

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea Magistrale in Ingegneria dell'Automazione

Tesi di Laurea Magistrale

BOOLEAN CONTROL NETWORKS E LORO
APPLICAZIONI ALLA TEORIA DEI GIOCHI

Laureando:
Sorarù Nicola

Relatrice:
Prof. Maria Elena Valcher

8 LUGLIO 2019

ANNO ACCADEMICO 2018–2019

Indice

Introduzione	1
1 Conoscenze preliminari	5
1.1 Prodotto Semi-tensore: definizione e proprietà	5
1.1.1 Proprietà del prodotto semi-tensore	8
1.2 Matrici Logiche	9
1.3 Matrici non negative	12
1.4 Il Prodotto di Khatri-Rao	14
2 Boolean Control Networks	15
2.1 Funzioni Booleane in forma algebrica	15
2.1.1 Logica Booleana	15
2.1.2 Le matrici di struttura	17
2.1.3 Sistemi di equazioni logiche	22
2.2 Boolean Networks	27
2.2.1 Rappresentazione tramite grafi	29
2.2.2 Boolean networks seriali	32
2.3 Boolean Control Networks	34
2.3.1 Rappresentazione tramite grafi	37
2.4 Ricostruzione	38
3 Proprietà delle reti Booleane	43
3.1 Punti fissi e cicli limite nelle BNs	44
3.1.1 Analisi temporale	47
3.2 Divisione per classi	50
3.2.1 Attrattori e bacini di attrazione	50
3.2.2 Accessibilità e classi di comunicazione	51
3.3 Stabilità nelle BNs	54
3.3.1 Cicli globalmente attrattivi	55
3.4 Raggiungibilità e controllabilità nelle BCNs	61
3.4.1 Raggiungibilità	62

3.4.2	Controllabilità	64
3.5	Stabilizzabilità	66
3.6	Controllo tramite retroazione dallo stato	69
4	Applicazione alla teoria dei giochi	75
4.1	K-valued networks	75
4.1.1	Sistemi di equazioni logiche a k valori	76
4.1.2	Reti logiche a k valori	77
4.2	Breve Introduzione alla teoria dei giochi	78
4.2.1	Definizione e rappresentazione di gioco	80
4.2.2	Esempi di gioco strategico	82
4.2.3	Equilibrio di Nash	85
4.3	Networked Evolutionary Games	85
4.3.1	Dinamica di un NEG	89
4.3.2	NEGs omogenei	95
4.4	Controllo dei NEGs	101
	Conclusioni	109
	Bibliografia	113

Introduzione

Questo elaborato tratta l'argomento delle Boolean networks, particolari sistemi dinamici in cui le variabili sono Booleane e le funzioni che ne regolano la dinamica sono funzioni logiche. Nel momento in cui nelle reti Booleane vengono introdotti degli ingressi esterni (anch'essi Booleani) si parla di Boolean control networks.

La prima applicazione delle reti Booleane in campo biologico si deve a Kauffman che, nel 1962 [22], modellò i geni come variabili Booleane. Da allora l'interesse per le BCNs si è sempre espanso, motivato dalla grande quantità di processi fisici e non in cui le variabili possono assumere soltanto due valori (acceso/spento, alto/basso, 1/0, ...). Tra i campi in cui le reti Booleani sono applicate ci sono, oltre il campo biologico, il campo neurologico, il campo ecologico e lo studio dei sistemi multi-agente. Recentemente Cheng et al. [1] hanno sviluppato uno strumento algebrico, il prodotto semitensoriale, che permette di esprimere le BCNs mediante sistemi bilineari a tempo discreto. Ciò ha contribuito ad un incremento delle applicazioni delle reti Booleane; un esempio è il campo della Teoria dei giochi, dove è possibile modellare particolari tipi di giochi, i Networked Evolutionary Games (NEGs), mediante BCNs, come illustrato da Cheng, He, Qi e Xu in [14]. A tale settore verrà dedicato un capitolo di questo elaborato. L'ambito nel quale le reti Booleane hanno riscosso più successo rimane comunque quello della regolazione genica, del quale si riporta un breve esempio.

Il ciclo cellulare è una sequenza temporale di eventi che avvengono nella cellula e portano alla sua divisione e duplicazione. Il ciclo di una cellula è suddiviso in fasi ben definite; la replicazione del DNA avviene durante la fase di Sintesi (indicata con S), mentre durante la fase di Mitosi (indicata con M) l'energia cellulare è focalizzata nella divisione della cellula in due figlie. Le fasi di Sintesi e Mitosi sono separate da due fasi intermedie, G1 che intercorre tra M e S e G2 tra S e M. Nel caso durante la fase G1 ci sia un'assenza di stimoli, si presenta una quinta fase, indicata con G0, durante la quale la cellula rimane in uno stato di quiete. La divisione cellulare delle cellule dei mammiferi è strettamente controllata e la coordinazione del ciclo cellulare è ottenuta mediante dei segnali, positivi e negativi, il cui equilibrio decide se una cellula rimane nella fase di quiete G0, oppure si divide.

I segnali positivi o i fattori di crescita provocano l'attivazione della proteina Cyclin D (CycD) nella cellula.

La rete centrale che regola il ciclo cellulare dei mammiferi può essere modellata mediante una BCN, come discusso da Fauré, Naldi, Chaouiya e Thieffry in [21]. In particolare il modello include un singolo ingresso Booleano $\mu(t)$ che corrisponde allo stato (attivo/non attivo) della proteina CycD, e nove variabili di stato Booleane $\chi_1(t), \chi_2(t), \dots, \chi_9(t)$ che rappresentano rispettivamente l'attività o inattività al tempo t delle proteine: Rb, E2F, CycE, CycA, p27, Cdc20, Cdh1, UbcH10 e CycB. Indicando con \wedge, \vee e $\bar{\cdot}$ rispettivamente le operazioni logiche AND, OR e NOT, la BCN che descrive l'interazione tra le proteine è

$$\left\{ \begin{array}{l} \chi_1(t+1) = (\bar{\mu}(t) \wedge \bar{\chi}_3(t) \wedge \bar{\chi}_4(t) \wedge \bar{\chi}_9(t)) \vee (\chi_5(t) \wedge \bar{\mu}(t) \wedge \bar{\chi}_9(t)) \\ \chi_2(t+1) = (\bar{\chi}_1(t) \wedge \bar{\chi}_4(t) \wedge \bar{\chi}_9(t)) \vee (\chi_5(t) \wedge \bar{\chi}_1(t) \wedge \bar{\chi}_9(t)) \\ \chi_3(t+1) = \chi_2(t) \wedge \bar{\chi}_1(t) \\ \chi_4(t+1) = (\bar{\chi}_2(t) \wedge \bar{\chi}_1(t) \wedge \bar{\chi}_6(t) \wedge (\chi_7(t) \wedge \bar{\chi}_8(t))) \\ \quad \vee (\chi_4(t) \wedge \bar{\chi}_1(t) \wedge \bar{\chi}_6(t) \wedge (\chi_7(t) \wedge \bar{\chi}_8(t))) \\ \chi_5(t+1) = (\bar{\mu}(t) \wedge \chi_3(t) \wedge \bar{\chi}_4(t) \wedge \bar{\chi}_9(t)) \\ \quad \vee (\bar{\chi}_5(t) \wedge (\chi_3(t) \wedge \chi_4(t)) \wedge \bar{\mu}(t) \wedge \bar{\chi}_9(t)) \\ \chi_6(t+1) = \chi_9(t) \\ \chi_7(t+1) = (\bar{\chi}_4(t) \wedge \bar{\chi}_9(t)) \vee \chi_6(t) \vee (\chi_5(t) \wedge \bar{\chi}_9(t)) \\ \chi_8(t+1) = \bar{\chi}_7(t) \vee (\chi_7(t) \wedge \chi_8(t) \wedge (\chi_6(t) \vee \chi_4(t) \vee \chi_9(t))) \\ \chi_9(t+1) = \bar{\chi}_6(t) \wedge \bar{\chi}_7(t). \end{array} \right.$$

Tramite lo studio di questa BCN è immediato constatare gli effetti dello stato di una proteina sulle altre; ad esempio può essere di grande interesse valutare in quali condizioni $\chi_1(t) = 0$, ovvero capire quando Rb, che ha la funzione di soppressore tumorale, è disattivata.

La tesi è strutturata nei seguenti capitoli:

- Nel primo capitolo vengono introdotti i principali strumenti algebrici necessari a modellare la dinamica di una rete Booleana tramite un sistema lineare a tempo discreto. Particolare attenzione è riservata al prodotto semitensooriale, ma vengono richiamati anche argomenti quali le matrici logiche ed i sistemi positivi.
- Nel secondo capitolo vengono richiamati brevemente i concetti principali della logica Booleana per poi studiare come poter ricavare la forma algebrica di un sistema di equazioni logiche. Sono poi presentate le Boolean networks e le Boolean control networks: per descrivere la dinamica di entrambe viene ricavato il modello in spazio di stato ed è illustrata una loro rappresentazione mediante l'utilizzo di grafi.

- Nel terzo capitolo sono studiate diverse proprietà sia di Boolean networks che di Boolean control networks, quali stabilità, controllabilità, raggiungibilità e stabilizzabilità. Un altro argomento trattato è la stabilizzazione delle reti Booleane mediante retroazione dallo stato.
- Nel quarto capitolo viene estesa la descrizione delle reti Booleane alle reti logiche a k valori per poi passare all'applicazione di queste ultime alla Teoria dei Giochi. Dopo una breve introduzione a tale teoria, sono approfonditi i Networked Evolutionary Games, mediante l'applicazione degli strumenti esposti nei capitoli precedenti.

Capitolo 1

Conoscenze preliminari

In questo capitolo verranno introdotte tutte le conoscenze matematiche necessarie alla comprensione delle reti Booleane. Sarà presentato innanzitutto un tipo di prodotto tra matrici relativamente recente, chiamato prodotto semitensoriale, del quale saranno anche approfondite le proprietà. Saranno poi introdotte le matrici logiche, le loro proprietà ed il loro “rapporto” con il prodotto semitensoriale. Sarà infine richiamato il concetto di matrice non negativa, utile allo studio di determinate proprietà delle reti Booleane.

1.1 Prodotto Semi-tensore: definizione e proprietà

Il prodotto tra matrici convenzionale richiede che il numero di colonne della prima matrice sia uguale al numero di righe della seconda matrice; tale condizione limita il suo utilizzo, rendendo necessaria una sua generalizzazione applicabile anche nei casi in cui le dimensioni delle due matrici non corrispondano. Il prodotto semi-tensore nasce come naturale estensione del classico prodotto tra matrici e, come verrà approfondito in questa sezione, non richiede alcuna condizione sulla dimensione delle matrici coinvolte.

Il prodotto semi-tensore è stato introdotto da un gruppo di ricercatori guidato dal Prof. D. Cheng nel 2009 e deve il suo nome alla parziale somiglianza col prodotto tensoriale, conosciuto come prodotto di Kronecker. Mentre il prodotto convenzionale si ottiene come somma di prodotti, elemento per elemento, tra vettori riga e vettori colonna, ed il prodotto di Kronecker corrisponde ad una moltiplicazione elemento per matrice, il prodotto semitensoriale effettua entrambe le operazioni, “aggiustando” e quindi rendendo compatibili le dimensioni di righe e colonne delle matrici.

Tale prodotto trova applicazioni in svariati campi: esso ha permesso, ad esempio,

di poter maneggiare dati di qualsiasi dimensione finita, al contrario del prodotto convenzionale che permetteva di trattare solamente dati al più bidimensionali. Ad oggi, l'applicazione in cui è più utilizzato resta quella delle funzioni Booleane, permettendo di rappresentare un sistema di equazioni Booleane tramite forma algebrica, come verrà approfondito nel Capitolo 2.

Prima di introdurre formalmente il prodotto semitensoriale, viene riportato un esempio in cui il prodotto convenzionale non è ben definito.

Esempio 1.1.1. Si considerino 4 vettori di dimensione n : $X, Y, Z, W \in \mathbb{R}^n$. L'espressione $X Y^T Z W^T \in \mathbb{R}^{n \times n}$ risulta ben definita. Tuttavia, considerando che $Y^T Z$ è uno scalare e che il prodotto convenzionale tra matrici gode della proprietà commutativa rispetto alla moltiplicazione per uno scalare e della proprietà associativa, la stessa espressione può essere riscritta come:

$$\begin{aligned} (X Y^T)(Z W^T) &= X (Y^T Z) W^T \\ &= (Y^T Z)(X W^T) \\ &= Y^T (Z X) W^T . \end{aligned} \quad (1.1)$$

Risulta evidente che ora l'espressione non è più chiara, non essendo definito il prodotto $Z X$. △

Consideriamo ora due matrici $A \in \mathbb{R}^{r_1 \times c_1}$ e $B \in \mathbb{R}^{r_2 \times c_2}$; nel caso in cui risulti $c_1 = nr_2$ verrà utilizzata la notazione $A \succ_n B$ mentre nel caso in cui $r_2 = nc_1$ allora verrà usata la scrittura $A \prec_n B$.

La seguente definizione parte definendo il prodotto semi-tensore tra un vettore riga ed un vettore colonna per poi estenderlo al caso di matrici.

Definizione 1.1 (Prodotto semi-tensore). Si consideri il vettore riga $X = (x_1, \dots, x_p) \in \mathbb{R}^{1 \times p}$ ed il vettore colonna $Y = (y_1, \dots, y_q)^T \in \mathbb{R}^q$. Si prenda in considerazione il caso in cui $X \succ_n Y$, ovvero $p = nq$, allora X può essere partizionato in q blocchi di lunghezza n nel seguente modo:

$$X = (X^1, \dots, X^q), \quad X^i \in \mathbb{R}^{1 \times n}, \quad i = 1, \dots, q. \quad (1.2)$$

L'espressione:

$$X \times Y := \sum_{i=1}^q X^i y_i \in \mathbb{R}^{1 \times n} \quad (1.3)$$

si definisce prodotto semi-tensore sinistro tra X e Y . Equivalentemente, nel caso in cui $X \prec_n Y$, è Y ad essere partizionato in p blocchi di lunghezza n ed il prodotto semi-tensore sinistro tra X e Y diventa:

$$X \times Y := \sum_{i=1}^p x_i Y^i \in \mathbb{R}^n. \quad (1.4)$$

Si considerino ora le matrici $A \in \mathbb{R}^{r_1 \times c_1}$ e $B \in \mathbb{R}^{r_2 \times c_2}$ tali che sia verificato $A \succ_n B$ o $A \prec_n B$, allora $C = A \times B$ si definisce prodotto semi-tensore sinistro tra le matrici A e B . La matrice risultante C consta di $r_1 \times c_2$ blocchi, calcolabili tramite

$$C^{ij} = \text{Row}_i(A) \times \text{Col}_j(B) \quad i = 1, \dots, r_1, \quad j = 1, \dots, c_2. \quad (1.5)$$

Risulta evidente che il prodotto semitensoriale diventa il prodotto convenzionale tra matrici nel caso in cui $c_1 = r_2$, il che rende naturale vedere il primo prodotto come estensione del secondo. Nella Definizione 1.1 viene richiesto che $A \succ_n B$ o $A \prec_n B$, non permettendo di applicare il prodotto semi-tensore a matrici di dimensione arbitraria.

