



Università degli Studi di Napoli

Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea Specialistica in Informatica

Lezione 6 di Calcolabilità e Complessità mod. B : Giochi dalla durata infinita

Autori

Alessandro Nisci

Matricola N97/68

Rosario Nardo

Matricola N97/67

Docenti

Prof. Aniello Murano

Ing. Fabio Mogavero

Indice

1	Definizione di arena di gioco	2
2	Definizione di gioco dalla durata infinita	4
3	Definizione di gioco di parità	5
4	Trasformazione da giochi ad automi	5
5	Trasformazioni da Automi a Giochi	9
5.1	G.Membership	9
5.2	G.Emptyness	11

1 Definizione di arena di gioco

Definizione 1.1. *Un' arena di gioco è un grafo partizionato, cioè un grafo in cui gli insiemi degli stati è partizionato in due componenti disgiunte che saranno una l'insieme dei vertici del primo giocatore e l'altra l'insieme dei vertici del secondo giocatore.*

Formalmente un'arena A , è una tupla (V_0, V_1, E) dove:

- $V = V_0 \cup V_1$
- $V_0 \cap V_1 = \emptyset$
- $E \subseteq V \times V$

Un'arena A può avere anche uno stato iniziale, in tal caso all'interno della tupla appena definita, vi sarà anche l'elemento $v_i \in V$.

Prenderemo in considerazione esclusivamente giochi a due giocatori, indicati con 0 ed 1, ed a turni, ovvero giochi in cui i due giocatori non giocheranno mai contemporaneamente ed il singolo giocatore $i \in \{0, 1\}$ potrà giocare più turni consecutivi.

Definizione 1.2. $\rho \in V^*$ è una traccia se $\forall i \in [0, |\rho|-1[$ si ha che $(\rho_i, \rho_{i+1}) \in E$

Definizione 1.3. $\pi \in V^\omega$ è un percorso se $\forall i \in \mathbb{N}$ si ha che $(V_i, V_{i+1}) \in E$.

Definizione 1.4. Con $Trk_A \subseteq V^*$ denotiamo il sottoinsieme di tutte le possibili tracce.

Definizione 1.5. Con $Pth_A \subseteq V^\omega$ denotiamo il sottoinsieme di tutti i possibili percorsi.

Definizione 1.6. $\rho \in Trk_A$ è una posizione per il giocatore i , con $i \in \{0, 1\}$, se $last(\rho) \in V_i$.

Definizione 1.7. Sia $i \in \{0, 1\}$, con $Pos_A^i \subseteq Trk_A$ indicheremo quel sottoinsieme delle tracce di A che terminano in 0 quando la posizione è di 0 e terminano in 1 quando la posizione è di 1.

Definizione 1.8. Sia $v \in V$, indicheremo con :

- $Trk_A(v) \subseteq V$, l'insieme delle tracce che partono da v .
- $Pos_A(v) \subseteq V$, l'insieme delle posizioni che partono da v .
- $Pth_A(v) \subseteq V$, l'insieme dei percorsi che partono da v .

Definizione 1.9. Una strategia per il giocatore $i \in \{0, 1\}$ è una funzione $f_i : Pos_A^i \rightarrow V_A$ tale che per ogni $\rho \in Pos_A^i$ con $\rho \in dom(f_i)$, si ha che $(last(\rho), f_i(\rho)) \in E$.

Definizione 1.10. Una strategia f per il giocatore $i \in \{0, 1\}$ si definisce v totale, con $v \in V_i$ se $Pos_A^i(v) \subseteq dom(f_i)$.

Definizione 1.11. Una strategia f per il giocatore $i \in \{0, 1\}$ è senza memoria se $\forall \rho_1, \rho_2 \in dom(f_i)$ con $last(\rho_1) = last(\rho_2)$, si ha che $f_i(\rho_1) = f_i(\rho_2)$.

Un'importante proprietà delle strategie senza memoria è la seguente: data una strategia senza memoria f , allora esiste sempre una funzione $\hat{f}_i : V_i \rightarrow V$ tale che $\forall \rho \in \text{dom}(f_i)$ si ha che $\hat{f}_i(\text{last}(\rho)) = f_i(\rho)$. Se una funzione è senza memoria allora dipende solo dallo stato finale e quindi possiamo associarla ad una funzione che dipende solo da uno stato, tralasciando gli altri.

