e soli gli enunciati di $S1S$ che sono soddisfatti nella struttura $(\omega, 0, +1, <)$. In alcuni testi, $(\omega, 0, +1, <)$ viene riscritto come $(\omega, 0, +1, <, \in)$, mettendo in evidenza la relazione $\in$ che consente di imporre l'appartenenza di un termine (individuo) ad una variabile al second'ordine (insieme). Un esempio di un enunciato appartenente alla teoria è:

$$\forall X \ \exists Y \ \forall x(x \in X \rightarrow x \in Y);$$

un esempio di enunciato che non vi appartiene è:

$$\forall X \ \exists y \ \forall x(x \in X \rightarrow x < y).$$

Dato un enunciato φ di $S1S$, possiamo applicare il Teorema 2.39 e costruire un corrispondente automa $\mathcal{A}$ *input-free*, ossia un automa le cui transizioni non sono etichettate da simboli dell'alfabeto (equivalentemente, sono tutte etichettate con lo stesso simbolo).[2] È immediato vedere che verificare

[2]Se interpretiamo gli automi di Büchi come sistemi di transizioni, gli stati dell'automa corrispondono agli stati del sistema di transizioni e i simboli dell'alfabeto alle transizioni. Un automa input-free è un automa per il quale non interessa tenere traccia delle sequenze di transizioni che corrispondono alle (eventuali) computazioni di successo: un enunciato è valido (vero) o insoddisfacibile (falso).

se $\mathcal{A}$ accetta qualche parola è un problema decidibile (cf. Teorema 2.14). Ne segue la validità del seguente teorema che stabilisce la decidibilità della teoria $S1S$.

Teorema 2.41 (*Büchi 1960*)
La verità degli enunciati di $S1S$ è decidibile.

Storicamente, tale risultato di decidibilità fu originariamente dimostrato per la teoria monadica debole al second'ordine di un successore $WS1S$ (*weak* $S1S$). Tale teoria si ottiene da $S1S$ imponendo (a livello semantico) la restrizione delle variabili (quantificate) al second'ordine a insiemi *finiti*.

È importante sottolineare come la teoria $WS1S$ possa essere definita in $S1S$ senza modificarne la semantica, ma semplicemente imponendo a livello sintattico la restrizione che ogni variabile (quantificata) al second'ordine sia interpretata come l'insieme vuoto oppure come un insieme che contiene un elemento massimo. Formalmente, per ogni variabile (quantificata) al second'ordine X, si impone che $X = \emptyset \vee \exists x(x \in X \wedge \forall y(x < y \rightarrow y \notin X)$. Dal Teorema 2.41 segue immediatamente il seguente corollario che stabilisce la decidibilità della teoria $WS1S$.

Corollario 2.42 *La verità degli enunciati di $WS1S$ è decidibile.*

Il risultato di decidibilità per $S1S$ è stato esteso in vari modi. In particolare, è stata investigata la possibilità di aggiungere funzioni e/o relazioni alla teoria $S1S$, da affiancare alla funzione successore e alla relazione d'ordine, preservandone la decidibilità. Ad esempio, si è provato che l'aggiunta del predicato unario "è potenza di k" (con $k \geq 2$) o del predicato unario "è un fattoriale" preserva la decidibilità, mentre l'aggiunta della funzione $f(x) = 2 \cdot x$, sufficiente per catturare l'intera aritmetica al prim'ordine, rende la teoria indecidibile. Un'altra estensione significativa è stata proposta da Rabin che ha dimostrato la decidibilità della teoria monadica al second'ordine di due successori $S2S$ (teoria dell'albero binario completo infinito).

2.4 Determinismo, automi di Muller e automi di Rabin

In questa sezione presenteremo due classi di automi deterministici, gli automi deterministici di Muller e gli automi deterministici di Rabin, ciascuna delle quali cattura la stessa classe di linguaggi catturati dagli automi di Büchi non deterministici (*determinization theorem*). Quale passo preliminare, mostriamo come tale ruolo non possa essere assunto dagli automi di Büchi deterministici.

Definizione 2.43 (*Automa di Büchi deterministico*)
Un automa di Büchi *deterministico è una quintupla $\mathcal{A} = (Q, A, \delta, q_0, F)$, dove Q, A, q_0 ed F sono definiti come nel caso degli automi di Büchi non deterministici e $\delta : Q \times A \rightarrow Q$ è la funzione di transizione. La* computazione *di $\mathcal{A}$ su una ω-parola α è la ω-parola σ su Q tale che $\sigma(0) = q_0$ e, per ogni $i \geq 0$, $\delta(\sigma(i), \alpha(i)) = \sigma(i+1)$. La computazione σ* ha successo *se $In(\sigma) \cap F \neq \emptyset$. L'automa $\mathcal{A}$ accetta una ω-parola α se la computazione di $\mathcal{A}$ su α ha successo. Il linguaggio $L(\mathcal{A})$ accettato da $\mathcal{A}$ è l'insieme delle ω-parole accettate da $\mathcal{A}$.* □

Non è difficile mostrare che la classe dei linguaggi riconosciuti dagli automi di Büchi deterministici è chiusa rispetto alle operazioni di unione e intersezione.