Per poterne estendere l'utilizzo a matrici di dimensioni qualsiasi, si rende necessario l'utilizzo del prodotto di Kronecker, che viene brevemente richiamato nelle seguente definizione.

Definizione 1.2. Date due matrici $A \in \mathbb{R}^{r_1 \times c_1}$ e $B \in \mathbb{R}^{r_2 \times c_2}$, viene definito prodotto di Kronecker tra A e B l'espressione:

$$A \otimes B := \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1c_1}B \\ a_{21}B & a_{22}B & \dots & a_{2c_1}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{r_11}B & a_{r_12}B & \dots & a_{r_1c_1}B \end{bmatrix} \in \mathbb{R}^{r_1 r_2 \times c_1 c_2}, \quad (1.6)$$

dove a_{ij} indica l'elemento in posizione (i, j) della matrice A .

È ora possibile dare la definizione di prodotto semi-tensore tra matrici di dimensione arbitraria.

Definizione 1.3. Date due matrici $A \in \mathbb{R}^{r_1 \times c_1}$ e $B \in \mathbb{R}^{r_2 \times c_2}$, sia $T = \text{m.c.m.}(c_1, r_2)$, il prodotto semi-tensore sinistro tra A e B è definito come segue:

$$A \times B := (A \otimes I_{T/c_1})(B \otimes I_{T/r_2}). \quad (1.7)$$

Anche in questo caso è possibile notare che, nel caso in cui $c_1 = r_2$, il prodotto semi-tensore diventa il prodotto convenzionale tra matrici.

Naturalmente nel caso in cui $A \succ_n B$ o $A \prec_n B$ le due definizioni di prodotto semitensoriale si equivalgono; per convincersene si consideri il seguente esempio.

Esempio 1.1.2. Siano dati $X = [1 \ 2 \ 2 \ 1] \in \mathbb{R}^{1 \times 4}$ e $Y = [-2 \ 1]^T \in \mathbb{R}^2$, si ha dunque che $A \succ_2 B$ e dalla prima definizione si ottiene:

$$X \times Y = [1 \ 2] (-2) + [2 \ 1] (1) = [0 \ -3]. \quad (1.8)$$

Dalla seconda definizione, considerando che $T = \text{m.c.m}(4, 2) = 4$ si ottiene lo stesso risultato, ovvero:

$$\begin{aligned} X \times Y &= (X \otimes I_1)(Y \otimes I_2) \\ &= [1 \ 2 \ 2 \ 1] \begin{bmatrix} -2 & 0 \\ 0 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= [0 \ -3]. \end{aligned} \tag{1.9}$$

△

Osservazione 1. Nelle Definizioni 1.1 e 1.3, si fa riferimento al prodotto semi-tensore *sinistro*; questa specificazione si rende necessaria in quanto esiste anche il prodotto semi-tensore *destra*, rappresentato dal simbolo \times : essi godono di proprietà molto simili, tuttavia in questa tesi non verrà mai utilizzato il prodotto semi-tensoriale destro e quando si farà riferimento al prodotto semi-tensore si intenderà sempre quello sinistro.

1.1.1 Proprietà del prodotto semi-tensore

Come è facile constatare, il prodotto semitensoriale mantiene invariate quasi tutte le proprietà di cui gode il prodotto usuale tra matrici. Nel seguito vengono esposte le proprietà più significative.

Proposizione 1.1.1. *Il prodotto semi-tensore sinistro gode delle seguenti proprietà:*

1. *Commutatività rispetto alla moltiplicazione per uno scalare:*

$$(\alpha A) \times (\beta B) = \alpha\beta(A \times B), \tag{1.10}$$

2. *Associatività:*

$$A \times (B \times C) = (A \times B) \times C, \tag{1.11}$$

3. *Distributività rispetto all'addizione tra matrici:*

$$(\alpha A + \beta B) \times C = \alpha(A \times C) + \beta(B \times C), \quad \alpha, \beta \in \mathbb{R}, \tag{1.12}$$

$$A \times (\beta B + \gamma C) = \beta(A \times B) + \gamma(A \times C), \quad \beta, \gamma \in \mathbb{R}, \tag{1.13}$$

4. *Commutatività della trasposta del prodotto (con ordine inverso):*

$$(A \times B)^T = B^T \times A^T. \tag{1.14}$$

Si assuma ora che le matrici $A \in \mathbb{R}^{m \times m}$ e $B \in \mathbb{R}^{n \times n}$ siano quadrate, allora si aggiungono le seguenti proprietà:

5. *Commutatività dell'inversa del prodotto (con ordine inverso) se A^{-1} e B^{-1} esistono:*

$$(A \times B)^{-1} = B^{-1} \times A^{-1}, \quad (1.15)$$

6. *La funzione caratteristica del prodotto semi-tensore tra due matrici è invariante rispetto all'ordine delle matrici, ovvero le matrici $(A \times B)$ e $(B \times A)$ hanno la stesso determinante, la stessa traccia e gli stessi autovalori ed autovettori. In particolare, dato $T = m.c.m(m, n)$, si ha che*

$$\det(A \times B) = (\det(A))^{T/m} (\det(B))^{T/n}. \quad (1.16)$$

Oltre alle proprietà fondamentali sopraelencate, è possibile semplificare l'espressione (1.7) di prodotto semitensoriale in particolari casi, come esposto nella seguente proposizione.

Proposizione 1.1.2. *Date due matrici $A \in \mathbb{R}^{r_1 \times c_1}$ e $B \in \mathbb{R}^{r_2 \times c_2}$:*

1. *se $A \succ_n B$, allora:*

$$A \times B = A(B \otimes I_n); \quad (1.17)$$

2. *se $A \prec_n B$, allora:*

$$A \times B = (A \otimes I_n)B; \quad (1.18)$$

3. *siano $X \in \mathbb{R}^p$ e $Y \in \mathbb{R}^q$ due vettori colonna, allora:*

$$X \times Y = (X \otimes Y). \quad (1.19)$$

È importante notare che, come per il prodotto convenzionale, nemmeno per il prodotto semi-tensore vale la proprietà commutativa, ovvero, in generale $A \times B \neq B \times A$. A differenza del prodotto usuale però, è possibile introdurre delle proprietà pseudo-commutative, ottenibili grazie all'utilizzo di particolari matrici di permutazione che saranno introdotte nella prossima sezione.

1.2 Matrici Logiche

Un ruolo fondamentale nell'analisi delle Boolean networks è ricoperto dalle matrici logiche. Nel seguito verrà introdotta la notazione utilizzata per definirle e verranno presentate alcune matrici logiche di particolare importanza, utili a ricavare la rappresentazione algebrica delle reti Booleane.

Definizione 1.4 (Matrice logica). Si consideri l'insieme $\mathcal{B} = \{0, 1\}$. Data la matrice identità I_n di dimensione n , la sua i -esima colonna verrà indicata con δ_n^i e rappresenta l' i -esimo vettore canonico di dimensione n . $\mathcal{L}_n \subset \mathcal{B}^n$ indica l'insieme di tutti i vettori canonici di dimensione n .

Una matrice $L \in \mathbb{R}^{r \times c}$ è definita matrice logica se le sue colonne, denotate con $\text{Col}(L)$, sono nella forma δ_r^i , ovvero se $\text{Col}(L) \subset \mathcal{L}_r$. L'insieme delle matrici logiche di dimensione $r \times c$ è indicato da $\mathcal{L}_{r \times c} \subset \mathcal{B}^{r \times c}$

Le matrici logiche di dimensione $r \times c$ possono essere rappresentate come $L = [\delta_r^{i_1} \ \delta_r^{i_2} \ \dots \ \delta_r^{i_c}]$ o nella forma $L = \delta_r[i_1, i_2, \dots, i_c]$ dove $i_1, i_2, \dots, i_c \in [1, r]$. I vettori canonici δ_n^i sono detti anche vettori logici. Vettori e matrici i cui elementi appartengono a \mathcal{B} vengono detti rispettivamente vettori Booleani e matrici Booleane.

Un'utile proprietà del prodotto semi-tensore applicato alle matrici logiche è esposta nella seguente proposizione.

Proposizione 1.2.1. *L'insieme \mathcal{L}_n è un insieme chiuso rispetto al prodotto semi-tensore \times , ovvero, il prodotto tra due matrici logiche $A \in \mathcal{L}_{r_1 \times c_1}$ e $B \in \mathcal{L}_{r_2 \times c_2}$ è ancora una matrice logica.*

Tra le matrici logiche si distinguono particolari matrici che permettono di maneggiare con molta libertà i vettori logici. Un esempio molto utile è dato da due matrici chiamate *front-maintaining operator* e *rear-maintaining operator*, le cui strutture e proprietà sono esposte nel seguente lemma.

Lemma 1.2.2. *Si definiscono front-maintaining operator e rear-maintaining operator rispettivamente le matrici:*

$$\begin{aligned} D_f^{p,q} &= I_p \otimes \mathbb{1}_q^T \\ &= [\underbrace{\delta_p^1 \dots \delta_p^1}_q \quad \underbrace{\delta_p^2 \dots \delta_p^2}_q \quad \dots \quad \underbrace{\delta_p^p \dots \delta_p^p}_q] \end{aligned} \quad (1.20)$$

$$\begin{aligned} D_r^{p,q} &= \mathbb{1}_p^T \otimes I_q \\ &= [\underbrace{\delta_q^1 \ \delta_q^2 \ \dots \ \delta_q^q}_p \quad \underbrace{\delta_q^1 \ \delta_q^2 \ \dots \ \delta_q^q}_p \quad \dots \quad \underbrace{\delta_q^1 \ \delta_q^2 \ \dots \ \delta_q^q}_p]. \end{aligned} \quad (1.21)$$

Dati due vettori logici, $X \in \mathcal{L}_p$ e $Y \in \mathcal{L}_q$, si ha che:

$$D_f^{p,q}XY = X \quad (1.22)$$

$$D_r^{p,q}XY = Y \quad (1.23)$$

Osservazione 2. Nel Lemma 1.2.2 le matrici $D_f^{p,q}$ e $D_r^{p,q}$ mantengono la stessa funzione anche nel caso in cui X ed Y siano due vettori probabilistici, ovvero vettori la cui somma delle componenti è unitaria.

L'utilità di queste due matrici sta nel poter far comparire un vettore logico alla sinistra o alla destra di un altro vettore logico, come illustrato nel seguente esempio.

Esempio 1.2.1. Dati quattro vettori Booleani $X_1, X_2, X_3 \in \mathcal{L}_2$ ed $Y \in \mathcal{L}_2$, si considerino le relazioni

$$X = X_1 \times X_2 \times X_3 \in \mathcal{L}_8 \quad (1.24)$$

$$Y = M X_2 \times X_3, \quad (1.25)$$

dove $M \in \mathcal{L}_{2 \times 4}$. Si desidera che (1.25) sia del tipo $Y = M' X$ per un'opportuna matrice M' . Considerando che $X_1 \in \mathcal{L}_2$ e $(X_2 \times X_3) \in \mathcal{L}_4$, la matrice da introdurre è $D_r^{2,4}$ ed in particolare si ha che

$$\begin{aligned} Y &= M X_2 \times X_3 \\ &= M D_r^{2,4} X_1 \times X_2 \times X_3 \\ &= M' X, \end{aligned} \quad (1.26)$$

dove $M' = M D_r^{2,4}$. △

Tra le matrici logiche un ruolo importante viene ricoperto dalle matrici di permutazione.

Data una permutazione

$$\sigma = \begin{pmatrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n \end{pmatrix}, \quad (1.27)$$

ad essa è associata un'unica matrice di permutazione Π_σ la cui struttura è

$$\Pi_\sigma = \begin{bmatrix} \delta_n^{i_1} & \delta_n^{i_2} & \dots & \delta_n^{i_n} \end{bmatrix}. \quad (1.28)$$

Π_σ trasforma una base (v_1, v_2, \dots, v_n) nella nuova base permutata

$$\left(v_1, v_2, \dots, v_n \right) = \left(v_{i_1}, v_{i_2}, \dots, v_{i_n} \right) \Pi_\sigma. \quad (1.29)$$

Ogni matrice di permutazione Π è dunque una matrice quadrata non singolare in $\mathcal{L}_{n \times n}$. Inoltre due matrici che differiscono per una similarità indotta da una matrice di permutazione vengono dette cogredienti, ovvero esse mantengono le stesse proprietà di non negatività. Le principali matrici non negative verranno analizzate nella Sezione 1.3.

Un particolare tipo di matrice di permutazione è rappresentato dalle matrici *swap*.

Definizione 1.5. Viene definita matrice *swap* e viene indicata con $W_{[n,m]} \in \mathcal{L}_{mn \times mn}$ una matrice di permutazione la cui forma è:

$$W_{[n,m]} := \delta_{mn}[1, m+1, 2m+1, \dots, (n-1)m+1, 2, m+2, 2m+2, \dots, (n-1)m+2, \dots, n, m+n, 2m+n, \dots, mn]. \quad (1.30)$$

L'utilizzo di matrici swap permette di ottenere delle proprietà pseudo-commutative nel prodotto semitensoriale. Nella seguente proposizione vengono riportate le due proprietà più significative per lo studio delle reti Booleane.

Proposizione 1.2.3. *Si considerino due vettori $X \in \mathbb{R}^m$ e $Y \in \mathbb{R}^n$ ed una matrice $A \in \mathbb{R}^{r \times c}$, il prodotto semi-tensore sinistro gode delle seguenti proprietà:*

1.

$$W_{[m,n]} \times X \times Y = Y \times X. \quad (1.31)$$

2.

$$X \times A = W_{[r,m]} \times A \times W_{[m,c]} \times X = (I_m \otimes A) \times X. \quad (1.32)$$

Dalla Proposizione 1.2.3 si evince che attraverso l'utilizzo di matrici swap adeguate, il prodotto semi-tensore permette di invertire agevolmente l'ordine dei fattori nel prodotto tra vettori colonna o nel prodotto tra vettori colonna e matrici.