Definizione 1.12. Str_A^i , con $i \in \{0, 1\}$, è il sottoinsieme delle funzioni $\text{Pos}_A^i \rightarrow V$ che sono strategie. Analogamente con $\text{Str}_A^i(v)$ indicheremo il sottoinsieme delle funzioni che sono strategie v totali.

Definizione 1.13. Sia $f_0 \in \text{Str}_A^0(v)$, con $v \in V_0$ e sia $f_1 \in \text{Str}_A^1$, dove f_1 deve essere v totale per tutti $i \in \text{succ}_A^*(v) \cap V_1$; date due strategie così fatte, diremo che $\pi \in \text{Pth}_A(v)$ è un play rispetto alla coppia (f_0, f_1) se $\forall i \in \mathbb{N}$ accade che:

- se $\pi_i \in V_0$ allora si ha che $\pi_{i+1} = f_0(\pi_{\leq i})$
- se $\pi_i \in V_1$ allora si ha che $\pi_{i+1} = f_1(\pi_{\leq i})$

2 Definizione di gioco dalla durata infinita

Un gioco G è definito da una coppia (A, W) dove A è un'arena e $W \subseteq V^\omega$ denota le parole vincenti, ovvero avendo una strategia per il giocatore 0 ed una per il giocatore 1, se il play che ne risulta appartiene all'insieme W , allora la vittoria sarà assegnata a 0, altrimenti ad 1. Formalmente $\pi \in Pth_A$ è vincente per 0 se $\pi \in W$, analogamente $\pi \in Pth_A$ è vincente per 1 se $\pi \notin W$.

Vediamo ora alcuni giochi particolari G definiti dalla tupla (A, C, λ, α) dove :

- A è l'arena del gioco.
- C è un insieme di colori.
- λ è una funzione di etichettatura che ad ogni vertice associa un elemento di C , cioè $\lambda : V \rightarrow C$ ¹
- $\alpha : C \rightarrow \{0, 1\}$ cioè una funzione booleana sui colori.

A differenza dei giochi definiti dalla coppia (A, W) , in un gioco particolare diremo che $\pi \in Pth_A$ è vincente per 0 se $Inf_G(\pi) \models \alpha$, dove $Inf_G(\pi)$ è l'insieme dei colori che occorrono nel gioco infinitamente spesso.

Formalmente $Inf_G(\pi) \triangleq \{ c \in C : |\{ i \in \mathbb{N} \text{ tale che } \lambda(\pi_i) = c \}| = \omega \}$.

Corollario 2.1. *I giochi particolari sono determinati, ovvero in una partita o vince il giocatore 0 oppure il giocatore 1.*

Osservazione 2.1. *Un qualsiasi gioco particolare può essere definito come un gioco $G = (A, W)$, ponendo semplicemente*

$W \triangleq \{ \pi \in Pth_A \text{ tale che } Inf_G(\pi) \models \alpha \}$

Definizione 2.1. *Il giocatore 0 vince G fortemente se esiste una $f_0 \in Str_A^0$ tale che per ogni $f_1 \in Str_A^1$, $\pi \in W$ dove π è il play della coppia (f_0, f_1) . Analogamente il giocatore 1 vince G fortemente se esiste una $f_1 \in Str_A^1$ tale che per ogni $f_0 \in Str_A^0$, $\pi \notin W$.*

Definizione 2.2. *Il giocatore 0 vince G debolmente se per ogni $f_1 \in Str_A^1$ esiste $f_0 \in Str_A^0$, $\pi \in W$ dove π è il play delle coppia (f_0, f_1) . Analogamente il giocatore 1 vince G debolmente se per ogni $f_0 \in Str_A^0$ esiste $f_1 \in Str_A^1$, $\pi \notin W$.*

La differenza sostanziale tra le due definizioni sopra citate è la seguente: nel caso di vittoria forte un giocatore, utilizzando una strategia, vince il gioco G prescindendo dalla strategia utilizzata dall'avversario mentre nel caso di vittoria debole un giocatore, usando una strategia, vince il gioco G conoscendo la strategia adoperata dall'avversario.