Esercizio 2.44 *Dimostrare la chiusura della classe dei linguaggi riconosciuti dagli automi di Büchi deterministici rispetto alle operazioni di unione e intersezione.*

Mostriamo ora come gli automi di Büchi deterministici non siano chiusi rispetto all'operazione di complementazione.

Proposizione 2.45 (*Caratterizzazione dei linguaggi riconosciuti dagli automi di Büchi deterministici*)

Un linguaggio $L \subseteq A^\omega$ è riconosciuto da una automa di Büchi deterministico se e solo se $L = \overrightarrow{W}$, per qualche insieme regolare $W \subseteq A^*$.

Prova

Siano $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automa di Büchi deterministico e $\mathcal{A}' = (Q, A, \delta, q_0, F)$ il corrispondente automa finito deterministico ($\mathcal{A}$ e $\mathcal{A}'$ hanno un'identica struttura). Inoltre, sia $W \subseteq A^*$ il linguaggio riconosciuto da $\mathcal{A}'$ ($W = L(\mathcal{A}')$). Mostriamo che $L(\mathcal{A}) = \overrightarrow{W}$. Sia $\alpha \in L(\mathcal{A})$. Dalla definizione di condizione di accettazione degli automi (deterministici) di Büchi, si ha che $\mathcal{A}$ accetta una ω-parola α se (e solo se) la computazione σ di $\mathcal{A}$ su α è di successo, ossia se $In(\sigma) \cap F \neq \emptyset$. Ne segue che esistono infiniti prefissi w di α tali che $w \in W$, da cui $\alpha \in \overrightarrow{W}$. Assumiamo ora che $\alpha \in \overrightarrow{W}$. Per ogni prefisso w di α appartenente a $W(= L(\mathcal{A}'))$, la computazione di successo di $\mathcal{A}'$ su w può essere vista come un prefisso della (unica) computazione σ di $\mathcal{A}$ su α ($\mathcal{A}$ è un automa di Büchi deterministico). Ne segue che $In(\sigma) \cap F \neq \emptyset$, da cui $\alpha \in L(\mathcal{A})$. ■

Si osservi come, nella prova della Proposizione 2.45, il determinismo giochi un ruolo fondamentale nella dimostrazione dell'inclusione $\overrightarrow{W} \subseteq L(\mathcal{A})$ (non gioca, invece, alcun ruolo nella dimostrazione dell'inclusione opposta).

Esercizio 2.46 *Sia $A = \{a, b\}$ e $L = \{\alpha \in A^\omega.\ \exists^{<\omega} n\ \alpha(n) = a\}$. Si costruisca un automa di Büchi non deterministico che riconosca il linguaggio L.*

Esempio 2.47 (*Linguaggio ω-regolare non riconoscibile da un automa di Büchi deterministico*)

Sia L il linguaggio dell'Esercizio 2.46. Mostriamo che L non è riconosciuto da alcun automa di Büchi deterministico. In virtù della Proposizione 2.45, è sufficiente mostrare che L non può essere scritto nella forma $\overrightarrow{W}$, per alcun insieme regolare W. Supponiamo, per assurdo, che $L = \overrightarrow{W}$, per un qualche insieme $W \subseteq A^*$ regolare. Poiché $b^\omega \in L = \overrightarrow{W}$, allora esiste n_1 tale che $b^{n_1} \in W$. Similmente, da $b^{n_1}ab^\omega \in L = \overrightarrow{W}$, segue che esiste n_2 tale che $b^{n_1}ab^{n_2} \in W$. Procedendo in tal modo, riusciamo a costruire una ω-parola $b^{n_1}ab^{n_2}ab^{n_3}\ldots$ con infiniti prefissi in W. Per definizione di $\overrightarrow{W}$, tale ω-parola appartiene a $\overrightarrow{W} = L$. Essa, però, per costruzione, contiene infinite occorrenze di a e dunque non può appartenere a L (contraddizione).

Dall'Esempio 2.47 segue che gli automi di Büchi deterministici sono *strettamente* meno espressivi degli automi di Büchi non deterministici. Inoltre, è possibile dimostrare che il linguaggio complementare $\overline{L} = \overrightarrow{(b^*a)^*}$ è ω-regolare e deterministico. Ciò prova che gli automi di Büchi deterministici non sono chiusi rispetto all'operazione di complementazione.

Esercizio 2.48 *Sia $A = \{a, b\}$ e $L = \overrightarrow{(b^*a)^*}$. Si costruisca un automa di Büchi deterministico che riconosca il linguaggio L.*

Definizione 2.49 (*Automa di Muller deterministico*)

Un automa di Muller è una quintupla $\mathcal{A} = (Q, A, \delta, q_0, \mathcal{F})$, dove Q è un insieme finito di stati, A è un alfabeto finito di simboli, $\delta : Q \times A \to Q$ è la funzione di transizione, $q_0 \in Q$ è lo stato iniziale e $\mathcal{F} \subseteq 2^Q$ è una collezione di insiemi di stati finali.

La computazione σ di $\mathcal{A}$ su una ω-parola α ha successo se (e solo se) $In(\sigma) \in \mathcal{F}$. L'automa $\mathcal{A}$ accetta α se la computazione di $\mathcal{A}$ su α è di successo. Il linguaggio $L(\mathcal{A})$ accettato da $\mathcal{A}$ è l'insieme delle ω-parole su A accettate da $\mathcal{A}$. □

$$\begin{array}{ccc} \text{Automi di Büchi NonDet} & \equiv & \text{Automi di Muller NonDet} \\ \bigcup & & \bigcup| \\ \text{Automi di Büchi Det} & \subseteq & \text{Automi di Muller Det} \end{array}$$

Figure 2.3: Relazioni tra automi di Büchi e di Muller deterministici (Det) e non deterministici (NonDet) - schema preliminare.