Esempio 1.2.2. Si considerino le due seguenti espressioni

$$\begin{aligned} x &= x_1 \times x_2 \times x_3, \\ y &= M \times x_2 \times x_1, \end{aligned} \quad (1.33)$$

dove $x \in \mathcal{L}_8$, $x_1, x_2, x_3 \in \mathcal{L}_2$, $y \in \mathcal{L}_2$ ed $M \in \mathcal{L}_{2 \times 4}$. Si desidera che y sia espressa nella forma

$$y = M' \times x. \quad (1.34)$$

Innanzitutto, sfruttando la (1.22) si introduce x_3 utilizzando la matrice $D_f^{4,2}$ ottenendo

$$y = M \times D_f^{4,2} \times x_2 \times x_1 \times x_3. \quad (1.35)$$

A questo punto per scambiare l'ordine di x_1 ed x_2 viene introdotta la matrice swap $W_{[2,2]}$, riuscendo così ad ottenere l'equazione nella forma desiderata:

$$\begin{aligned} y &= M \times D_f^{4,2} \times W_{[2,2]} \times x_1 \times x_2 \times x_3 \\ &= M' \times x. \end{aligned} \quad (1.36)$$

Dove $M' = M \times D_f^{4,2} \times W_{[2,2]}$. △

1.3 Matrici non negative

Gli elementi presenti nei vettori logici e nelle matrici logiche appartengono all'insieme $\mathcal{B} = \{0, 1\}$; si ha dunque che una matrice $A = [a_{ij}] \in \mathcal{L}_{r \times c}$ è una matrice *non negativa*, ovvero $a_{ij} \geq 0 \forall i, j$.

Vedremo nel seguito che Boolean networks e Boolean control networks possono essere espressi come particolari sistemi positivi, ovvero sistemi dinamici nei quali le variabili di stato, ingresso ed uscita possono assumere soltanto valori non negativi. Naturalmente i sistemi lineari positivi a tempo discreto sono descritti da matrici non negative.

Risulta dunque utile richiamare brevemente la classificazione delle matrici non negative ed alcune loro proprietà.

Definizione 1.6. Data una matrice $M = [m_{ij}] \in \mathbb{R}^{r \times c}$ si ha che:

- se $m_{ij} \geq 0 \forall i, j$ (senza escludere il caso che M sia la matrice nulla) allora M viene definita matrice **non negativa** ed indicata con $M \geq 0$;
- se $m_{ij} \geq 0 \forall i, j$ ed almeno un elemento della matrice è positivo, allora M viene definita matrice **positiva** ed indicata con $M > 0$;
- se $m_{ij} > 0 \forall i, j$ allora M viene definita matrice **strettamente positiva** ed indicata con $M \gg 0$.

Naturalmente l'insieme delle matrici non negative comprende quello delle matrici positive e strettamente positive. Essendo i vettori delle matrici composta da una sola colonna o una sola riga, tale classificazione rimane invariata anche per i vettori.

Tra le matrici quadrate non negative è possibile distinguerne alcuni particolari tipi, ovvero le matrici primitive, irriducibili e riducibili.

Definizione 1.7. Una matrice non negativa $M \in \mathbb{R}^{k \times k}$ si dice:

- **primitiva** se esiste $s \in \mathbb{Z}$, $s > 0$, tale che $M^s \gg 0$;
- **irriducibile** se in corrispondenza ad ogni $i, j \in \{1, 2, \dots, k\}$ esiste un esponente $s \in \mathbb{Z}$ tale che $[M^s]_{i,j} > 0$, dove $[M^s]_{i,j}$ indica l'elemento in posizione (i, j) della matrice M^s ;
- **riducibile** se M non è irriducibile.

Le matrici quadrate strettamente positive sono primitive e le matrici primitive sono irriducibili.

Nel caso di matrici logiche è possibile verificare l'irriducibilità di una matrice $M \in \mathbb{R}^{k \times k}$ in maniera diretta: si consideri la somma Booleana

$$\mathbb{M} := \bigvee_{i=0}^{k-1} M^i, \quad (1.37)$$

M risulta irriducibile se \mathbb{M} ha tutti gli elementi unitari.

1.4 Il Prodotto di Khatri-Rao

Terminiamo questo capitolo introducendo il prodotto di Khatri-Rao.

Definizione 1.8. Date le matrici $A \in \mathbb{R}^{r_1 \times c}$ e $B \in \mathbb{R}^{r_2 \times c}$, il prodotto di Khatri-Rao tra A e B è definito come

$$A * B := [\text{Col}_1(A) \otimes \text{Col}_1(B), \text{Col}_2(A) \otimes \text{Col}_2(B), \dots, \text{Col}_c(A) \otimes \text{Col}_c(B)] \quad (1.38)$$

Si noti che $A * B \in \mathbb{R}^{r_1 r_2 \times c}$ e che tale prodotto è definito solo se le matrici coinvolte hanno lo stesso numero di colonne. Il Prodotto di Khatri-Rao gode della proprietà associativa e della proprietà distributiva, caratteristica che, come visto, accomuna tutti i prodotti tra matrici presentati.

Questo tipo di prodotto tra matrici, a differenza del prodotto semi-tensore, non risulta indispensabile nell'ottenimento della forma algebrica delle reti Booleane; tuttavia esso risulta uno strumento molto utile nello semplificarne la procedura, come verrà illustrato nel prossimo capitolo.

Osservazione 3. Nel prosieguo della tesi verrà omesso il simbolo \times per indicare il prodotto semitensoriale, in quanto ogni prodotto tra matrici sarà considerato tale; ciò è anche giustificato dal fatto che il prodotto usuale è soltanto un caso particolare del prodotto semitensoriale. Verranno invece sempre indicati dai simboli \otimes e $*$, rispettivamente i prodotti di Kronecker e Khatri-Rao.

Capitolo 2

Boolean Control Networks

Le reti Booleane sono sistemi dinamici in cui gli stati sono variabili logiche Booleane la cui evoluzione è determinata da funzioni logiche. Nel caso in cui non ci siano ingressi l'evoluzione è autonoma; le reti Booleane la cui evoluzione è autonoma vengono indicate con l'acronimo BNs, dall'inglese Boolean networks. Nel caso in cui nella rete siano presenti degli ingressi, anch'essi Booleani, le reti vengono chiamate reti logiche controllate e vengono indicate con l'acronimo BCNs, dall'inglese Boolean control networks.

Prima di poter definire in maniera rigorosa questo tipo di reti è utile richiamare alcuni concetti fondamentali della logica Booleana ed illustrare un metodo per poter trattare sistemi di equazioni logiche tramite rappresentazione algebrica.

2.1 Funzioni Booleane in forma algebrica

2.1.1 Logica Booleana

L'algebra Booleana si basa sull'interazione di variabili appartenenti a $\mathcal{B} = \{0, 1\}$ il cui risultato appartiene ancora a \mathcal{B} .

Definizione 2.1. Una variabile x i cui valori appartengono all'insieme \mathcal{B} viene definita variabile *Booleana* (o logica).

Un ruolo fondamentale nell'algebra Booleana è rappresentato dai connettori logici: essi vanno a definire le operazioni tra variabili logiche. Ad ogni operazione logica si può associare una tabella di verità, tabella nella quale ad ogni possibile combinazione delle variabili logiche viene associato il risultato dell'operazione. Date due variabili logiche $x, y \in \mathcal{B}$, i principali connettori logici sono tre:

- NOT: esprime l'operazione di negazione ed è indicato con il simbolo $\bar{\cdot}$; questo operatore logico ha sempre la precedenza sugli altri connettori. La sua tabella di verità è la seguente:

x	y	$x \rightarrow y$	$x \leftrightarrow y$
0	0	1	1
0	1	1	0
1	0	0	0
1	1	1	1

Tabella 2.1: Tabelle di verità degli operatori \rightarrow e \leftrightarrow .

x	\bar{x}
0	1
1	0

- OR: esprime l'operazione di disgiunzione ed è indicato con il simbolo \vee ; in particolare si ha che $x \vee y = 0$ se e solo se sia x che y sono pari a 0. La sua tabella di verità è la seguente:

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

- AND: esprime l'operazione di congiunzione ed è indicato con il simbolo \wedge ; in particolare si ha che $x \wedge y = 1$ se e solo se sia x che y assumono il valore 1. La sua tabella di verità è la seguente:

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Esistono molte altre operazioni logiche, tuttavia qualsiasi operazione diversa dalle tre appena elencate (NOT, OR, AND) può essere ottenuta mediante una combinazione di esse, come chiarito nel seguente esempio.

Esempio 2.1.1. Due connettori logici molto utilizzati sono l'implicazione logica e la doppia implicazione logica, indicati rispettivamente con \rightarrow e \leftrightarrow ; se ne riportano le tabelle di verità in Tabella 2.1. Entrambe possono essere riscritte in funzione

dei tre connettori principali $\bar{\cdot}$, \wedge , \vee ed in particolare si ha che:

$$x \rightarrow y \Leftrightarrow \bar{x} \vee y, \quad (2.1)$$

$$\begin{aligned} x \leftrightarrow y &\Leftrightarrow (x \rightarrow y) \wedge (y \rightarrow x) \\ &\Leftrightarrow (\bar{x} \vee y) \wedge (\bar{y} \vee x). \end{aligned} \quad (2.2)$$

△

Il connettore $\bar{\cdot}$ opera su di una sola variabile logica e per questo viene detto “unario” mentre i connettori \wedge e \vee che operano su due variabili vengono chiamati “binari”. Si consideri ora un connettore n -ario che opera su n variabili logiche: ciascuna variabile può assumere due valori, dunque un operatore n -ario è una mappa da un dominio di 2^n elementi ad una regione di due elementi.

Definizione 2.2. Una funzione logica nelle variabili $x_1, x_2, \dots, x_n \in \mathcal{B}$ è un’espressione logica che coinvolge x_1, x_2, \dots, x_n (ed eventualmente delle costanti, ovvero variabili che assumono un valore fisso, 0 o 1) mediante connettori logici. Dunque una funzione logica è una mappa $f : \mathcal{B}^n \rightarrow \mathcal{B}$. Essa è anche chiamata operazione n -aria.

Si deduce che anche gli operatori logici fondamentali sono delle funzioni logiche.

2.1.2 Le matrici di struttura

Per poter scrivere funzioni logica in forma algebrica, ovvero tramite il prodotto tra matrici e vettori, è necessario innanzitutto esprimere le variabili logiche in forma vettoriale.

Si consideri una variabile $\chi \in \mathcal{B}$, ad essa è associato un vettore $x \in \mathcal{L}_2$ tale che:

$$x = \begin{bmatrix} \chi \\ \bar{\chi} \end{bmatrix}, \quad (2.3)$$

ovvero

$$\chi = 1 \sim \begin{bmatrix} 1 \\ 0 \end{bmatrix} = x, \quad \chi = 0 \sim \begin{bmatrix} 0 \\ 1 \end{bmatrix} = x. \quad (2.4)$$

Il seguente risultato è di fondamentale importanza in quanto permette di rappresentare n variabili logiche in forma vettoriale tramite un unico vettore senza perdita di informazione.

Lemma 2.1.1. *Date $x_i \in \mathcal{L}_2$, $i = 1, 2, \dots, n$, variabili logiche in forma vettoriale, il vettore*

$$x = x_1 x_2 \cdots x_n = \times_{i=1}^n x_i \quad (2.5)$$

determina in maniera univoca gli $x_i \in \mathcal{L}_2$, $i = 1, 2, \dots, n$.

Dimostrazione. Il vettore x appartiene a \mathcal{L}_{2^n} . Si assuma che $x = \delta_{2^n}^i$ e lo si divida in due parti di uguale dimensione, ovvero

$$x = \begin{bmatrix} x' \\ x'' \end{bmatrix}. \quad (2.6)$$

Si ha che solo uno tra x' e x'' può essere un vettore logico appartenente a $\mathcal{L}_{2^{n-1}}$ mentre l'altro deve essere il vettore nullo di dimensione 2^{n-1} . Si supponga dapprima che $x' \in \mathcal{L}_{2^{n-1}}$ e $x'' = \mathbf{0}$. Risulta evidente dalla definizione di prodotto semi-tensore che deve risultare $x_1 = \delta_2^1$. Viceversa, nel caso in cui $x' = \mathbf{0}$ e $x'' \in \mathcal{L}_{2^{n-1}}$ risulta $x_1 = \delta_2^2$. Per calcolare x_2 si procede allo stesso modo, dividendo in due parti uguali quello che tra x' e x'' risulta essere il vettore logico. Il metodo può essere eseguito iterativamente fino al calcolo di x_n . \square

Come visto nel Capitolo 1, il prodotto semi-tensore tra vettori colonna risulta particolarmente agevole essendo equivalente al prodotto di Kronecker. In alternativa, dato $x = \times_{i=1}^n x_i = \delta_{2^n}^i$ è possibile calcolare l'indice i tramite l'espressione

$$i = \sum_{j=1}^n (1 - \chi_j) 2^{n-j} + 1, \quad (2.7)$$

dove χ_j rappresenta la variabile logica associata a x_j .

Definizione 2.3. Data una funzione logica $f(x_1, x_2, \dots, x_n)$, con $x_i \in \mathcal{L}_2$ variabile logica in forma vettoriale, f può essere espressa in forma algebrica tramite

$$f(x_1, x_2, \dots, x_n) = M_f x, \quad (2.8)$$

dove $x = \times_{i=1}^n x_i$ e M_f è una matrice di dimensione 2×2^n detta *matrice di struttura* di f .

Per ottenere le matrici di struttura di una qualsiasi funzione logica si può partire definendo le matrici di struttura dei tre operatori logici fondamentali, sfruttando il fatto che tutte le funzioni logiche possono essere riscritte tramite NOT, AND ed OR.