I giochi come sono stati definiti finora possono avere la patta.

Definizione 2.3. *Il gioco sarà patto se nessuno dei due giocatori vince fortemente, ossia entrambi vincono debolmente*

Teorema di Martin. *Se W è Boreliano allora G è determinato, ovvero non ammette patta.*

Osservazione 2.2. *Il teorema di Martin vale solo per i giochi G a turni e non per giochi concorrenti.*

¹La funzione λ potrebbe essere anche parziale e non necessariamente totale.

3 Definizione di gioco di parità

Definizione 3.1. Un gioco di parità G è definito formalmente dalla tupla (A, C, λ, α) dove :

- A è l'arena del gioco.
- C è un insieme di colori.
- λ è una funzione di etichettatura che ad ogni vertice associa un elemento di C , cioè $\lambda : V \rightarrow C$ ²
- $\alpha = \bigvee_{c \in C}^{c \bmod 2=0} c \wedge \bigwedge_{c' \in C}^{c' < c} \neg c'$.

In un gioco di parità la condizione di accettazione richiede che il più piccolo colore che occorre infinitamente spesso sia pari, infatti l'OR della formula α è vero se esiste un colore c pari mentre l'AND risulta vero se i colori c' più piccoli di c non sono mai trovati infinitamente spesso.

Un'importante proprietà per giochi di parità (proprietà di memoryless o positionality) è enunciata di seguito :

Proprietà 3.1. Se il giocatore $i \in \{0, 1\}$ vince il gioco G di parità $\Rightarrow i$ vince senza memoria.

Tale proprietà dice che il giocatore i può giocare e vincere in maniera posizionale, ovvero la mossa successiva di i dipende solo dallo stato attuale e non dalla storia della partita.

Lemma 3.1. La risoluzione dei giochi di parità è in $Np \cap Co - Np$.

Dimostrazione. Sia F l'insieme delle strategie, rappresentate da sottografi, senza memoria per il giocatore $i \in \{0, 1\}$. Si seleziona un sottografo $f \in F$ e si verifica se per il giocatore i si ha una strategia vincente. Considerando la proprietà 3.1, si evince che tale verifica è possibile farla utilizzando un certificato polinomiale che verifica, in tempo polinomiale, se nel sottografo esiste un percorso che soddisfa la condizione di parity. Per quanto detto, quindi, il gioco appartiene alla classe NP. Inoltre poiché il giocatore $\neg i$, se vince, anch'egli ha una strategia vincente $f \in F$, verificando così che anche il complemento del gioco è in NP. Da qui si ottiene che la risoluzione dei giochi di parità è in $Np \cap Co - Np$. \square

In realtà, la risoluzione dei giochi di parità è presente in una classe più ristretta rispetto a $Np \cap Co - Np$, cioè $Up \cap Co - Up$ ma la sua dimostrazione sarà omessa in questa lezione.

4 Trasformazione da giochi ad automi

Prima di entrare nel dettaglio della trasformazione, è utile osservare che se nel gioco il giocatore $i \in \{0, 1\}$ ha una strategia vincente, allora l'automata accetterà una data parola.

²La funzione λ potrebbe essere anche parziale e non necessariamente totale.

Dato un gioco $G = (A, C, \lambda, \alpha)$ la cui arena è $A = (V_0, V_1, E, v_i)$, possiamo costruire un **automa alternante di Muller**³ $A_G = \langle \{a\}, V_A, \delta, V_i, \aleph \rangle$ in cui:

- $\{a\}$ è l'alfabeto di riferimento.
- V_A è l'insieme di stati del gioco.
- δ è la funzione di transizione.
- V_i è lo stato iniziale.
- \aleph è la condizione di accettazione di Muller.