È facile mostrare che ogni automa di Büchi deterministico è equivalente ad un automa di Muller deterministico (d'ora in poi, useremo l'espressione automa di Muller per indicare un automa di Muller deterministico). Sia $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automa di Büchi deterministico. Un automa di Muller ad esso equivalente è l'automa $\mathcal{A}' = (Q, A, \delta, q_0, \mathcal{F})$, dove $\mathcal{F}$ contiene tutti e soli i sottoinsiemi di Q che hanno intersezione non vuota con F, ossia $\mathcal{F} = \{P \in 2^Q.\ P \cap F \neq \emptyset\}$.

La variante non deterministica dell'automa di Muller si ottiene rimpiazzando la funzione di transizione δ della Definizione 2.49 con una relazione $\Delta \subseteq Q \times A \times Q$ e riscrivendo nel modo ovvio la condizione di accettazione. Gli ω-linguaggi riconoscibili da un automa di Muller non deterministico sono definibili in $S1S$, sulla falsariga di quanto fatto per gli ω-linguaggi riconoscibili da un automa di Büchi (cf. Teorema 2.39), da cui segue facilmente la loro coincidenza con gli ω-linguaggi ω-regolari.

Esercizio 2.50 *Dimostrare che la classe degli ω-linguaggi ω-regolari coincide con la classe degli ω-linguaggi riconosciuti dagli automi di Muller non deterministici.*

Le relazioni tra automi di Büchi e di Muller deterministici e non deterministici, sulla base dei risultati sin qui ottenuti, è riassunta in Figura 2.3.

È facile provare che gli automi di Muller sono chiusi rispetto alle operazioni booleane (si procede in modo simile a quanto fatto per gli automi finiti deterministici).

Teorema 2.51 (*Proprietà di chiusura*)

Gli automi di Muller sono chiusi rispetto alle operazioni booleane.

Prova

Mostriamo che gli automi di Muller sono chiusi rispetto alle operazioni di complementazione e unione. Sia $\mathcal{A} = (Q, A, \delta, q_0, \mathcal{F})$ un automa di Muller che riconosce $L \subseteq A^\omega$. Assumiamo senza perdita di generalità che $\mathcal{A}$ sia completo (cf. Teorema 1.7). È immediato vedere che l'automa di Muller $\mathcal{A}' = (Q, A, \delta, q_0, 2^Q \setminus \mathcal{F})$ riconosce il linguaggio $A^\omega \setminus L$.

Siano $\mathcal{A} = (Q, A, \delta, q_0, \mathcal{F})$ e $\mathcal{A}' = (Q', A, \delta', q_0', \mathcal{F}')$ gli automi di Muller che riconoscono gli ω-linguaggi L e L', rispettivamente. Come nel caso della complementazione, assumiamo che entrambi gli automi siano completi. Un automa di Muller che riconosce il linguaggio $L \cup L'$ è l'automa prodotto di $\mathcal{A}$ e $\mathcal{A}'$, il cui insieme di insiemi di stati finali contiene $\{(q_1, q_1'), \ldots, (q_n, q_n')\}$ se e solo se $\{q_1, \ldots, q_n\} \in \mathcal{F}$ oppure $\{q_1', \ldots, q_n'\} \in \mathcal{F}'$. ■

È interessante osservare come la dimostrazione della chiusura rispetto alla complementazione degli automi deterministici di Muller non funzioni nel caso degli automi deterministici di Büchi. Sia $\mathcal{A} = (Q, A, \delta, q_0, F)$ un automa (completo) deterministico di Büchi. Data una ω-parola α, la computazione σ di $\mathcal{A}$ su α è di successo, e, quindi, $\alpha \in L(\mathcal{A})$, se esiste almeno uno stato $q \in F$ che si presenta infinite volte in σ. Non si può, però, escludere che anche uno stato $q' \in Q \setminus F$ si presenti infinite volte in σ. Se così fosse, α verrebbe riconosciuta anche dall'automa $\mathcal{A}' = (Q, A, \delta, q_0, Q \setminus F)$, che non può, quindi, essere un automa per il linguaggio complementare $A^\omega - L(\mathcal{A})$.

Dal Teorema 2.51 segue che gli automi (deterministici) di Muller sono strettamente più espressivi degli automi deterministici di Büchi, che non sono chiusi rispetto alla complementazione.

Il seguente lemma fornisce un'importante caratterizzazione dei linguaggi riconosciuti dagli automi di Muller (deterministici).