- La matrice di struttura dell'operatore NOT, indicata con M_N è

$$M_N = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \delta_2[2 \ 1]. \quad (2.9)$$

Data la variabile logica x si ha dunque che $\bar{x} = M_N x$, infatti

$$M_N \delta_2^1 = \delta_2^2 \quad e \quad M_N \delta_2^2 = \delta_2^1. \quad (2.10)$$

- La matrice di struttura dell'operatore OR, indicata con M_O , è

$$M_O = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \delta_2[1 \ 1 \ 1 \ 2]. \quad (2.11)$$

Date le variabili logiche x_1 e x_2 si ha dunque che $x_1 \vee x_2 = M_O x_1 x_2$, infatti

$$\begin{aligned} M_O \delta_2^1 \delta_2^1 &= \delta_2^1, & M_O \delta_2^1 \delta_2^2 &= \delta_2^1, \\ M_O \delta_2^2 \delta_2^1 &= \delta_2^1, & M_O \delta_2^2 \delta_2^2 &= \delta_2^2. \end{aligned} \quad (2.12)$$

- La matrice di struttura dell'operatore AND, indicata con M_A , è

$$M_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} = \delta_2[1 \ 2 \ 2 \ 2]. \quad (2.13)$$

Date le variabili logiche x_1 e x_2 si ha dunque che $x_1 \wedge x_2 = M_A x_1 x_2$, infatti

$$\begin{aligned} M_A \delta_2^1 \delta_2^1 &= \delta_2^1, & M_A \delta_2^1 \delta_2^2 &= \delta_2^2, \\ M_A \delta_2^2 \delta_2^1 &= \delta_2^2, & M_A \delta_2^2 \delta_2^2 &= \delta_2^2. \end{aligned} \quad (2.14)$$

Lemma 2.1.2. *Sia $f(x_1, x_2, \dots, x_n)$ una funzione logica nelle variabili logiche x_1, x_2, \dots, x_n , allora f può essere espressa come prodotto semitensoriale tra x_1, x_2, \dots, x_n e le matrici di struttura M_N, M_O e M_A .*

Dimostrazione. Discende direttamente dal fatto che ogni funzione logica può essere espressa tramite gli operatori logici $\bar{\cdot}, \vee$ e \wedge . \square

Esempio 2.1.2. Si consideri la funzione logica

$$f(x_1, x_2) = x_2 \rightarrow (x_1 \wedge x_2). \quad (2.15)$$

Essa può essere riscritta come

$$\begin{aligned} f(x_1, x_2) &= x_2 \rightarrow (x_1 \wedge x_2) \\ &= \bar{x}_2 \vee (x_1 \wedge x_2) \\ &= M_N x_2 \vee (x_1 \wedge x_2) \\ &= M_N M_O x_2 (x_1 \wedge x_2) \\ &= M_N M_O x_2 M_A x_1 x_2. \end{aligned} \quad (2.16)$$

\triangle

Per riuscire a portare nella forma (2.8) qualsiasi funzione lineare è utile introdurre un'ulteriore matrice logica, detta *matrice di riduzione di potenza* ed indicata con M_r .

Proposizione 2.1.3. *Data la matrice*

$$M_r = \delta_4[1 \ 4] \quad (2.17)$$

ed un vettore $x \in \mathcal{L}_2$, si ha che

$$x^2 = M_r x. \quad (2.18)$$

Dimostrazione. Sia $x = [t, 1 - t]^T$, allora

$$x^2 = x \otimes x = [t^2, t(1 - t), (1 - t)t, (1 - t)^2]. \quad (2.19)$$

Dato che $t \in \{0, 1\}$, si ha che $t(1 - t) = 0$, $t^2 = t$ e $(1 - t)^2 = 1 - t$, ottenendo

$$x^2 = [t, 0, 0, (1 - t)] = M_r x, \quad (2.20)$$

□

Sono ora disponibili tutti gli strumenti necessari a dimostrare l'esistenza e l'unicità delle matrici di struttura.

Teorema 2.1.4. *Data una funzione logica $f(x_1, x_2, \dots, x_n)$ nelle variabili logiche x_1, x_2, \dots, x_n , esiste un'unica matrice logica $M_f \in \mathcal{L}_{2 \times 2^n}$, detta matrice di struttura di f , tale che*

$$f(x_1, x_2, \dots, x_n) = M_f \times_{i=1}^n x_i. \quad (2.21)$$

Dimostrazione. Verrà prima dimostrata l'esistenza della matrice M_f e poi la sua unicità.

- **Esistenza:** usando il Lemma 2.1.2, è sufficiente provare che il prodotto semitensoriale tra le matrici di struttura degli operatori NOT, OR ed AND e le variabili x_1, x_2, \dots, x_n può essere portato nella forma (2.21). La procedura per il calcolo di M_f può essere divisa in tre parti:

1. Si spostano tutte le variabili a destra sfruttando l'equivalenza (vista nel Capitolo 1)

$$y M = (I_{2^k} \otimes M) y, \quad (2.22)$$

dove $y \in \mathcal{L}_{2^k}$ ed M è una matrice logica qualsiasi. A sinistra rimangono dunque le matrici logiche, eventualmente nella forma $(I_{2^k} \otimes M)$, $k = 1, 2, \dots$.

2. Si riordinano le variabili utilizzando matrici swap adeguate. Le matrici swap vanno eventualmente spostate a sinistra, sfruttando nuovamente la (2.22).

3. Le variabili, anche se ordinate, possono ripetersi. Si utilizzano dunque le matrici di riduzione di potenza per porre l'esponente di ciascuna variabile uguale a 1. Anche in questo caso le matrici inserite vanno spostate a sinistra.

Seguendo questa procedura viene creata la matrice di struttura (vedi Esempio 2.1.3).

- **Unicità:** si supponga (per assurdo) che esistano due matrici di struttura distinte $M_f \neq M'_f$ che descrivono tale relazione logica. Esse devono differire almeno per una colonna, per esempio la i -esima, ovvero $\text{Col}_i(M_f) \neq \text{Col}_i(M'_f)$. Si supponga che $\times_{i=1}^n x_i = \delta_{2^n}^i$, allora

$$f(x_1, x_2, \dots, x_n) = M_f \times_{i=1}^n x_i = M_f \delta_{2^n}^i = \text{Col}_i(M_f) \quad (2.23)$$

e

$$f(x_1, x_2, \dots, x_n) = M'_f \times_{i=1}^n x_i = M'_f \delta_{2^n}^i = \text{Col}_i(M'_f), \quad (2.24)$$

il ché porta ad una contraddizione.

Il fatto che M_f sia una matrice logica appartenente a $\mathcal{L}_{2 \times 2^r}$ discende direttamente dalle proprietà dell'insieme \mathcal{L}_k di essere chiuso rispetto al prodotto semi-tensore, infatti tutte le matrici che portano al calcolo di M_f sono anch'esse matrici logiche. \square

Si riporta un esempio per chiarire la procedura illustrata nella dimostrazione dell'esistenza di M_f .

Esempio 2.1.3. Con riferimento al precedente Esempio 2.1.2, si desidera ottenere la matrice di struttura di f .

Il primo passo è portare a destra tutte le variabili:

$$\begin{aligned} f(x_1, x_2) &= M_N M_O x_2 M_A x_1 x_2 \\ &= M_N M_O (I_2 \otimes M_A) x_2 x_1 x_2. \end{aligned} \quad (2.25)$$

Si procede riordinando le variabili tramite matrici swap:

$$\begin{aligned} f(x_1, x_2) &= M_N M_O (I_2 \otimes M_A) x_2 x_1 x_2 \\ &= M_N M_O (I_2 \otimes M_A) W_{[2,2]} x_1 x_2 x_2. \end{aligned} \quad (2.26)$$

Infine si conclude riducendo l'esponente di ciascuna variabile a 1:

$$\begin{aligned} f(x_1, x_2) &= M_N M_O (I_2 \otimes M_A) W_{[2,2]} x_1 x_2^2 \\ &= M_N M_O (I_2 \otimes M_A) W_{[2,2]} x_1 M_r x_2 \\ &= M_N M_O (I_2 \otimes M_A) W_{[2,2]} (I_2 \otimes M_r) x_1 x_2 \\ &= M_f x_1 x_2, \end{aligned} \quad (2.27)$$

Dove $M_f = M_N M_O (I_2 \otimes M_A) W_{[2,2]} (I_2 \otimes M_r)$ è la matrice struttura di f . \triangle

Osservazione 4. La procedura descritta per il calcolo delle matrici struttura non è unica: esiste un metodo basato sull'ispezione della tabella di verità della funzione f che permette anch'esso di ottenere la stessa matrice M_f . Tuttavia il metodo esposto nella dimostrazione del Teorema 2.1.4 descrive in modo particolarmente chiaro i passaggi algebrici necessari per ottenere il risultato.

2.1.3 Sistemi di equazioni logiche

Come si vedrà nel seguito, una rete Booleana composta da n nodi può essere espressa mediante un sistema di n equazioni logiche. In questa sezione verrà approfondito il caso generale in cui il sistema è composto da m funzioni e n variabili logiche.

Definizione 2.4. Si consideri il sistema

$$\begin{cases} f_1(\chi_1, \chi_2, \dots, \chi_n) = \beta_1 \\ f_2(\chi_1, \chi_2, \dots, \chi_n) = \beta_2 \\ \vdots \\ f_m(\chi_1, \chi_2, \dots, \chi_n) = \beta_m, \end{cases} \quad (2.28)$$

dove f_i , $i = 1, \dots, m$, sono funzioni logiche, χ_j , $j = 1, \dots, n$, sono variabili logiche e β_i , $i = 1, \dots, m$, sono costanti logiche. Un sistema di questo tipo viene definito *sistema di equazioni logiche*. Un insieme di costanti logiche c_i , $i = 1, \dots, n$, tali che

$$\chi_i = c_i, \quad i = 1, \dots, n, \quad (2.29)$$

soddisfa il sistema (2.28) è detto soluzione del sistema.

Esempio 2.1.4. Si consideri il seguente sistema di equazioni logiche:

$$\begin{cases} \chi_1 \wedge \chi_2 = \beta_1 \\ \chi_2 \vee \chi_3 = \beta_2. \end{cases} \quad (2.30)$$

Verifichiamo per ogni valore di β_1, β_2 quali sono le soluzioni del sistema.

1. $\beta_1 = 0, \beta_2 = 0$: le soluzioni del sistema sono due:

$$\begin{cases} \chi_1 = 1 \\ \chi_2 = 0 \\ \chi_3 = 0 \end{cases} \quad \text{e} \quad \begin{cases} \chi_1 = 0 \\ \chi_2 = 0 \\ \chi_3 = 0. \end{cases} \quad (2.31)$$

2. $\beta_1 = 0, \beta_2 = 1$: le soluzioni del sistema sono quattro:

$$\begin{cases} \chi_1 = 1 \\ \chi_2 = 0 \\ \chi_3 = 1 \end{cases}, \quad \begin{cases} \chi_1 = 0 \\ \chi_2 = 1 \\ \chi_3 = 1 \end{cases}, \quad \begin{cases} \chi_1 = 0 \\ \chi_2 = 1 \\ \chi_3 = 0 \end{cases} \quad \text{e} \quad \begin{cases} \chi_1 = 0 \\ \chi_2 = 0 \\ \chi_3 = 1. \end{cases} \quad (2.32)$$

3. $\beta_1 = 1, \beta_2 = 1$: le soluzioni del sistema sono due:

$$\left\{ \begin{array}{l} \chi_1 = 1 \\ \chi_2 = 1 \\ \chi_3 = 1 \end{array} \right. \quad \text{e} \quad \left\{ \begin{array}{l} \chi_1 = 1 \\ \chi_2 = 1 \\ \chi_3 = 0. \end{array} \right. \quad (2.33)$$

4. $\beta_1 = 1, \beta_2 = 0$: non ci sono soluzioni.

△

Ci si pone ora l'obiettivo di portare il sistema (2.28) in forma algebrica. Date le variabili $\chi_i, i = 1, \dots, n$, e le costanti logiche $\beta_j, j = 1, \dots, m$, la loro versione vettoriale verrà indicata rispettivamente con $x_i, i = 1, \dots, n$ e con $b_j, j = 1, \dots, m$. Sono ora presenti tutti gli strumenti necessari a convertire un sistema di equazioni logiche nella rispettiva equazione algebrica.

Teorema 2.1.5. *Dato il sistema di equazioni logiche*

$$\left\{ \begin{array}{l} f_1(\chi_1, \chi_2, \dots, \chi_n) = \beta_1 \\ f_2(\chi_1, \chi_2, \dots, \chi_n) = \beta_2 \\ \vdots \\ f_m(\chi_1, \chi_2, \dots, \chi_n) = \beta_m, \end{array} \right. \quad (2.34)$$

esso può sempre essere convertito nella forma algebrica

$$Lx = b, \quad (2.35)$$

dove $x = \times_{i=1}^n x_i$, $b = \times_{j=1}^m b_j$ ed L è una matrice logica di dimensione $2^m \times 2^n$, ovvero $L \in \mathcal{L}_{2^m \times 2^n}$.

Dimostrazione. Si consideri la forma vettoriale di ciascuna equazione in (2.34). Moltiplicando tra loro tutte le equazioni ed applicando la procedura per il calcolo delle matrici di struttura è sempre possibile ricondursi alla forma (2.35). □

Il risultato ottenuto è di grande importanza; si riporta un esempio in cui un sistema di equazioni logiche viene portato nella sua forma algebrica.

Esempio 2.1.5. Si riconsideri l'Esempio 2.1.4. Le equazioni del sistema in forma algebrica sono:

$$\left\{ \begin{array}{l} M_A x_1 x_2 = b_1 \\ M_O x_2 x_3 = b_2. \end{array} \right. \quad (2.36)$$

Moltiplicando tra loro le due equazioni si ottiene:

$$\begin{aligned}
M_A x_1 x_2 M_O x_2 x_3 &= M_A(I_4 \otimes M_O) x_1 x_2^2 x_3 \\
&= M_A(I_4 \otimes M_O) x_1 M_r x_2 x_3 \\
&= M_A(I_4 \otimes M_O)(I_2 \otimes M_r) x_1 x_2 x_3.
\end{aligned} \tag{2.37}$$

Definendo $L = M_A(I_4 \otimes M_O)(I_2 \otimes M_r)$, $x = x_1 x_2 x_3$ e $b = b_1 b_2$, la forma algebrica del sistema risulta

$$Lx = b, \tag{2.38}$$

in particolare $L = \delta_4[1, 1, 3, 4, 3, 3, 3, 4]$. \triangle

Il metodo esposto nella dimostrazione del Teorema 2.1.5 può risultare molto laborioso e coinvolgere matrici di dimensioni notevoli, basti pensare che nell'Esempio 2.1.5, che coinvolge solo 2 semplici equazioni, la matrice $(I_4 \otimes M_O)$ ha dimensione $2^3 \times 2^4$.

Un altro metodo, basato sul prodotto di Khatri-Rao, semplifica notevolmente i calcoli e permette di ridurre le dimensioni delle matrici che portano all'ottenimento di L .

Prima di esporlo, risulta fondamentale il seguente lemma.