La funzione di transizione δ sarà la seguente :

$$\delta(v, a) = \begin{cases} \bigvee_{(u,v) \in E} u, & \text{se } v \in V_0 \\ \bigwedge_{(v,u) \in E} u, & \text{se } v \in V_1 \end{cases}$$

mentre la condizione di accettazione sarà :

$$\aleph = \{S \subseteq V_A \text{ tale che } \lambda(S) \models \alpha\}^4$$

dove $\lambda(S) = \{\lambda(v) \text{ tale che } v \in S\}$, con $S \subseteq Q$ e $Q = V_A$.

Verifichiamo la correttezza della costruzione appena definita :

Correttezza. *Il giocatore 0 vince il gioco $G \Leftrightarrow A_G$ accetta a^ω , dove $a^\omega \in \alpha(A_G)$.*

Dimostrazione. Il giocatore 0 vince il gioco $G \Rightarrow A_G$ accetta a^ω :

Se il giocatore 0 vince, allora per definizione esiste f_0 tale che per ogni f_1 , $Play(f_0, f_1) \models \alpha$, dove f_0 e f_1 rappresentano delle strategie rispettivamente per il giocatore 0 e per il giocatore 1. Ciò equivale a dire che $Inf_G(Play(f_0, f_1)) \models \alpha$.

³L'automa A_G lo vedremo con accettazione di Muller, però si ricordi che un automa di Muller può essere trasformato in uno Parity.

⁴Considera l'insieme di stati i cui colori soddisfano la funzione α .

Intuizione sul prosieguo della dimostrazione :

Sia $f_0 \in Str^0$, una strategia vincente per il giocatore 0, si costruisce un albero che si verifica essere un run dell'automa su a^ω ; dopodichè si controlla che i branch di tale albero soddisfino tutti le condizioni di accettazione. Sapendo che ogni branch rappresenterà un possibile play quindi una possibile strategia f_1 , se per ogni strategia f_1 si soddisfa la condizione di accettazione, allora ogni branch dell'albero deve necessariamente soddisfare la condizione di accettazione di Muller.

Sia $f_0 \in Str^0$, si costruisce un run R che è un Q -tree, così definito:

- $R_0 = \{\varepsilon\}$ rappresenta la radice dell'albero
- $R_{i+1} = R_i \cup F_0(R_i) \cup F_1(R_i)$
- $R = \bigcup_{i=0}^{\infty} R_i$

Si definisce inoltre $F_1(R)$ ed $F_0(R)$ ⁵ nel seguente modo:

$$F_1(R) \triangleq \left\{ x.u \text{ tale che } x \in Leaf(R)^6, q_0.x \in Pos_G^1(V_i), (last(V_i.x), u) \in E \right\}$$

$$F_0(R) \triangleq \left\{ x.u \text{ tale che } x \in Leaf(R), V_i.x \in Pos_G^0(V_i), u = f_0(V_i.x) \right\}$$

si verifica che l'insieme R così costruito rappresenta un run :

dato un nodo $y = last(x)$ rappresentante la posizione in quel momento dell'automa, avremo che:

- se $y \in V_0$ deve soddisfare la δ che rappresenta un OR sui successori di quel nodo, ma per costruzione si seleziona proprio uno di questi successori soddisfacendo quindi l'OR;
- se $y \in V_1$ bisogna soddisfare la δ che rappresenta un AND sui successori di quel nodo, ma per costruzione si selezionano tutti i successori del nodo, soddisfacendo quindi l'AND.

Quindi il run è corretto.

Dato un branch b qualsiasi dell'albero, si sa che $f_0 \in Str^0$ è vincente quindi, indipendentemente dalla $f_1 \in Str^1$, il giocatore 0 vince. Date f_0 ed f_1 si individua un play $\pi = v_0, v_1, v_2, \dots$ dal quale escludendo il suo nodo iniziale v_0 , si ottiene, per costruzione, esattamente uno dei possibili branch di R . Se il play π soddisfa la condizione $Inf_G(\pi(f_0, f_1)) \models \alpha$, cioè che i colori che si trovano infinitamente spesso soddisfano α , allora l'insieme dei nodi che occorrono infinitamente spesso devo soddisfare necessariamente la condizione di accettazione di Muller, poichè definita come l'insieme di stati dal quale, presa l'etichettatura, si soddisfa α cioè $Inf_G(\pi) = \lambda(Inf(\pi)) \models \alpha$ ma ciò significa che $Inf(\pi) \in \mathcal{R}$.