Lemma 2.52 *Un ω-linguaggio $L \subseteq A^\omega$ è riconosciuto da un automa di Muller se e solo se L è una combinazione Booleana (su A^ω) di insiemi $\overrightarrow{W}$, con $W \subseteq A^*$ regolare.*

Prova

Sia $\mathcal{A} = (Q, A, \delta, q_0, \mathcal{F})$ un automa di Muller che riconosce L. Sia $W_q = \{v \in A^*.\ \delta(q_0, v) = q\}$ l'insieme delle parole $v \in A^*$ per le quali esiste una computazione di $\mathcal{A}$ su v dallo stato iniziale q_0 allo stato q (insieme delle parole riconosciute dall'automa finito $\mathcal{A}_q = (Q, A, \delta, q_0, \{q\})$). Abbiamo che una ω-parola α appartiene a $\overrightarrow{W_q}$ se e solo se la computazione di $\mathcal{A}$ su α passa infinite volte per lo stato q. Dunque, per la definizione di automa di Muller, $\mathcal{A}$ accetta α se e solo se esiste $F \in \mathcal{F}$ tale che la computazione di $\mathcal{A}$ su α attraversa infinite volte tutti e soli gli stati di F se e solo se esiste $F \in \mathcal{F}$ tale che, per ogni stato $q \in F$, $\alpha \in \overrightarrow{W_q}$ e, per ogni stato $q \in Q \setminus F$, $\alpha \notin \overrightarrow{W_q}$. Dunque (scrivendo $\sim L$ per denotare $A^\omega \setminus L$):

$$L = \bigcup_{F \in \mathcal{F}} \left(\bigcap_{q \in F} \overrightarrow{W_q} \cap \bigcap_{q \in Q \setminus F} \sim \overrightarrow{W_q} \right).$$

Viceversa, supponiamo che L sia una combinazione Booleana di insiemi $\overrightarrow{W}$, con $W \subseteq A^*$ regolare. Per la Proposizione 2.45, $\overrightarrow{W}$ è riconoscibile da un automa di Büchi deterministico e, quindi, come mostrato in precedenza, anche da un automa di Muller. La tesi segue dal fatto che i linguaggi riconoscibili da automi di Muller sono chiusi rispetto alle operazioni Booleane (cf. Teorema 2.51). ■

Introduciamo ora una terza classe di automi, detti automi di Rabin, che presenta caratteristiche assai simili alle classe degli automi di Muller.

Definizione 2.53 (*Automa di Rabin deterministico*)

Un automa di Rabin deterministico è una quintupla $\mathcal{A} = (Q, A, \delta, q_0, \Omega)$, dove Q, A, q_0 e δ sono definite come per gli automi di Muller deterministici e $\Omega = \{(L_1, U_1), \ldots, (L_n, U_n)\}$, dove ogni coppia (L_i, U_i), con $L_i, U_i \subseteq Q$, è detta coppia accettante. *La computazione σ di $\mathcal{A}$ su una ω-parola α è di successo se e solo se esiste un indice j, con $1 \leq j \leq n$, tale che $In(\sigma) \cap L_j = \emptyset$ e $In(\sigma) \cap U_j \neq \emptyset$.*

La variante non deterministica dell'automa di Rabin si ottiene rimpiazzando la funzione di transizione δ della Definizione 2.53 con una relazione $\Delta \subseteq Q \times A \times Q$ e riscrivendo nel modo ovvio la condizione di accettazione. Ogni automa di Büchi (non deterministico) con insieme di stati finali F può essere visto come un automa di Rabin (non deterministico) con insieme di coppie accettanti $\Omega = \{(\emptyset, F)\}$. Inoltre, come accade per gli automi di Muller (non deterministici), gli ω-linguaggi riconoscibili da un automa di Rabin non deterministico sono definibili in $S1S$, sulla falsariga di quanto fatto per gli ω-linguaggi riconoscibili da un automa di Büchi (cf. Teorema 2.39). Ne segue che la classe degli ω-linguaggi riconoscibili da un automa di Rabin non deterministico coincide con la classe degli ω-linguaggi ω-regolari.

Mostriamo ora l'equivalenza degli automi di Muller (deterministici) e di Rabin (deterministici). A tal fine, risulta utile la seguente caratterizzazione alternativa dei linguaggi riconosciuti dagli automi di Muller.

Lemma 2.54 *Un ω-linguaggio $L \subseteq A^\omega$ è riconosciuto da un automa di Muller se e solo se $L = \bigcup_{1 \leq i \leq n} (\overrightarrow{U_i} \setminus \overrightarrow{V_i})$, con $U_i, V_i \subseteq A^*$ linguaggi regolari.*

Prova

Sia $\mathcal{A} = (Q, A, \delta, q_0, \mathcal{F})$ un automa di Muller. Il linguaggio $\mathcal{L}(\mathcal{A})$ si può esprimere come:

$$\bigcup_{F \in \mathcal{F}} \mathcal{L}(\mathcal{A}_F),$$

dove, per ogni $F \in \mathcal{F}$, $\mathcal{A}_F = (Q, A, \delta, q_0, \{F\})$ è l'automa di Muller ottenuto da $\mathcal{A}$ sostituendo $\mathcal{F}$ con l'insieme singoletto $\{F\}$. Ne segue che si può assumere, senza perdita di generalità, che $\mathcal{F}$ contenga un solo elemento F.