Lemma 2.1.6. *Ogni equazione logica*

$$f_j(\chi_1, \chi_2, \dots, \chi_n) = \beta_j \tag{2.39}$$

del sistema (2.28) può essere espressa nella seguente forma vettoriale

$$M_j x = b_j, \tag{2.40}$$

dove $x = \times_{i=1}^n x_i$ e $M_j \in \mathcal{L}_{2 \times 2^n}$.

Dimostrazione. Nel caso in cui f_j sia funzione di tutte n le variabili allora M_j risulta essere la sua matrice di struttura, della quale è già stata provata l'esistenza. Nel caso in cui nell'espressione di f_j non compaiano k variabili $\chi_{i_1}, \dots, \chi_{i_k}$, si procede nel seguente modo:

1. Nella forma algebrica dell'equazione viene inserita la variabile x_{i_λ} , $i_\lambda \in \{i_1, \dots, i_k\}$, servendosi delle matrici $D_r^{2, 2^{n-k}}$ detta rear-maintaining operator, infatti (come visto nel Capitolo 1), vale l'uguaglianza

$$D_r^{2, 2^{n-k}} x_{i_\lambda} \times_{\neq i_1, \dots, i_k} x_i = \times_{\neq i_1, \dots, i_k} x_i. \tag{2.41}$$

La variabile x_{i_λ} viene detta *variabile fittizia*, in quanto non influenza l'espressione di f_j . Le altre $k-1$ variabili vengono inserite allo stesso modo, con la naturale modifica della matrice D_r in base al numero di variabili mancanti.

2. Una volta presenti tutte le n variabili si procede al calcolo della “nuova” matrice di struttura di f_j .

□

Osservazione 5. Per introdurre le variabili fittizie è possibile utilizzare anche il front-maintaining operator $D_f^{[2^{n-k}, 2]}$, considerando che la variabile inserita tramite la matrice D_f sarà posizionata alla destra delle $n - k$ variabili già presenti.

Esempio 2.1.6. Si riconsideri l’Esempio 2.1.4. La prima equazione del sistema può essere riscritta come

$$\begin{aligned} M_A x_1 x_2 &= M_A D_f[4, 2] x_1 x_2 x_3 \\ &= M_{f_1} x, \end{aligned} \quad (2.42)$$

mentre la seconda diventa

$$\begin{aligned} M_O x_2 x_3 &= M_O D_r[2, 4] x_1 x_2 x_3 \\ &= M_{f_2} x. \end{aligned} \quad (2.43)$$

△

È possibile ora introdurre il metodo che sfrutta il prodotto di Khatri-Rao per il calcolo della forma algebrica di un sistema di equazioni logiche.

Proposizione 2.1.7. *Si consideri il sistema (2.28) espresso tramite equazioni algebriche*

$$\begin{cases} M_{f_1} x = b_1 \\ M_{f_2} x = b_2 \\ \vdots \\ M_{f_m} x = b_m, \end{cases} \quad (2.44)$$

dove $x = \times_{i=1}^n x_i$. Dato $b = \times_{j=1}^m b_j$, il sistema può essere riscritto come

$$Lx = b, \quad (2.45)$$

dove la matrice $L \in \mathcal{L}_{2^m \times 2^n}$ è ottenibile tramite prodotto di Khatri-Rao delle matrici M_{f_i} , $i = 1, \dots, m$, ovvero:

$$\begin{aligned} L &= M_{f_1} * M_{f_2} * \dots * M_{f_m} \\ &= [Col_1(M_{f_1}) \times \dots \times Col_1(M_{f_m}), \dots, Col_{2^n}(M_{f_1}) \times \dots \times Col_{2^n}(M_{f_m})]. \end{aligned} \quad (2.46)$$

Esempio 2.1.7 (Continuazione Esempio 2.1.6). Si consideri il sistema di equazioni algebriche

$$\begin{cases} M_{f_1} x = b_1 \\ M_{f_2} x = b_2 \end{cases} \quad (2.47)$$

dove $M_{f_1} = \delta_2[1, 1, 2, 2, 2, 2, 2, 2]$ e $M_{f_2} = \delta_2[1, 1, 1, 2, 1, 1, 1, 2]$. Dato $b = b_1 b_2$, il sistema può essere riscritto come

$$L x = b, \quad (2.48)$$

dove $L = M_{f_1} * M_{f_2} = \delta_4[1, 1, 3, 3, 3, 3, 3, 4]$.

Come ci si aspettava, la matrice L coincide con quella calcolata nell'Esempio 2.1.5, tuttavia in questo caso la procedura è risultata più immediata e la dimensione delle matrici incontrate risulta inferiore. \triangle

Una volta ottenuta la forma $L x = b$, per un sistema di equazioni logiche risulta molto agevole trovarne le soluzioni o constatarne le non esistenza nel caso non ce ne fossero. Indicando con $\text{Col}(L)$ l'insieme di tutte le colonne di L , il sistema di equazioni logiche ammette soluzione se e solo se $b \in \text{Col}(L)$. Indicando con λ l'indice di una colonna tale che $b = \text{Col}_\lambda(L)$, si avrà che una soluzione è data da $x = \delta_{2^n}^\lambda$.

Il risultato ottenuto è riassunto dal seguente teorema.

Teorema 2.1.8. *Il sistema di equazioni algebriche (2.28), scritto nella forma algebrica $L x = b$, ha soluzione se e sole se $b \in \text{Col}(L)$. In particolare le soluzioni sono date da*

$$x = \delta_{2^n}^\lambda, \quad \lambda \in \Lambda, \quad (2.49)$$

dove $\Lambda = \{\lambda \mid \text{Col}_\lambda(L) = b\}$.

Si conclude questa sezione evidenziando la correttezza delle soluzioni ottenute mediante analisi della forma algebrica, confrontandole con quelle ottenute in modo empirico nell'Esempio 2.1.4.

Esempio 2.1.8. Si consideri il sistema di equazioni logiche dell'Esempio 2.1.4, la cui forma algebrica è

$$L x = b, \quad (2.50)$$

dove $x = x_1 x_2 x_3$, $b = b_1 b_2$ e $L = \delta_4[1, 1, 3, 3, 3, 3, 3, 4]$. Si determini la soluzione x per ogni valore di b .

1. $b = \delta_4^4$ ($\beta_1 = 0, \beta_2 = 0$): risulta che $b = \text{Col}_4(L) = \text{Col}_8(L)$, ovvero $\Lambda = \{4, 8\}$. Le soluzioni sono dunque

$$\begin{aligned} x = \delta_8^4 &\Rightarrow (\chi_1 = 1, \chi_2 = 0, \chi_3 = 0), \\ x = \delta_8^8 &\Rightarrow (\chi_1 = 0, \chi_2 = 0, \chi_3 = 0). \end{aligned} \quad (2.51)$$

2. $b = \delta_3^4$ ($\beta_1 = 0, \beta_2 = 1$): risulta che $b = \text{Col}_3(L) = \text{Col}_4(L) = \text{Col}_6(L) = \text{Col}_7(L)$, ovvero $\Lambda = \{3, 5, 6, 7\}$. Le soluzioni sono dunque

$$\begin{aligned} x = \delta_8^3 &\Rightarrow (\chi_1 = 1, \chi_2 = 0, \chi_3 = 1), \\ x = \delta_8^5 &\Rightarrow (\chi_1 = 0, \chi_2 = 1, \chi_3 = 1), \\ x = \delta_8^6 &\Rightarrow (\chi_1 = 0, \chi_2 = 1, \chi_3 = 0), \\ x = \delta_8^7 &\Rightarrow (\chi_1 = 0, \chi_2 = 0, \chi_3 = 1). \end{aligned} \quad (2.52)$$

3. $b = \delta_1^4$ ($\beta_1 = 1, \beta_2 = 1$): risulta che $b = \text{Col}_1(L) = \text{Col}_2(L)$, ovvero $\Lambda = \{1, 2\}$. Le soluzioni sono dunque

$$\begin{aligned} x = \delta_8^1 &\Rightarrow (\chi_1 = 1, \chi_2 = 1, \chi_3 = 1), \\ x = \delta_8^2 &\Rightarrow (\chi_1 = 1, \chi_2 = 1, \chi_3 = 0). \end{aligned} \quad (2.53)$$

4. $b = \delta_2^4$ ($\beta_1 = 1, \beta_2 = 0$): risulta che $b \notin \text{Col}(L)$, dunque il sistema non ha soluzioni.

△

2.2 Boolean Networks

Definizione 2.5. Una Boolean Network è composta da un insieme di variabili, $\chi_1, \chi_2, \dots, \chi_n$, che interagiscono tra loro. Ad ogni istante $t = 0, 1, 2, \dots$, ciascuna variabile, chiamata nodo della rete, può assumere solo i valori 1 o 0, ovvero i nodi sono variabili logiche. L'evoluzione della rete è determinata dal seguente sistema di equazioni logiche alle differenze del primo ordine

$$\begin{cases} \chi_1(t+1) = f_1(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \chi_2(t+1) = f_2(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \vdots \\ \chi_n(t+1) = f_n(\chi_1(t), \chi_2(t), \dots, \chi_n(t)), \end{cases} \quad (2.54)$$

dove le f_i , $i = 1, 2, \dots, n$, sono funzioni logiche.

Una Boolean network è dunque un sistema dinamico a tempo discreto che ha la peculiarità di essere descritto da variabili ed equazioni logiche.

Esempio 2.2.1. In sistema a tempo discreto descritto dal sistemi di equazioni

$$\begin{cases} \chi_1(t+1) = \overline{\chi_1(t)} \vee \chi_2(t) \\ \chi_2(t+1) = (\chi_1(t) \vee \overline{\chi_3(t)}) \wedge (\chi_2(t) \vee \overline{\chi_3(t)}) \\ \chi_3(t+1) = \overline{\chi_1(t)} \wedge \chi_3(t), \end{cases} \quad (2.55)$$

rappresenta una Boolean network composta da tre nodi.

△

Una rete Booleana è descritta da un sistema di equazioni logiche, dunque può essere portata in forma algebrica utilizzando i metodi visti nella Sezione 2.1.3. Esprimendo ogni nodo $\chi_i(t)$, $i = 1, 2, \dots, n$, con il rispettivo vettore logico $x_i(t) \in \mathcal{L}_2$, $i = 1, 2, \dots, n$, si ricava il vettore

$$x(t) = \times_{i=1}^n x_i(t) \quad (2.56)$$

che viene chiamato *vettore di stato*, in quanto esso identifica in maniera univoca lo stato della rete al tempo t . Naturalmente $x(t) \in \mathcal{L}_{2^n}$ e quindi la rete descritta dal sistema (2.54) può trovarsi in al più 2^n stati differenti, ovvero $x(t) = \delta_{2^n}^i$, $i = 1, 2, \dots, 2^n$.

La Boolean Network descritta da (2.54) può essere riscritta nella seguente forma algebrica componente per componente

$$\begin{cases} x_1(t+1) = M_{f_1} x(t) \\ x_2(t+1) = M_{f_2} x(t) \\ \vdots \\ x_n(t+1) = M_{f_n} x(t), \end{cases} \quad (2.57)$$

in cui $M_{f_i} \in \mathcal{L}_{2 \times 2^n}$, $i = 1, 2, \dots, n$. Utilizzando infine il prodotto di Khatri-Rao, è possibile ottenere la forma algebrica della rete Booleana (2.54), descritta dall'equazione

$$x(t+1) = L x(t), \quad (2.58)$$

in cui $L = M_{f_1} * M_{f_2} * \dots * M_{f_n}$, $L \in \mathcal{L}_{2^n \times 2^n}$. La matrice L è detta matrice di transizione della rete.

Per convincersi che la matrice L sia effettivamente una matrice logica si consideri il seguente lemma.

Lemma 2.2.1. *Data la Boolean Network (2.54), descritta dalla forma algebrica (2.58) si ha che, per ogni $i \in [1, 2^n]$,*

$$\text{Col}_i(L) \in \mathcal{L}_{2^n}. \quad (2.59)$$

Dimostrazione. Si deve dimostrare che $\text{Col}(L) \subset \mathcal{L}_{2^n}$. Si assuma (per assurdo) che esista un i , $i \in [1, 2^n]$, tale che $\text{Col}_i(L) \notin \mathcal{L}_{2^n}$. Allora dato lo stato $x(t) = \delta_{2^n}^i$ si ottiene

$$x(t+1) = L x(t) = \text{Col}_i(L). \quad (2.60)$$

Dunque $x(t+1) \notin \mathcal{L}_{2^n}$, che è una contraddizione, essendo ogni stato di una Boolean Network un vettore logico. \square

Esempio 2.2.2. Con riferimento all'Esempio 2.2.1, il vettore di stato è $x(t) = x_1(t) x_2(t) x_3(t)$ e la forma algebrica componente per componente della BN risulta

$$\begin{cases} x_1(t+1) = M_{f_1} x(t) \\ x_2(t+1) = M_{f_2} x(t) \\ x_3(t+1) = M_{f_3} x(t), \end{cases} \quad (2.61)$$

in cui le matrici $M_{f_i} \in \mathcal{L}_{2 \times 2^3}$, $i = 1, 2, 3$, sono:

$$\begin{aligned} M_{f_1} &= M_O M_N D_f^{4,2} \\ &= \delta_2[1, 1, 2, 2, 1, 1, 1, 1] \\ M_{f_2} &= M_A M_O (I_2 \otimes M_N) (I_4 \otimes M_O) (I_8 \otimes M_N) (I_2 \otimes W_{[2,2]}) (I_4 \otimes M_r) \\ &= \delta_2[1, 1, 2, 1, 2, 1, 2, 1] \\ M_{f_3} &= M_A M_N D_r^{2,4} W_{[2,2]} \\ &= \delta_2[2, 2, 2, 2, 1, 2, 1, 2]. \end{aligned}$$

Infine la forma algebrica della BN è descritta dall'equazione

$$x(t+1) = L x(t), \quad (2.62)$$

in cui $L = M_{f_1} * M_{f_2} * M_{f_3} = \delta_{2^3}[2, 2, 8, 6, 3, 2, 3, 2]$. \triangle

Teorema 2.2.2. *La dinamica della Boolean network (2.54) è completamente determinata dal sistema lineare a tempo discreto (2.58).*

Dimostrazione. Dal sistema (2.58) si ha che

$$\begin{aligned} x(1) &= L x(0) \\ x(2) &= L x(1) = L^2 x(0) \\ &\vdots \\ x(t-1) &= L^{t-1} x(0). \end{aligned} \quad (2.63)$$

Segue che

$$x_i(t) = M_{f_i} L^{t-1} x(0), \quad t = 1, 2, \dots, \quad i = 1, 2, \dots, n, \quad (2.64)$$

quindi (2.58) determina univocamente la dinamica della Boolean Network (2.54). \square

2.2.1 Rappresentazione tramite grafi

La rappresentazione più naturale delle Boolean networks è attraverso i grafi, la cui definizione viene richiamata brevemente.