□

⁵ R in tale contesto rappresenta uno dei possibili R_i .

⁶ $Leaf(R)$ rappresenta l'insieme di tutti i nodi che non hanno successori.

Dimostrazione. A_G accetta $a^\omega \Rightarrow$ il giocatore 0 vince il gioco G :
 Intuizione sul prosieguo della dimostrazione :

Dato un run R accettante per A_G su a^ω , lo si analizza prendendo i nodi che diventano posizioni per il giocatore 0 quando sono concatenati alla radice. A questo punto si prende quel nodo che è presente nel run successore alla posizione di 0, e lo si inserisce come assegnamento alla funzione della strategia. Inoltre, siccome ad ogni branch corrisponde un play e viceversa e siccome ogni branch era vincente, anche il play coerente con f_0 è vincente. Proseguendo, poichè per ogni branch c'è una strategia f_1 che crea quel branch, quel play deve essere di accettazione.

Dato un run R accettante per A_G su a^ω e si definisce $f_0 \in Str_G^0(V_i)$ nel seguente modo:

$$\forall x \in R \text{ tale che } V_i.x \in Pos_G^0(V_i), \text{ si definisce } f_0(V_i.x) \in succ_R(x).$$

Preso un qualsiasi $f_1 \in Str^1$, si costruisce un play $\pi = v_0, v_1, v_2, v_3, \dots$ con f_0 ed f_1 ottenendo così un play dal quale, se elimino v_0 , si ottiene un branch b del run. Se il run R è accettante allora tutti i branch che soddisfavano la funzione di accettazione, quindi gli stati che occorreano infinitamente spesso erano in \aleph , ma, se sono in \aleph , significa che la loro etichettatura soddisfa α . A questo punto se si reinserisce v_0 non cambia nulla e quindi si ottiene un play che soddisfa la condizione di accettazione. \square

5 Trasformazioni da Automi a Giochi

Per trasformare automi in giochi vedremo due modi : il primo modo consiste nel verificare la membership di una parola ad un automa e nel costruire un gioco per la membership, cioè l'automata accetta la parola se e solo se il giocatore $i \in \{0, 1\}$ ha una strategia vincente nel gioco; un secondo modo si riduce alla verifica del vuoto di un automa ovvero l'automata accetta il linguaggio vuoto se e solo se il giocatore $i \in \{0, 1\}$ ha una strategia vincente nel gioco.

5.1 Trasformazione da automi a giochi di appartenenza

Dato un automa $A = \langle \Sigma, Q, \delta, q_0, \aleph \rangle$ e data una parola $w \in \Sigma^\omega$ si costruisce un arena $A_{A,w} = (V_0, V_1, E, V_i)$ dove :

- $V_0 = Q \times (N)$
- $V_1 = 2^Q \times (N)$
- $V_i = (q_0, 0)$
- l'insieme E sarà dato dal più piccolo sottoinsieme che soddisfa le seguenti due condizioni:
 1. $((q, i), (s, i + 1)) \in E$ con $S \models_{ex} \delta(q, w_i), \forall S \subseteq Q$
 2. $((S, i), (q, i)) \in E, \forall q \in S$

e si costruisce il gioco $G_{A,w} = (A_{A,w}, Q, \lambda, \alpha)$ dove :

- $A_{A,w}$ è l'arena del gioco.
- L'insieme di stati Q rappresenta l'insieme dei colori.
- il $dom(\lambda) = V_0$, e $\forall (q, i) \in V_0$ si avrà che $\lambda((q, i)) = q$
- $\alpha = \bigvee_{F \in \aleph} \bigwedge_{q \in F} q \wedge \bigwedge_{q \notin F} \neg q$,
cioè gli insiemi che soddisfano tale formula sono solo ed esattamente quelli contenuti in \aleph .

Verifichiamo la correttezza del gioco appena costruito :

Dimostrazione. $w \in L(A) \Rightarrow$ il giocatore 0 vince il gioco $G_{A,w}$.