Il linguaggio $\mathcal{L}(\mathcal{A})$ accettato dall'automa di Muller $\mathcal{A} = (Q, A, \delta, q_0, \{F\})$ può essere espresso in termini di un insieme di linguaggi accettati da automi di Büchi deterministici nel seguente modo:

$$\mathcal{L}(\mathcal{A}) = \bigcap_{q \in F} \mathcal{L}(\mathcal{A}_q) \setminus \bigcup_{q \notin F} \mathcal{L}(\mathcal{A}_q),$$

dove $\mathcal{A}_q = (Q, A, \delta, q_0, \{q\})$. Dato che la classe dei linguaggi riconoscibili da un automa di Büchi deterministico è chiusa rispetto alle operazioni di unione e intersezione e che ogni linguaggio riconosciuto da un automa di Büchi deterministico si può esprimere come chiusura vettoriale di un linguaggio regolare, $\mathcal{L}(\mathcal{A})$ è della forma voluta.

L'implicazione opposta segue immediatamente dal Lemma 2.52: per la Proposizione 2.45, i linguaggi riconoscibili dagli automi di Büchi deterministici sono tutti (e soli) i linguaggi esprimibili come chiusure vettoriali di linguaggi regolari e, per il Lemma 2.52, tutti (e soli) i linguaggi esprimibili come una combinazione Booleana (su A^ω) di insiemi $\overrightarrow{W}$, con $W \subseteq A^*$ regolare, sono riconoscibili da un automa di Muller. ■

Teorema 2.55 *Per ogni automa di Rabin (deterministico), esiste un automa di Muller (deterministico) che accetta lo stesso linguaggio, e viceversa. Inoltre, $L \subseteq A^\omega$ è riconosciuto da un automa di Rabin con n coppie accettanti se e solo se L è esprimibile come l'unione di n differenze insiemistiche di linguaggi in A^ω riconoscibili da un automa di Büchi deterministico.*

Prova

Sia $\mathcal{A} = (Q, A, \delta, q_0, \Omega)$ un automa di Rabin (deterministico) e sia

$$\mathcal{F} = \{F \subseteq Q : \exists (L, U) \in \Omega \ (F \cap L = \emptyset \wedge F \cap U \neq \emptyset)\}.$$

Non è difficile mostrare che l'automa di Muller (deterministico) $\mathcal{A}' = (Q, A, \delta, q_0, \mathcal{F})$ riconosce il linguaggio $\mathcal{L}(\mathcal{A})$. Ciò consente di concludere che per ogni automa di Rabin (deterministico) esiste un automa di Muller (deterministico) equivalente.

La dimostrazione della direzione opposta è meno immediata. Per ogni $n \geq 1$, sia ε_n la classe dei linguaggi $L \subseteq A^\omega$ della forma $\bigcup_{1 \leq i \leq n} (\overrightarrow{U_i} \setminus \overrightarrow{V_i})$, con U_i, V_i regolari. In virtù del Lemma 2.54, per dimostrare che per ogni automa di Muller (deterministico) esiste un automa di Rabin (deterministico) equivalente è sufficiente mostrare che, per ogni $n \geq 1$, la classe di linguaggi ε_n coincide con la classe dei linguaggi riconoscibili da un automa di Rabin (deterministico) con n coppie accettanti.
Sia $\mathcal{A} = (Q, A, \delta, q_0, \Omega)$ un automa di Rabin con n coppie accettanti, con $\Omega = \{(L_j, U_j) \ : \ 1 \leq j \leq n\}$. Il linguaggio accettato da $\mathcal{A}$ può essere espresso nel modo seguente:

$$\mathcal{L}(\mathcal{A}) = \bigcup_{1 \leq j \leq n} \mathcal{L}(\mathcal{A}_j)$$

dove $\mathcal{A}_j = (Q, A, \delta, q_0, \Omega_j)$ è l'automa di Rabin che si ottiene da $\mathcal{A}$ sostituendo Ω con $\Omega_j = \{(L_j, U_j)\}$. Inoltre,

$$\mathcal{L}(\mathcal{A}_j) = \mathcal{L}(\mathcal{A}_{U_j}) \setminus \mathcal{L}(\mathcal{A}_{L_j}),$$

dove $\mathcal{A}_{U_j}$ e $\mathcal{A}_{L_j}$ sono gli automi di Büchi deterministici (Q, A, δ, q_0, U_j) e (Q, A, δ, q_0, L_j), rispettivamente. L'appartenenza di $\mathcal{L}(\mathcal{A})$ a ε_n segue immediatamente.

Per quanto riguarda l'implicazione opposta, consideriamo due linguaggi L_1 ed L_2 riconosciuti rispettivamente dagli automi di Büchi deterministici $\mathcal{A}_1 = (Q_1, A, \delta_1, q_{0,1}, F_1)$ e $\mathcal{A}_2 = (Q_2, A, \delta_2, q_{0,2}, F_2)$. Il linguaggio $L = L_1 \setminus L_2$ è riconosciuto dall'automa di Rabin $\mathcal{A} = (Q_1 \times Q_2, A, \delta, (q_{0,1}, q_{0,2}), \Omega)$, dove:

$$\delta = \{((q_1, q_2), a, (q_1', q_2')) : \delta_1(q_1, a) = q_1' \wedge \delta_2(q_2, a) = q_2'\};$$
$$\Omega = \{(Q_1 \times F_2, F_1 \times Q_2)\}.$$

Ciò consente di concludere che per ogni linguaggio L in ε_1 esiste un automa di Rabin con una sola coppia accettante che riconosce L.