Definizione 2.6. Dati un insieme $\mathcal{V} = \{V_i : i = 1, 2, \dots\}$ ed un insieme $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, $\Sigma = \{\mathcal{V}, \mathcal{E}\}$ è chiamato *grafo*, dove \mathcal{V} è detto insieme dei vertici ed \mathcal{E} è detto insieme degli archi. $(V_j, V_i) \in \mathcal{E}$ significa che esiste un arco con origine in V_j e termine in V_i . In particolare, se $(V_j, V_i) \in \mathcal{E}$ implica che $(V_i, V_j) \in \mathcal{E}$, il grafo viene definito *non orientato*, in caso contrario viene detto *orientato*.

Il numero di archi entranti in un nodo V_i viene detto grado entrante del nodo V_i , il numero di archi uscenti da V_i è definito grado uscente del nodo V_i .

Una BN può essere rappresentata principalmente tramite due tipi di grafi: il primo viene detto grafo di rete e mette in corrispondenza diretta i vertici di un grafo e i nodi della rete.

Definizione 2.7. Data la Boolean network descritta da (2.54), il grafo orientato il cui insieme dei vertici è composto dai nodi della Boolean Network, ovvero $\mathcal{V} = \{\chi_i : 1, 2, \dots, n\}$, viene definito *grafo di rete* della BN. In particolare l'arco $(\chi_i, \chi_j) \in \mathcal{E}$ se e solo se $\chi_j(t+1)$ è influenzato da $\chi_i(t)$, ovvero se $\chi_i(t)$ è argomento della funzione f_j e contribuisce a determinare il valore di $\chi_j(t+1)$.

Un grafo Σ composto da n vertici può essere rappresentato anche tramite una matrice Booleana di dimensione $n \times n$ detta matrice di incidenza ed indicata con $\mathcal{J}(\Sigma)$. La struttura di $\mathcal{J}(\Sigma)$ è definita come segue:

$$[\mathcal{J}(\Sigma)]_{i,j} = \begin{cases} 1, & \text{se } (V_j, V_i) \in \mathcal{E}, \\ 0, & \text{altrimenti.} \end{cases} \quad (2.65)$$

Matrice di incidenza e grafo di rete portano esattamente la stessa informazione, ovvero una rappresentazione è equivalente all'altra.

Esempio 2.2.3. Con riferimento all'Esempio 2.2.1, il grafo della rete $\Sigma = (\mathcal{V}, \mathcal{E})$ è rappresentato in Figura 2.1. In particolare risulta $\mathcal{V} = \{\chi_1, \chi_2, \chi_3\}$, $\mathcal{E} =$

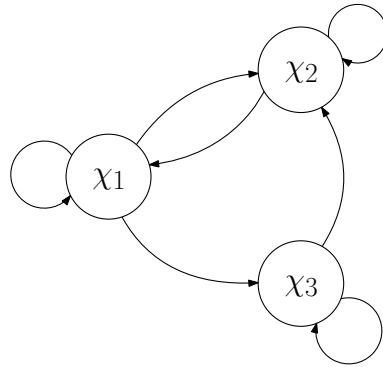


Figura 2.1: Grafo di rete della BN dell'Esempio 2.2.3.

$\{(\chi_1, \chi_1), (\chi_2, \chi_1), (\chi_2, \chi_2), (\chi_2, \chi_3), (\chi_3, \chi_1), (\chi_3, \chi_3)\}$, mentre la matrice di incidenza è

$$\mathcal{J}(\Sigma) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}. \quad (2.66)$$

△

La corrispondenza tra Boolean networks e grafi di rete non è biunivoca: ad ogni BN è associato un solo grafo, ma ad ogni grafo possono corrispondere più BNs. Infatti il grafo di rete non considera che tipo di operazioni logiche intercorrono tra i nodi, ma solo se un nodo ne influenza un altro.

Questa ambiguità può essere risolta tramite l'utilizzo dei grafi di stato.

Definizione 2.8. Si consideri una Boolean network descritta da (2.54) e la corrispondente forma algebrica (2.58). Si definisce *grafo di stato* associato alla BN il grafo orientato $\Lambda = (\mathcal{V}, \mathcal{E})$ in cui i vertici corrispondono alle 2^n configurazioni che lo stato può assumere, ovvero $\mathcal{V} = \{x_i = \delta_{2^n}^i : i = 1, 2, \dots, 2^n\}$. Ogni stato può transitare solo verso un unico altro stato (eventualmente se stesso), dunque il grado di uscita di ciascun vertice x_i , $i = 1, 2, \dots, 2^n$ è pari ad 1. Di conseguenza anche il numero di archi $(x_j, x_i) \in \mathcal{E}$ è pari a 2^n .

Questa rappresentazione garantisce che ad ogni grafo di stato possano corrispondere solo Boolean networks equivalenti tra loro, cioè reti la cui dinamica è esattamente la stessa. Naturalmente ad ogni BN è associato un solo grafo di stato. È comunque da notare che per una BN composta da n nodi, il grafo di rete è composto da n vertici, mentre in un grafo di stato i vertici sono 2^n , ovvero il numero di vertici cresce esponenzialmente all'aumentare dei nodi; come conseguenza, per reti di dimensioni considerevoli il grafo di stato perde la sua importante caratteristica di riuscire a comunicare in maniera immediata ed intuitiva la dinamica di una BN. Per determinare la configurazione di un grafo di stato risulta particolarmente utile l'analisi della matrice L che compare nella forma algebrica (2.58): un arco $(x_j, x_i) \in \mathcal{E}$ se e solo se $\text{Col}_j(L) = \delta_{2^n}^i$.

Esempio 2.2.4. Con riferimento all'Esempio 2.2.1, nell'Esempio 2.2.2 è stata ricavata la forma algebrica della BN. In particolare $L = \delta_{2^3}[2, 2, 8, 6, 3, 2, 3, 2]$. Se ne deduce che non c'è nessun arco che porta agli stati x_1, x_4, x_5, x_7 e l'insieme \mathcal{E} è composto dagli archi $(x_1, x_2), (x_2, x_2), (x_3, x_8), (x_4, x_6), (x_5, x_3), (x_6, x_2), (x_7, x_3)$ e (x_8, x_2) . l'insieme dei vertici è composto da tutti i 2^3 stati in cui può trovarsi la Boolean network, ovvero $\mathcal{V} = \{x_i = \delta_{2^3}^i : i = 1, 2, \dots, 2^3\}$. Il grafo di stato $\Lambda = (\mathcal{V}, \mathcal{E})$ associato alla Boolean network è rappresentato in Figura 2.2.

△

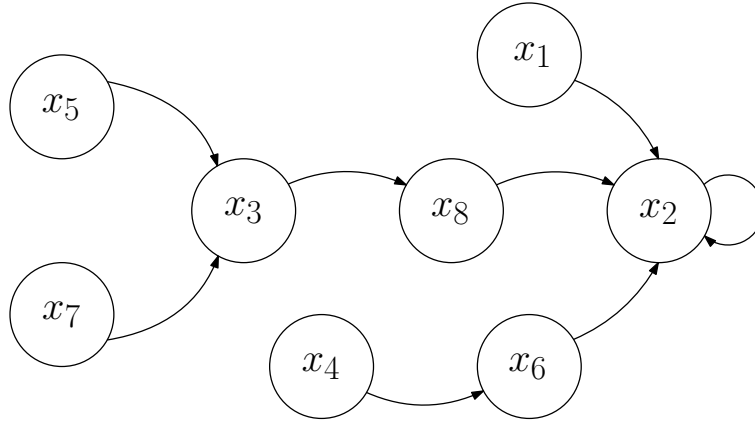


Figura 2.2: Grafo di stato della BN dell'Esempio 2.2.4.

2.2.2 Boolean networks seriali

Le Boolean networks definite da (2.54) sono chiamate Boolean networks in parallelo, in quanto l'aggiornamento dei nodi della rete avviene per l'appunto "in parallelo", simultaneamente.

Spesso avviene che l'aggiornamento dei nodi avvenga in maniera seriale, come descritto dal seguente sistema:

$$\begin{cases} \chi_1(t+1) = f_1(\chi_1(t), \chi_2(t), \dots, \chi_{n-1}(t), \chi_n(t)) \\ \chi_2(t+1) = f_2(\chi_1(t+1), \chi_2(t), \dots, \chi_{n-1}(t), \chi_n(t)) \\ \vdots \\ \chi_n(t+1) = f_n(\chi_1(t+1), \chi_2(t+1), \dots, \chi_{n-1}(t+1), \chi_n(t)), \end{cases} \quad (2.67)$$

Boolean networks descritte da sistemi nella forma (2.67) vengono chiamate Boolean networks *seriali*.

Tale tipo di rete Booleana trova applicazione, ad esempio, nella teoria dei giochi, come verrà approfondito nel Capitolo 4. Si riporta un semplice esempio di questo tipo di applicazione, con l'idea di cominciare ad evidenziare il legame che lega reti Booleane e teoria dei giochi.

Esempio 2.2.5. Si consideri un gioco composto da n giocatori, indicati con P_1, P_2, \dots, P_n . Ciascun giocatore può compiere solo due possibili azioni, che identifichiamo con $\mathcal{B} = \{0, 1\}$, mentre indichiamo con $\chi_i(t), i = 1, 2, \dots, n$, l'azione che il giocatore P_i ha eseguito al tempo t (in questo ambito si veda t come rappresentazione di un turno di gioco). Naturalmente l'azione $\chi_i(t+1)$ che il giocatore P_i eseguirà al tempo $t+1$ si baserà sulle azioni degli altri giocatori. In particolare ci sono due metodi in cui il gioco può svolgersi:

1. Tutti i giocatori giocano simultaneamente: in questo modo la mossa di ciascun giocatore al tempo $t + 1$ può basarsi solamente sulle mosse eseguite dai giocatori al tempo t . Risulta naturale modellare giochi di questo tipo tramite Boolean networks in parallelo.
2. Gioca un giocatore alla volta: in questo modo ciascun giocatore al tempo $t + 1$ può basare la sua mossa sulle mosse degli altri giocatori al tempo t e sulle mosse dei giocatori che al tempo $t + 1$ hanno giocato prima di lui. La scelta più adatta è modellare giochi di questo tipo tramite Boolean networks seriali.

△

Risulta comunque che Boolean networks in parallelo e seriali siano fortemente correlate le une con le altre, come esposto nella seguente proposizione.

Proposizione 2.2.3. *Data una Boolean network nella forma seriale descritta da (2.67) è sempre possibile ricondursi alla corrispondente Boolean network in parallelo descritta da (2.54).*

Dimostrazione. È sufficiente sostituire in (2.67) le variabili $\chi_i(t+1)$, $i = 1, 2, \dots, n-1$, con le rispettive equazioni di aggiornamento. □

Esempio 2.2.6. Si consideri la Boolean network in forma seriale descritta da

$$\begin{cases} \chi_1(t+1) = \chi_1(t) \vee \chi_2(t) \\ \chi_2(t+1) = \chi_1(t+1) \wedge \chi_3(t) \\ \chi_3(t+1) = \bar{\chi}_2(t+1). \end{cases} \quad (2.68)$$

Nell'espressione di $\chi_2(t+1)$ si sostituisce $\chi_1(t+1)$ con la prima equazione del sistema, ottenendo:

$$\begin{cases} \chi_1(t+1) = \chi_1(t) \vee \chi_2(t) \\ \chi_2(t+1) = (\chi_1(t) \vee \chi_2(t)) \wedge \chi_3(t) \\ \chi_3(t+1) = \bar{\chi}_2(t+1). \end{cases} \quad (2.69)$$

Allo stesso modo, sostituendo $\chi_2(t+1)$ nell'espressione di $\chi_3(t+1)$, si ottiene la Boolean network in parallelo. Nello specifico la dinamica dei nodi è descritta da:

$$\begin{cases} \chi_1(t+1) = \chi_1(t) \vee \chi_2(t) \\ \chi_2(t+1) = (\chi_1(t) \vee \chi_2(t)) \wedge \chi_3(t) \\ \chi_3(t+1) = (\bar{\chi}_1(t) \wedge \bar{\chi}_2(t)) \vee \bar{\chi}_3(t). \end{cases} \quad (2.70)$$

△

2.3 Boolean Control Networks

Inserendo delle variabili logiche controllabili dall'esterno in una Boolean network è possibile influire sulla dinamica della rete. La rete così ottenuta viene chiamata Boolean control network (BCN), mentre le variabili logiche introdotte vengono dette ingressi di controllo. Eventualmente è possibile considerare anche la presenza di uscite, le quali sono variabili logiche il cui valore è funzione statica dei nodi della rete.

Definizione 2.9. Una Boolean control network è un sistema dinamico a tempo discreto descritto dal sistema di equazioni

$$\begin{cases} \chi_1(t+1) = f_1(\chi_1(t), \chi_2(t), \dots, \chi_n(t), \mu_1(t), \mu_2(t), \dots, \mu_m(t)) \\ \chi_2(t+1) = f_2(\chi_1(t), \chi_2(t), \dots, \chi_n(t), \mu_1(t), \mu_2(t), \dots, \mu_m(t)) \\ \vdots \\ \chi_n(t+1) = f_n(\chi_1(t), \chi_2(t), \dots, \chi_n(t), \mu_1(t), \mu_2(t), \dots, \mu_m(t)), \end{cases} \quad (2.71)$$

ed eventualmente, nel caso siano presenti uscite, dal sistema

$$\begin{cases} \psi_1(t) = h_1(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \psi_2(t) = h_2(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \vdots \\ \psi_p(t) = h_p(\chi_1(t), \chi_2(t), \dots, \chi_n(t)). \end{cases} \quad (2.72)$$

Le funzioni $f_i : \mathcal{B}^{n+m} \rightarrow \mathcal{B}$, $i = 1, 2, \dots, n$, e $h_l : \mathcal{B}^n \rightarrow \mathcal{B}$, $l = 1, 2, \dots, p$, sono funzioni logiche; le n variabili logiche χ_i , $i = 1, 2, \dots, n$, sono i nodi della rete; le m variabili logiche μ_j , $j = 1, 2, \dots, m$, sono detti ingressi di controllo e le p variabili logiche ψ_i , $i = 1, 2, \dots, p$, sono le uscite della rete.