Se $w \in L(A)$ per definizione esiste un run R , di A su w , accettante che è un Q -tree. Sfruttando le informazioni del Q -tree, si costruisce una strategia vincente per il giocatore 0 nel gioco $G_{A,w}$. Siccome la strategia è una funzione $f_0 : Pos_{G_{A,w}}^0(V_i)$ bisogna assegnare un valore ad ogni traccia ρ che appartiene all'insieme di posizioni appena definito. Quindi, per ogni $\rho \in Pos_{G_{A,w}}^0(V_i)$ si dovrà definire $f_0(\rho)$ che conterrà un insieme di stati che soddisfano la δ . Una posizione $\rho \in Pos_{G_{A,w}}^0(V_i)$ sarà data da $(q^0, 0)(S^0, 1)(q^1, 1)(S^1, 2) \dots (q^{n-1}, n)$. Definiamo $x = q^0.q^1.q^2 \dots q^{n-1}$; ovviamente non tutte le parole x saranno contenute nel run R , poichè se non è contenuta nel run significa che la traccia ρ non è uno dei possibili prefissi del percorso che si va a costruire e quindi non è interessante; se invece è contenuta, essa rappresenterà un possibile prefisso

di un ipotetico play vincente. Quindi dopo aver ricavato la x , se $x \notin R$ allora definisco $f_0(\rho_x)$ con un qualsiasi valore, mentre se $x \in R$ definisco $f_0(\rho_x)$ in maniera tale da soddisfare la δ . Si prendono i successori del run su x sapendo che $\text{succ}_R(x) \models \delta(\text{last}(q_0.x), w_n)$ essendo R un run. Da ciò si può dedurre che $\exists S \subseteq \text{succ}_R(x)$ tale che $S \models_{ex} \delta(\text{last}(q_0.x), w_n)$. Ma è banale vedere che l'insieme S così ottenuto è coerente con la E . A questo punto si può definire $f_0(\rho) \models S$.

Per dimostrare che la strategia f_0 così costruita è vincente, si prende una qualsiasi strategia f_1 e le si fa giocare contro, ottenendo così un play π . Così come abbiamo fatto con ρ , si proietta π ottenendo un branch del run, per costruzione di f_0 , ma, se è un branch del run, allora il run è accettante; ciò significa che ogni branch soddisfa la condizione di accettazione. Se il branch derivato dal play π è accettante allora π deve soddisfare \aleph per costruzione. Quindi f_0 è vincente per il giocatore 0. □

Dimostrazione. Il giocatore 0 vince il gioco $G_{A,w} \Rightarrow w \in L(A)$

Se il giocatore 0 vince $G_{A,w}$ allora esiste $f_0 \in \text{Str}_{G_{A,w}}^0(V_i)$ vincente. Si costruisce un run R nel seguente modo:

- $R_0 = \{\varepsilon\}$ rappresenta la radice dell'albero
- $R_{i+1} = R_i \cup F_0(R_i) \cup F_1(R_i)$
- $R = \bigcup_{i=0}^{\infty} R_i$

Sia $f_0 \in \text{Str}^0$, tramite le due strategie f_0 ed f_1 , si costruisce un play π ; tale ragionamento sarà applicato su ogni $f_1 \in \text{Str}^1$ ottenendo così un insieme di play P . Si seleziona un $\pi \in P$ e si prende una traccia qualsiasi $\pi_{\leq i}$; a tale traccia si associa una parola x come visto nella precedente dimostrazione. Tale x sarà già contenuta nell' i -esimo livello del run e la $|x| = i$. Iterativamente, supponendo di aver già costruito fino al livello i -esimo, si ha che per la traccia ρ si ottiene una x la cui lunghezza è i , ciò significa che x è presente nell' i -esimo livello del run R . I successori S di x saranno costituiti dall'insieme $S = f_0(\rho)$, cioè $S \subseteq F(R_i) = \bigcup_{x \in \text{Leaf}(R_i)} f_0(\rho_x)$, si noti che se $x \in \text{Leaf}(R_i)$ sappiamo che tale parola avrà una corrispondenza 1 ad 1 con una traccia $\rho_x \in \text{Pos}_{G_{A,w}}^0(V_i)$. Ripetendo il ragionamento per ogni livello si costruisce un run, o meglio, si sta verificando la condizione di costruzione del run, cioè che, dato un $v \in V$, i successori di v rappresentano un insieme che soddisfa la δ . Si verifica che il run appena costruito è accettante.