Consideriamo ora un linguaggio $L \in \varepsilon_n$. Per definizione, L è l'unione di n insiemi $L^j \in \varepsilon_1$, ognuno dei quali è riconosciuto da un automa di Rabin $\mathcal{A}_j = (Q_j, A, \delta_j, q_{0,j}, \Omega_j)$, con $\Omega_j = \{(L_j, U_j)\}$. Il linguaggio L è riconosciuto dall'automa di Rabin:

$$\mathcal{A} = (Q_1 \times Q_2 \times \cdots \times Q_n, A, \delta, (q_{0,1}, q_{0,2}, \ldots, q_{0,n}), \Omega),$$

dove

$$\begin{array}{ll} \delta = & \{((q_1, \ldots, q_n), a, (q_1', \ldots, q_n')) : (q_1, a, q_1') \in \delta_1, \ldots, (q_n, a, q_n') \in \delta_n\}; \\ \Omega = & \{(Q_1 \times \ldots \times Q_{j-1} \times L_j \times Q_{j+1} \times \ldots \times Q_n, Q_1 \times \ldots \times Q_{j-1} \times U_j \times Q_{j+1} \times \ldots \times Q_n) : \\ & 1 \leq j \leq n \wedge (L_j, U_j) \in \Omega_j\}. \end{array}$$

Infatti, sia

$$\sigma = (q_{0,1}, \ldots, q_{0,n}) \xrightarrow{a_0} (q_{1,1}, \ldots, q_{1,n}) \xrightarrow{a_1} (q_{2,1}, \ldots, q_{2,n}) \cdots$$

una computazione di $\mathcal{A}$ su $\alpha = a_0 a_1 a_2 \cdots$. Per ogni $1 \leq j \leq n$, si può estrarre una computazione σ_j di $\mathcal{A}_j$ su α:

$$\sigma_j = q_{0,j} \xrightarrow{a_0} q_{1,j} \xrightarrow{a_1} q_{2,j} \cdots$$

Sia $(L, U) = (Q_1 \times \ldots \times Q_{j-1} \times L_j \times Q_{j+1} \times \ldots \times Q_n\,,\, Q_1 \times \ldots \times Q_{j-1} \times U_j \times Q_{j+1} \times \ldots \times Q_n)$ una coppia accettante in Ω, con $(L_j, U_j) \in \Omega_j$. Per definizione, $(q_{i,1}, \ldots, q_{i,n}) \in L \Leftrightarrow q_{i,j} \in L_j$ e $(q_{i,1}, \ldots, q_{i,n}) \in U \Leftrightarrow q_{i,j} \in U_j$. Ne segue che σ è una computazione di successo di $\mathcal{A}$ su α se e solo se almeno una delle computazioni σ_j è di successo per $\mathcal{A}_j$ su α. ■

Per dimostrare il risultato principale di questa sezione, ossia che la classe dei linguaggi riconoscibili da un automa di Muller (deterministico), o, equivalentemente, da un automa di Rabin (deterministico), coincide con la classe dei linguaggi riconoscibili da un automa di Büchi (non deterministico), occorre provare il seguente lemma fondamentale.

Lemma 2.56 *Ogni ω-linguaggio $L \subseteq A^\omega$ riconoscibile da un automa di Büchi è esprimibile come unione finita di insiemi della forma $\overrightarrow{U} \cap \sim \overrightarrow{V}$, con $U, V \subseteq A^*$ linguaggi regolari.*

Prova

Sia $\mathcal{A}$ un automa di Büchi che riconosce L. Sfruttando il Lemma 2.21 e il Corollario 2.26, possiamo esprimere L come:

$$L = \bigcup \{U \cdot V^\omega.\ U, V \ \approx_{\mathcal{A}}\text{-classi}, U \cdot V^\omega \cap L \neq \emptyset\}.$$

È, quindi, sufficiente provare la tesi per $L = U \cdot V^\omega$, con U e V classi di $\approx_{\mathcal{A}}$. Dal Lemma 2.24 (di cui mutuiamo la notazione), si ha che la condizione $\alpha \in U \cdot V^\omega$, con U, V $\approx_{\mathcal{A}}$-classi e $V \cdot V \subseteq V$, è equivalente alla seguente condizione:

$$\exists k_0(\alpha(0, k_0) \in U \wedge \exists^\omega k(\alpha(k_0, k) \in V \wedge k_0 \cong_\alpha k)). \tag{1}$$

Diciamo che un segmento $\alpha(k_0, m)$ è un V-testimone se, per qualche k, con $k_0 < k < m$, m è la più piccola posizione tale che $\alpha(k_0, k) \in V$ e $k_0 \cong_\alpha^m k$. Non è troppo difficile verificare che l'insieme $W_V \subseteq A^*$ dei V-testimoni è regolare. Poiché per ogni k della condizione (1) esiste un unico V-testimone $\alpha(k_0, m)$ (è richiesta la minimalità di m), la condizione (1) equivale alla seguente condizione:

$$\exists k_0(\alpha(0, k_0) \in U \wedge \exists^\omega m(\alpha(k_0, m) \text{ è un } V\text{-testimone}), \tag{2}$$

ossia $\alpha \in U \cdot \overrightarrow{W_V}$.