Come visto per le BNs, anche per le Boolean control networks è possibile ottenere la rappresentazione in forma algebrica.

Si consideri dapprima l'equivalente vettoriale delle variabili presenti in (2.71) e (2.72): in particolare agli ingressi di controllo μ_i , $i = 1, 2, \dots, m$, corrispondono i vettori logici $u_j \in \mathcal{L}_2$, $j = 1, 2, \dots, m$, alle uscite ψ_i , $i = 1, 2, \dots, p$, corrispondono i vettori logici $y_l \in \mathcal{L}_2$, $l = 1, 2, \dots, p$, ed ai nodi χ_i , $i = 1, 2, \dots, n$, corrispondono i vettori logici x_i , $i = 1, 2, \dots, n$.

Sfruttando il prodotto semitensoriale è poi possibile ottenere il vettore di stato $x(t) \in \mathcal{L}_{2^n}$, il vettore di ingresso $u(t) \in \mathcal{L}_{2^m}$ ed il vettore di uscita $y(t) \in \mathcal{L}_{2^p}$, definiti rispettivamente da:

$$x(t) = \times_{i=1}^n x_i(t), \quad u(t) = \times_{j=1}^m u_j(t), \quad y(t) = \times_{l=1}^p y_l(t). \quad (2.73)$$

È possibile quindi ottenere la forma algebrica componente per componente della BCN definita da (2.71) e (2.72):

$$\begin{cases} x_1(t+1) = M_{f_1} u(t) x(t) \\ x_2(t+1) = M_{f_2} u(t) x(t) \\ \vdots \\ x_n(t+1) = M_{f_n} u(t) x(t), \end{cases} \quad (2.74)$$

$$\begin{cases} y_1(t) = M_{h_1} x(t) \\ y_2(t) = M_{h_2} x(t) \\ \vdots \\ y_p(t) = M_{h_p} x(t), \end{cases}$$

in cui $M_{f_i} \in \mathcal{L}_{2 \times 2^{n+m}}$, $i = 1, 2, \dots, n$ e $M_{h_l} \in \mathcal{L}_{2 \times 2^n}$, $l = 1, 2, \dots, p$.

Infine, la forma algebrica di una Boolean control network è espressa da

$$\begin{aligned} x(t+1) &= L u(t) x(t) \\ y(t) &= H x(t), \end{aligned} \quad (2.75)$$

dove $H \in \mathcal{L}_{2^p \times 2^n}$, $H = M_{h_1} * \dots * M_{h_p}$ e $L \in \mathcal{L}_{2^n \times 2^{n+m}}$, $L = M_{f_1} * \dots * M_{f_n}$.

L è detta matrice di transizione della BCN.

Osservazione 6. Nell'equazione di aggiornamento dello stato (2.75) si è scelto di ordinare l'espressione mettendo prima il vettore di ingresso $u(t)$ e dopo il vettore dello stato $x(t)$. Questa scelta è del tutto arbitraria, in quanto invertendo l'ordine dei due fattori (ed aggiornando la matrice L di conseguenza), la dinamica descritta dall'equazione sarebbe la stessa.

Tuttavia l'ordine scelto in (2.75) permette un più immediato utilizzo degli strumenti forniti dalla Teoria dei Sistemi per lo studio di determinate proprietà delle BCNs.

Esempio 2.3.1. Si consideri la BN studiata negli Esempi 2.2.1 e 2.2.2 nella quale viene inserito un ingresso di controllo $\mu_1(t) \in \mathcal{B}$, ottenendo così la Boolean control network descritta dal seguente sistema

$$\begin{cases} \chi_1(t+1) = \overline{\chi_1}(t) \vee \chi_2(t) \\ \chi_2(t+1) = (\chi_1(t) \vee \overline{\chi_3}(t)) \wedge (\chi_2(t) \vee \overline{\chi_3}(t)) \\ \chi_3(t+1) = (\overline{\chi_1}(t) \wedge \chi_3(t)) \vee \mu_1(t). \end{cases} \quad (2.76)$$

Indicando con $u_1(t)$ la forma vettoriale di $\mu_1(t)$ e con $x_i(t)$ la forma vettoriale di $\chi_i(t)$, la forma algebrica componente per componente risulta essere

$$\begin{cases} x_1(t+1) = M_{f_1} u(t) x(t) \\ x_2(t+1) = M_{f_2} u(t) x(t) \\ x_3(t+1) = M_{f_3} u(t) x(t), \end{cases} \quad (2.77)$$

in cui

$$\begin{aligned} M_{f_1} &= \delta_2[1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1], \\ M_{f_2} &= \delta_2[1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1], \\ M_{f_3} &= \delta_2[1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2], \end{aligned} \quad (2.78)$$

$u(t) = u_1(t)$ e $x(t) = x_1(t) x_2(t) x_3(t)$. Infine la forma algebrica della BCN è

$$x(t+1) = L u(t) x(t), \quad (2.79)$$

in cui

$$L = \delta_8[1, 1, 7, 5, 3, 1, 3, 1, 2, 2, 8, 6, 3, 2, 3, 2]. \quad (2.80)$$

△

Si può notare che ad ogni diverso valore del vettore d'ingresso al tempo t , ovvero per ogni $u(t) = \delta_{2^m}^j$, $j = 1, \dots, 2^m$, l'equazione di aggiornamento dello stato in (2.75) può essere riscritta come

$$x(t+1) = L u(t) x(t) = L_j x(t), \quad (2.81)$$

dove $L_j \in \mathcal{L}_{2^n \times 2^n}$. La matrice L_j è facilmente deducibile dalla seguente suddivisione in 2^m blocchi della matrice L

$$L = [L_1 L_2 \cdots L_{2^m}]. \quad (2.82)$$

Di conseguenza si può considerare una Boolean control network come un Boolean switched system:

$$x(t+1) = L_{\sigma(t)} x(t), \quad (2.83)$$

in cui $\sigma(t)$, $t \in \mathbb{Z}_+$, è una sequenza di switch che assume valori in $[1, 2^m]$.

Riassumendo, ad ogni valore dell'ingresso $u(t) = \delta_{2^m}^j$ al tempo t è associata una Boolean network la cui matrice di transizione è L_j .

Esempio 2.3.2. Con riferimento all'Esempio 2.3.1, la BCN presenta un solo ingresso $\mu_1(t)$, dunque la matrice L può essere suddivisa in 2 blocchi

$$L = [L_1 L_2], \quad (2.84)$$

in cui

$$\begin{aligned} L_1 &= \delta_8[1, 1, 7, 5, 3, 1, 3, 1], \\ L_2 &= \delta_8[2, 2, 8, 6, 3, 2, 3, 2]. \end{aligned} \quad (2.85)$$

In particolare, quando $\mu_1(t) = 1 \rightarrow u(t) = \delta_2^1$ il sistema può essere rappresentato tramite $x(t+1) = L_1 x(t)$, mentre quando $\mu_1(t) = 0 \rightarrow u(t) = \delta_2^2$ il sistema è equivalente alla BN descritta da $x(t+1) = L_2 x(t)$.

Per verificare in maniera immediata la validità dei risultati ottenuti è facile constatare che la matrice L_2 è identica alla matrice di transizione della BN studiata nell'Esempio 2.2.2: l'unica differenza tra le 2 reti è infatti la presenza dell'ingresso $\mu_1(t)$ che va ad influire solo la dinamica del nodo χ_3 tramite l'operatore \vee e dunque, quando $\mu_1(t)$ assume il valore 0, è normale che la dinamica della BN e la dinamica della BCN siano la stessa. △

2.3.1 Rappresentazione tramite grafi

La rappresentazione tramite grafo di stato di una Boolean control network è molto simile a quella di una Boolean network, con la differenza che in questo caso, in base al valore dell'ingresso, è possibile transitare da uno stato verso uno o più stati e non più necessariamente verso uno soltanto.

Nello specifico l'insieme dei vertici è ancora composto dai 2^n possibili stati della rete, ovvero $\mathcal{V} = \{x_i = \delta_{2^n}^i, i = 1, \dots, 2^n\}$, mentre l'insieme degli archi è $\mathcal{E} = \bigcup_{j=1}^{2^m} \mathcal{E}_j$, dove \mathcal{E}_j rappresenta l'insieme degli archi associati alla Boolean network descritta da $x(t+1) = L_j x(t)$. Usando un'altra notazione un arco $(x_h, x_k) \in \mathcal{E}$ se e solo se $\text{Col}_h(L_j) = \delta_{2^n}^k$ per qualche $j \in [1, 2^m]$.

Esempio 2.3.3. Si consideri la Boolean control network introdotta nell'Esempio 2.3.1. Il grafo ad essa associato è rappresentato in Figura 2.3. In particolare si noti che partendo dagli stati $x_1, x_2, x_3, x_4, x_6, x_8$ è possibile transitare in due direzioni diverse a seconda che $u(t) = \delta_2^1$ o $u(t) = \delta_2^2$. L'insieme degli archi è $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$,

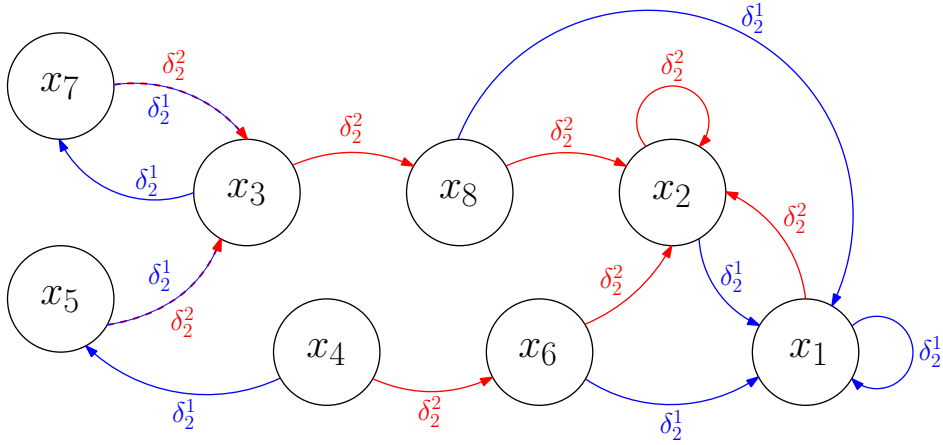


Figura 2.3: Grafo di stato della BCN dell'Esempio 2.3.3.

dove

$$\mathcal{E}_1 = \{(x_1, x_1), (x_2, x_1), (x_3, x_7), (x_4, x_5), (x_5, x_3), (x_6, x_1), (x_7, x_3), (x_8, x_1)\},$$

$$\mathcal{E}_2 = \{(x_1, x_2), (x_2, x_2), (x_3, x_8), (x_4, x_6), (x_5, x_3), (x_6, x_2), (x_7, x_3), (x_8, x_2)\}.$$

△

Per comprendere in modo immediato verso quali stati è possibile andare partendo da un determinato stato risulta molto utile l'analisi della seguente matrice:

$$L_{tot} = \bigvee_{j=1}^{2^m} L_j, \quad (2.86)$$

in cui \vee rappresenta la somma Booleana e L_{tot} è una matrice Booleana di dimensione $2^n \times 2^n$. La i -esima colonna di L_{tot} mostra tutti i nodi in cui si può arrivare in un passo partendo dallo stato x_i ; in particolare esiste un arco con origine in x_i che giunge in x_j se e solo se $[\text{Col}_i(L_{tot})]_j = 1$. Inoltre il numero degli elementi diversi da zero della i -esima colonna coincide con il grado uscente del vertice x_i .

Esempio 2.3.4. La matrice L_{tot} relativa alla Boolean control network analizzata nell'Esempio 2.3.3 è

$$L_{tot} = L_1 \vee L_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.87)$$

Dalla struttura di L_{tot} è possibile constatare immediatamente che sia dallo stato x_5 che dallo stato x_7 c'è un solo arco uscente e quindi la presenza di un ingresso non influenza la rete quando la BCN si trova in uno di questi due stati. \triangle

La matrice L_{tot} risulterà particolarmente utile nello studio di proprietà fondamentali delle Boolean control networks, quali raggiungibilità e controllabilità, come verrà studiato nella Sezione 3.4.

2.4 Ricostruzione

È stato dimostrato che la dinamica di una Boolean network costituita da n nodi è completamente determinata dal sistema lineare a tempo discreto

$$x(t+1) = Lx(t), \quad (2.88)$$

dove $x(t) \in \mathcal{L}_{2^n}$ è lo stato della rete al tempo t e $L \in \mathcal{L}_{2^n \times 2^n}$ è la matrice di transizione della rete. È stato altresì dimostrato come il valore di $x(t)$ determini in maniera univoca lo stato di ogni nodo χ_i , $i = 1, 2, \dots, n$, della rete al tempo t ed è stata esposta una procedura per risalire ai valori dei nodi partendo dal vettore di stato. A questo punto è lecito chiedersi come poter ricostruire la dinamica della Boolean network conoscendo solo la struttura della matrice L .

Si supponga dunque che la matrice L sia nota e si voglia ricavare l'espressione tramite sistema di equazioni logiche della Boolean network, ovvero si vuole ricavare

il sistema

$$\begin{cases} \chi_1(t+1) = f_1(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \chi_2(t+1) = f_2(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \vdots \\ \chi_n(t+1) = f_n(\chi_1(t), \chi_2(t), \dots, \chi_n(t)). \end{cases} \quad (2.89)$$

Anche in questo caso risulta essenziale il passaggio intermedio in cui ciascuna equazione in (2.89) viene espressa come equazione algebrica usando le matrici di struttura M_{f_i} , $i = 1, 2 \dots n$.

Dalla definizione di prodotto di Khatri-Rao, ciascuna colonna di L può essere espressa come

$$\text{Col}_j(L) = \text{Col}_j(M_{f_1}) \times \text{Col}_j(M_{f_2}) \times \dots \times \text{Col}_j(M_{f_n}), \quad j = 1, 2, \dots, 2^n. \quad (2.90)$$

I vettori $\text{Col}_j(M_{f_i})$, $i = 1, 2, \dots, n$, sono vettori logici di dimensione 2 e di conseguenza, per ricavarli partendo da $\text{Col}_j(L)$, è possibile applicare la stessa procedura utilizzata nella dimostrazione del Lemma 2.1.1. Applicando tale metodo a partire da ognuna delle 2^n colonne di L è dunque possibile ottenere le matrici di struttura M_{f_i} , $i = 1, 2 \dots n$.