Ogni branch deve soddisfare la condizione di accettazione ma, per come sono stati costruiti, ogni branch ha una relazione 1 ad 1 con un play π ; ciò significa che π era vincente, e siccome il branch è la proiezione sulle lettere di indice pari sulle componenti q , tali componenti q che occorrono infinitamente spesso soddisfano la α perchè sono vincenti per il gioco. Ricordiamo che α è costruita come una formula soddisfatta solo dagli insiemi contenuti in \aleph . La prima parte dell'insieme delle componenti di indici pari di questo gioco soddisfa la α , quindi tale insieme è per costruzione contenuto in \aleph , ma tale insieme è anche l'insieme degli stati del branch relativo nel run che occorre infinitamente spesso, quindi l'Inf di quel branch sarà contenuto in \aleph e per definizione significa che accetto la parola.

□

5.2 Trasformazione da automi a giochi di vuoto

Sia $A = \langle \Sigma, Q, \delta, q_0, \aleph \rangle$ un automa alternante, costruiamo l'arena $A_A = (V_0, V_1, E, V_i)$ dove :

- $V_0 = Q$
- $V_1 = 2^Q$
- $V_i = q_0$
- l'insieme E sarà dato dal più piccolo sottoinsieme che soddisfa le seguenti due condizioni:
 1. $(q, S) \in E$ dove $q \in V_0$ e $S \in V_1 \iff \exists \sigma \in \Sigma$ tale che $S \models_{ex} \delta(q, \sigma)$,
con $S \in Q$
 2. $(S, q) \in E \iff q \in S$

e costruiamo il gioco $G_A = (A_A, Q, \lambda, \alpha)$ dove :

- A_A è l'arena del gioco.
- L'insieme di stati Q rappresenta l'insieme dei colori.
- il $dom(\lambda) = V_0$, e $\lambda(q) = q$
- $\alpha = \bigvee_{F \in \aleph} \bigwedge_{q \in F} q \wedge \bigwedge_{q \notin F} \neg q$,
cioè gli insiemi che soddisfano tale formula sono solo ed esattamente quelli contenuti in \aleph .

Vediamo ora la correttezza del gioco appena costruito :

Dimostrazione. Se $\exists w \in L(A) \Rightarrow$ il giocatore 0 vince il gioco G_A .

La dimostrazione è simile a quella della riduzione al gioco della Membership. Se $w \in L(A)$ effettuo la medesima costruzione della strategia vincente f_0 . La differenza consiste nel riverificare che esiste un σ che, inserito nella delta, mi permette di dire $S \models_{ex} \delta(last(q_0.x), w_n)$. Ma come σ si prende proprio l' i -esima lettera della parola w , dove i è la lunghezza della parola x nel run corrispondente alla traccia ρ . Quindi data la traccia ρ a cui è associata una parola x di lunghezza i , si inserisce l'insieme S all'interno di F , quindi per dire che f_0 è una strategia corretta si deve verificare che S è un insieme che soddisfa esattamente δ per un qualche σ . Ma come σ è stato inserito proprio l' i -esima lettera, quindi f_0 è una strategia corretta. Il continuo della dimostrazione è esattamente uguale alla dimostrazione della membership. □

Dimostrazione. Se il giocatore 0 vince il gioco $G_A \Rightarrow \exists w \in L(A)$.

Avendo una strategia vincente f_0 , per ogni traccia ρ , prefisso del percorso, la strategia dice dove andare; per definizione di gioco ciò significa che esiste un σ che, se inserita nella δ , permetterà a quest'ultima di essere soddisfatta dalla strategia f_0 . Presa σ , si è in posizione i , e la si considera come l' i -esima lettera della parola w . Quando si determina il run R , per costruzione sarà proprio il run di A su w . A questo punto bisogna far vedere che è accettante, ma ciò è identico a quanto fatto vedere nella dimostrazione per il gioco della Membership. \square