Mostriamo ora come la condizione (2) si possa riscrivere come una combinazione Booleana di condizioni del tipo $\exists^\omega m\ \alpha(0,m) \in W$, con W regolare. A tal fine occorre invertire l'ordine dei quantificatori $\exists k_0$ e $\exists^\omega m$ della condizione (2) per ottenere una condizione del tipo $\exists^\omega m \exists k_0 \ldots$. Inoltre, per preservare l'equivalenza tra la condizione così ottenuta e la condizione (2), dobbiamo garantire che k_0 possa essere scelto indipendentemente da m, ossia dobbiamo garantire che la stessa posizione k_0 vada bene per un'infinità di posizioni m. L'idea è quella di postulare l'esistenza di infiniti prefissi $\alpha(0,m)$ che ammettano una decomposizione della forma $\alpha(0,k_0)\alpha(k_0,m)$, con $\alpha(0,k_0) \in U$ e $\alpha(k_0,m) \in W_V$, garantendo che al crescere di m solo un numero finito di scelte diverse della posizione k_0 risulti necessario. In altri termini, occorre garantire che la scelta di k_0 sia tale che (i) k_0 risulti indipendente da m e (ii) esistano infiniti m per cui $\alpha(k_0,m) \in W_V$. Ad esempio, se scegliamo k_0 come la più piccola posizione $k < m$ tale che $\alpha(0,k) \in U$, tale scelta verifica ovviamente la condizione (i), ma non è detto che soddisfi la condizione (ii). Potrebbe, infatti, accadere che solo una data posizione $k_1 > k_0$, con k_1 appartenente ad una $\cong_\alpha$-classe diversa da quella di k_0, goda delle proprietà (i) e (ii). Supponiamo che esistano esattamente r classi dell'equivalenza $\cong_\alpha$ che contengano elementi k tali che $\alpha(0,k) \in U$, ossia che esistano r posizioni $k_1, \ldots, k_r$ tali che, per ogni i, $\alpha(0,k_i) \in U$ e non vale che $k_i \cong_\alpha k_j$ per ogni coppia $i \neq j$. Chiamiamo "caso r" questa ipotesi. La posizione k_0 può essere scelta fra le r posizioni k_i. Nell'ipotesi del "caso r", chiediamo quindi che per un'infinità di posizioni m esistano $k_1, \ldots, k_r$ tali che, per ogni k_i, valga $\alpha(0,k_i) \in U$ e, per ogni coppia k_i, k_j, con $i \neq j$, k_i e k_j non si riuniscano in m. Per garantire l'indipendenza di k_0 da m imponiamo che, al crescere di m, le posizioni k_i rimangano al di sotto di una posizione limite prefissata. Imponiamo tale vincolo garantendo che soltanto per un numero finito di posizioni m risulti necessaria una nuova scelta delle posizioni $k_1, \ldots k_r$ (o della sola posizione massima k_r). Formalmente, diciamo che k è r-appropriato per m se e solo se k è la più piccola posizione minore di m tale che k è l'ultimo elemento k_r di una r-upla $(k_1, \ldots k_r)$, con $0 < k_1 < \ldots < k_r$, $\alpha(0,k_i) \in U$, per ogni i e, per ogni coppia i,j, con $i \neq j$, non vale $k_i \cong_\alpha k_j$. Se, inoltre, $\alpha(k_i,m) \in W_V$, per qualche i, diciamo che k è r-appropriato per m via W_V. Infine, un nuovo k r-appropriato per m è una posizione r-appropriata per m (via W_V), ma, per ogni $m' < m$, non r-appropriata per m' (via W_V). Quindi, assumendo il "caso r" (si rilevi, in particolare, l'importanza del requisito di minimalità in esso contenuto), la condizione (2) equivale alla seguente condizione:

$$\exists^\omega m(\text{esiste } k \text{ } r\text{-appropriato per } m \text{ via } W_V) \wedge \exists^{<\omega} m\{\text{esiste un nuovo } k \text{ } r\text{-appropriato per } m\} \quad (3)$$

Entrambe le condizioni $(\ldots)$ e $\{\ldots\}$ dipendono solamente dal segmento $\alpha(0,m)$. Inoltre, è possibile costruire due automi finiti che riconoscono tutte e sole le parole $\alpha(0,m)$ che soddisfano, rispettivamente, le condizioni $(\ldots)$ e $\{\ldots\}$. Siano $W_r \subseteq A^*$ e $Z_r \subseteq A^*$ i linguaggi regolari riconosciuti dai due automi. La condizione (3) equivale alla condizione $\alpha \in \overrightarrow{W_r} \cap \sim \overrightarrow{Z_r}$. La disgiunzione di quest'ultima condizione al variare di r da 1 al numero finito n di classi dell'equivalenza $\cong_\alpha$ è equivalente alla condizione (2). Ciò prova che

$$U \cdot V^\omega = \bigcup_{r=1}^{n} (\overrightarrow{W_r} \cap \sim \overrightarrow{Z_r}).$$

■

Esercizio 2.57 *Dimostrare che l'insieme $W_V \subseteq A^*$ dei V-testimoni, con V classe della congruenza $\approx_{\mathcal{A}}$, è regolare.*

Il Lemma 2.56 si articola in due passaggi fondamentali: la trasformazione dei linguaggi della forma $U \cdot V^\omega$ in linguaggi della forma $U \cdot \overrightarrow{W_V}$, con $\overrightarrow{W_V}$ insieme dei V-testimoni, e la riscrittura dei linguaggi della forma $U \cdot \overrightarrow{W_V}$ come combinazioni Booleane di linguaggi del tipo $\overrightarrow{W}$, con W regolare. Il primo passaggio risulta essere relativamente semplice, il secondo decisamente più complicato. Vogliamo qui osservare come, in generale, le equivalenze $U \cdot V^\omega = U \cdot \overrightarrow{V}$ (anche sotto l'assunzione che $V \cdot V = V$)

$$\begin{array}{ccccc} \text{Automi di Büchi NonDet} & \equiv & \text{Automi di Muller NonDet} & \equiv & \text{Automi di Rabin NonDet} \\ \bigcup & & \| & & \| \\ \text{Automi di Büchi Det} & \subset & \text{Automi di Muller Det} & \equiv & \text{Automi di Rabin Det} \end{array}$$

Figure 2.4: Relazioni tra automi di Büchi, di Muller e di Rabin deterministici (Det) e non deterministici (NonDet) - schema raffinato.

e $U \cdot \overrightarrow{V} = \overrightarrow{U \cdot V}$ non valgano. Per quanto riguarda la prima equivalenza, assunto che $V \cdot V = V$, da $\alpha \in U \cdot V^\omega$ segue $\alpha \in U \cdot \overrightarrow{V}$, ma non viceversa. Da $V \cdot V = V$, infatti, non segue $V^\omega = \overrightarrow{V}$: $V \cdot V = V$ implica $V^\omega \subseteq \overrightarrow{V}$, ma non implica $\overrightarrow{V} \subseteq V^\omega$. Si consideri il seguente controesempio. Sia $V = (ab^*)^*$. È immediato vedere che $V \cdot V = V$. La stringa ab^ω appartiene a $\overrightarrow{V}$, ma non a V^ω. Mostriamo ora come anche l'uguaglianza $U \cdot \overrightarrow{V} = \overrightarrow{U \cdot V}$ non sia valida. È immediato verificare che $U \cdot \overrightarrow{V} \subseteq \overrightarrow{U \cdot V}$. Il viceversa non vale. Si consideri il seguente controesempio. Siano $U = b^*$ e $V = b$. La ω-parola b^ω appartiene a $\overrightarrow{U \cdot V}$, ma non a $U \cdot \overrightarrow{V}$ ($\overrightarrow{V}$ è il linguaggio vuoto e, quindi, tale è anche $U \cdot \overrightarrow{V}$).

Teorema 2.58 (*Teorema di McNaughton*)

Un ω-linguaggio è ω-regolare (ossia riconoscibile da un automa di Büchi) se e solo se è riconoscibile da un automa di Muller.

Prova

L'implicazione da sinistra a destra segue dai Lemmi 2.56 e 2.52. L'implicazione da destra a sinistra si ricava applicando il Lemma 2.52, la Proposizione 2.45 e il fatto che i linguaggi ω-regolari sono chiusi rispetto alle operazioni Booleane (Teoremi 2.6 e 2.27). ■

Il Teorema 2.58 consente di raffinare lo schema di Figura 2.3, relativo alle relazioni tra automi di Büchi e di Muller deterministici e non deterministici. Lo schema risultante è riportato in Figura 2.4.

Il Teorema 2.58 ha numerose interessanti conseguenze, più o meno dirette. Fra di esse va annoverato il seguente teorema che dimostra l'equivalenza espressiva (intesa come capacità di definire ω-linguaggi) di $S1S$ e $WS1S$.

Teorema 2.59 (*Espressività di $WS1S$*)

Un ω-linguaggio è ω-regolare se e solo se è definibile in $WS1S$.

Prova

Se un linguaggio è definibile in $WS1S$, allora è definibile anche in $S1S$ (cf. Sezione 2.3) e dunque, in virtù del Teorema 2.39, è ω-regolare. Viceversa, sia $L \subseteq A^\omega$ un linguaggio ω-regolare. Dal Teorema 2.58 segue che L è esprimibile come combinazione Booleana di insiemi $\overrightarrow{W}$, con W regolare. Mostriamo che è possibile caratterizzare ogni insieme del tipo $\overrightarrow{W}$, con W regolare, con una formula di $WS1S$, da cui segue immediatamente la tesi. Il Teorema 2.40 garantisce l'esistenza di una formula $\varphi(X_1, \ldots, X_n)$ di $S1S$ che definisce W qualora tale formula sia interpretata su parole finite. È facile ottenere una formula $\psi(X_1, \ldots, X_n, y)$ di $WS1S$ la quale, interpretata su parole infinite, impone che il prefisso fino alla posizione y di una ω-parola soddisfi $\varphi(X_1, \ldots, X_n)$. A tal fine basta relativizzare la quantificazione al second'ordine rispetto alle posizioni inferiori a y: ad esempio $\exists X(0 \in X)$ si riscrive come $\exists X(\forall x(x \in X \rightarrow x < y) \wedge 0 \in X)$, mentre $\forall X(0 \in X)$ si riscrive come $\forall X(\forall x(x \in X \rightarrow x < y) \rightarrow 0 \in X)$. La $WS1S$-formula che cattura $\overrightarrow{W}$ è $\forall x \exists y(x < y \wedge \psi(X_1, \ldots, X_n, y))$. ■