La forma algebrica componente per componente della Boolean network è espressa da

$$\begin{cases} x_1(t+1) = M_{f_1} x(t) \\ x_2(t+1) = M_{f_2} x(t) \\ \vdots \\ x_n(t+1) = M_{f_n} x(t), \end{cases} \quad (2.91)$$

ma generalmente il nodo x_i , $i \in [1, n]$, non dipende da tutti i nodi della rete; basta ricordare il processo per ricavare le matrici di struttura, in cui le variabili assenti nell'equazione di aggiornamento di x_i venivano inserite mediante l'utilizzo delle matrici D_f e D_r . La seguente proposizione esprime un metodo per calcolare quali nodi non influenzano direttamente la dinamica del nodo x_i o, equivalentemente, quali nodi non posseggono un arco con termine in x_i .

Proposizione 2.4.1. *Si consideri il sistema (2.91). Se M_{f_i} soddisfa*

$$M_{f_i} W_{[2,2^{j-1}]} = M_{f_i} W_{[2,2^{j-1}]} M_N, \quad (2.92)$$

allora il nodo x_j non influenza direttamente la dinamica di x_i , ovvero l'arco (χ_j, χ_i) non esiste. Inoltre l'equazione di aggiornamento di x_i può essere sostituita da

$$x_i(t+1) = M'_{f_i} \times_{k \neq j} x_k(t), \quad (2.93)$$

in cui

$$M'_{f_i} = M_{f_i} W_{[2,2^{j-1}]} \delta_2^1 \quad \text{oppure} \quad M'_{f_i} = M_{f_i} W_{[2,2^{j-1}]} \delta_2^2. \quad (2.94)$$

Dimostrazione. La i -esima equazione di aggiornamento di una BN può essere riscritta scambiando l'ordine delle variabili, portando al primo posto il nodo x_j , ovvero

$$x_i(t+1) = M_{f_i} W_{[2,2^{j-1}]} x_j(t) \times_{k \neq j} x_k(t) \quad (2.95)$$

Se $x_i(t+1)$ rimane invariato sostituendo $x_j(t)$ con $\bar{x}_j(t)$, allora significa che il nodo x_i è indipendente dal nodo x_j . L'equazione (2.92) esprime semplicemente questa invarianza in forma algebrica.

Dato che $x_j(t)$ non influenza $x_i(t+1)$ è dunque possibile sostituire la variabile con un valore costante, che può essere sia $x_j(t) = \delta_2^1$ che $x_j(t) = \delta_2^2$. Da ciò discende direttamente la forma (2.94) di M'_{f_i} . \square

Esempio 2.4.1. Si consideri la Boolean network composta da 3 nodi descritta da

$$x(t+1) = L x(t), \quad (2.96)$$

dove $L = \delta_2[2, 2, 8, 6, 3, 2, 3, 2]$. Il calcolo delle tre matrici di struttura si ottiene direttamente dalle colonne di L , ottenendo:

$$\begin{aligned} M_{f_1} &= \delta_2[1, 1, 2, 2, 1, 1, 1, 1] \\ M_{f_2} &= \delta_2[1, 1, 2, 1, 2, 1, 2, 1] \\ M_{f_3} &= \delta_2[2, 2, 2, 2, 1, 2, 1, 2]. \end{aligned} \quad (2.97)$$

Per analizzare l'indipendenza tra i nodi viene usata la relazione (2.92), in particolare risulta:

$$\begin{aligned} M_{f_1} &= \delta_2[1, 1, 2, 2, 1, 1, 1, 1] \\ M_{f_1} M_N &= \delta_2[1, 1, 1, 1, 1, 1, 2, 2] \\ \\ M_{f_1} W_{[2,2]} &= \delta_2[1, 1, 1, 1, 2, 2, 1, 1] \\ M_{f_1} W_{[2,2]} M_N &= \delta_2[1, 1, 2, 2, 2, 2, 2, 2] \\ \\ M_{f_1} W_{[2,4]} &= \delta_2[1, 2, 1, 1, 1, 2, 1, 1] \\ M_{f_1} W_{[2,4]} M_N &= \delta_2[1, 2, 1, 1, 1, 2, 1, 1], \end{aligned} \quad (2.98)$$

per cui $x_1(t+1)$ non dipende da $x_3(t)$. Per il nodo x_2 si ha che:

$$\begin{aligned} M_{f_2} &= \delta_2[1, 1, 2, 1, 2, 1, 2, 1] \\ M_{f_2} M_N &= \delta_2[2, 1, 2, 1, 1, 1, 2, 1] \\ \\ M_{f_2} W_{[2,2]} &= \delta_2[1, 1, 2, 1, 2, 1, 2, 1] \\ M_{f_2} W_{[2,2]} M_N &= \delta_2[2, 1, 2, 1, 1, 1, 2, 1] \\ \\ M_{f_2} W_{[2,4]} &= \delta_2[1, 2, 2, 2, 1, 1, 1, 1] \\ M_{f_2} W_{[2,4]} M_N &= \delta_2[1, 1, 1, 1, 1, 2, 2, 2], \end{aligned} \quad (2.99)$$

quindi $x_2(t+1)$ dipende da tutti gli altri nodi. Per il nodo x_3 si ha che:

$$\begin{aligned}
M_{f_3} &= \delta_2[2, 2, 2, 2, 1, 2, 1, 2] \\
M_{f_3} M_N &= \delta_2[1, 2, 1, 2, 2, 2, 2, 2] \\
M_{f_3} W_{[2,2]} &= \delta_2[2, 2, 1, 2, 2, 2, 1, 2] \\
M_{f_3} W_{[2,2]} M_N &= \delta_2[2, 2, 1, 2, 2, 2, 1, 2] \\
M_{f_3} W_{[2,4]} &= \delta_2[2, 2, 1, 1, 2, 2, 2, 2] \\
M_{f_3} W_{[2,4]} M_N &= \delta_2[2, 2, 2, 2, 2, 2, 1, 1],
\end{aligned} \tag{2.100}$$

per cui $x_3(t+1)$ non dipende da $x_2(t)$.

Dopo aver eliminato le variabili ininfluenti dalle equazioni di aggiornamento di x_1 e x_3 , si deve aggiornare il valore delle matrici che ne descrivono la dinamica, ottenendo:

$$\begin{aligned}
M'_{f_1} &= M_{f_1} W_{[2,4]} \delta_2^1 = \delta_2[1, 2, 1, 1], \\
M'_{f_3} &= M_{f_3} W_{[2,2]} \delta_2^1 = \delta_2[2, 2, 1, 2].
\end{aligned} \tag{2.101}$$

La Boolean network può essere ora riscritta come:

$$\begin{cases} x_1(t+1) &= M'_{f_1} x_1(t) x_2(t) \\ x_2(t+1) &= M_{f_2} x_1(t) x_2(t) x_3(t) \\ x_3(t+1) &= M'_{f_3} x_1(t) x_3(t). \end{cases} \tag{2.102}$$

△

L'ultimo passaggio per ricavare la forma (2.89) di una Boolean network consta nell'associare ad ogni matrice M_{f_i} (o M'_{f_i}), $i = 1, 2, \dots, n$, la corrispondente funzione logica f_i .

Proposizione 2.4.2. *Si consideri la variabile logica β ed il corrispondente vettore logico $b \in \mathcal{L}_2$, le cui espressioni sono descritte da*

$$\begin{aligned}
\beta &= f(\chi_1, \chi_2, \dots, \chi_n) \\
&\updownarrow \\
b &= M_f x_1 x_2 \cdots x_n,
\end{aligned} \tag{2.103}$$

dove $f : \mathcal{B}^n \rightarrow \mathcal{B}$, $M_f \in \mathcal{L}_{2 \times 2^n}$ è la matrice di struttura associata ad f , gli argomenti $\chi_1, \chi_2, \dots, \chi_n$ sono variabili logiche e x_1, x_2, \dots, x_n sono i corrispondenti vettori logici. Allora

$$\beta = [\chi_1 \wedge h_1(\chi_2, \dots, \chi_n)] \vee [\bar{\chi}_1 \wedge h_2(\chi_2, \dots, \chi_n)], \tag{2.104}$$

dove le matrici di struttura associate ad h_1 e h_2 sono rispettivamente la prima e la seconda metà di M_f , ovvero

$$M_f = [M_f^{h_1} \mid M_f^{h_2}]. \tag{2.105}$$

Dimostrazione. Si consideri che la variabile logica χ_1 sia uguale ad 1 ($x_1 = \delta_2^1$), l'espressione algebrica di b diventa

$$b = M_f \delta_2^1 x_2 \cdots x_n = M_f^{h_1} x_2 \cdots x_n, \quad (2.106)$$

mentre quando $\chi_1 = 0$ ($x_1 = \delta_2^2$), l'espressione algebrica di b può essere riscritta come

$$b = M_f \delta_2^2 x_2 \cdots x_n = M_f^{h_2} x_2 \cdots x_n. \quad (2.107)$$

Dunque segue l'equazione (2.104). \square

Per ottenere l'espressione logica di β basta applicare ricorsivamente la procedura descritta nella Proposizione 2.4.2. Il procedimento può terminare quando le matrici $M_f^{h_i}$ sono di dimensione 2×4 e sono associate ad operatori conosciuti come l'operatore AND o l'operatore OR; nel caso le matrici di dimensione 2×4 non siano identificabili con dei connettori conosciuti, il procedimento termina quando la dimensione delle matrici $M_f^{h_i}$ è 2×2 , come illustrato dal seguente esempio.

Esempio 2.4.2. Con riferimento all'Esempio 2.4.1, la matrice M'_{f_1} si divide in 2 parti come segue

$$M'_{f_1} = [M_f^{h_1} \mid M_f^{h_2}] = [\delta_2[1, 2] \mid \delta_2[1, 1]], \quad (2.108)$$

utilizzando la (2.104), l'equazione di aggiornamento di χ_1 può essere riscritta come

$$\begin{aligned} \chi_1(t+1) &= [\chi_1(t) \wedge h_1(\chi_2(t))] \vee [\bar{\chi}_1 \wedge h_2(\chi_2(t))] \\ &\quad \updownarrow \\ x_1(t+1) &= M_O[M_A x_1(t)(M_f^{h_1} x_2(t))] [M_A M_N(x_1(t))(M_f^{h_2} x_2(t))] \\ &= M_O[M_A x_1(t) x_2(t)] [M_A M_N x_1(t) \delta_2^1] \\ &= M_O[M_A x_1(t) x_2(t)] [M_N x_1(t)] \\ &\quad \updownarrow \\ \chi_1(t+1) &= (\chi_1(t) \wedge \chi_2) \vee \bar{\chi}_1(t) \\ &= \bar{\chi}_1(t) \wedge \chi_2. \end{aligned} \quad (2.109)$$

Similmente è possibile ottenere le espressioni logiche legate a M_{f_2} e M'_{f_3} .

La Boolean network è infine descritta dal seguente sistema di equazioni logiche

$$\begin{cases} \chi_1(t+1) = \bar{\chi}_1(t) \vee \chi_2(t) \\ \chi_2(t+1) = (\chi_1(t) \vee \bar{\chi}_3(t)) \wedge (\chi_2(t) \vee \bar{\chi}_3(t)) \\ \chi_3(t+1) = \bar{\chi}_1(t) \wedge \chi_3(t). \end{cases} \quad (2.110)$$

\triangle

Capitolo 3

Proprietà delle reti Booleane

Nel Capitolo 2 è stata introdotta la definizione di Boolean network e si è ricavata la forma algebrica che ne descrive la dinamica:

$$x(t+1) = Lx(t) \quad (3.1)$$

dove $x(t)$ è detto vettore di stato e L matrice di transizione. Nel caso in cui la BN sia composta da n nodi allora $x(t) \in \mathcal{L}_{2^n}$ e $L \in \mathcal{L}_{2^n \times 2^n}$.

Quando in una Boolean network vengono inseriti degli ingressi si parla di Boolean control network. Nel caso nella rete siano presenti n nodi e su di essa agiscano m ingressi, la forma algebrica di una BCN è

$$x(t+1) = Lu(t)x(t), \quad (3.2)$$

in cui $x(t) \in \mathcal{L}_{2^n}$ è il vettore di stato, $u(t) \in \mathcal{L}_{2^m}$ è il vettore di ingresso ed $L \in \mathcal{L}_{2^n \times 2^{n+m}}$ è la matrice di transizione della rete. Una BCN può essere anche considerata come un Boolean switched system descritto dalla seguente espressione

$$x(t+1) = L_j x(t), \quad (3.3)$$

dove L_j indica il j -esimo blocco di L selezionato quando $u(t) = \delta_{2^m}^j$. La matrice L_{tot} è la somma Booleana di tutte le matrici L_j , ovvero:

$$L_{tot} = \bigvee_{j=1}^{2^m} L_j. \quad (3.4)$$

Nel corso di questo capitolo verranno studiate le proprietà di maggiore rilievo di Boolean networks e Boolean control networks: per farlo ci serviremo proprio delle matrici di transizione L per quanto riguarda le BNs, mentre risulterà particolarmente utile la matrice L_{tot} nello studio delle BCNs.

3.1 Punti fissi e cicli limite nelle BNs

Punti fissi e cicli limite rappresentano un aspetto molto significativo di una Boolean network, in quanto forniscono informazione sul comportamento asintotico della rete e risultano indispensabili nello studio di determinate proprietà come, ad esempio, la stabilità. L'individuazione di cicli limite e punti fissi risulta particolarmente intuitiva qualora si studi il grafo di stato associato ad una BN, ma è possibile ottenere tutte le informazioni necessarie anche dall'analisi della matrice di transizione L .

Cominciamo richiamando le definizioni di punto fisso e ciclo limite.

Definizione 3.1.

1. Uno stato x_e si dice punto fisso del sistema (3.1) se $Lx_e = x_e$.
2. L'insieme $\{x_e, Lx_e, \dots, L^{k-1}x_e\}$ si dice ciclo limite di lunghezza k del sistema (3.1) se $L^k x_e = x_e$ e gli elementi dell'insieme $\{x_e, Lx_e, \dots, L^{k-1}x_e\}$ sono tutti distinti.

La matrice L fornisce in modo immediato sia il numero dei punti fissi, sia quali stati sono punti fissi, come esposto dal seguente teorema.

Teorema 3.1.1. *Data una Boolean network descritta dal sistema (3.1), lo stato $x_e = \delta_{2^n}^i$ è un punto fisso del sistema se e solo se l'elemento in posizione (i, i) della matrice L è pari ad 1, ovvero se $[L]_{i,i} = 1$. Di conseguenza, indicando con N_e il numero di punti fissi della BN, risulta che:*

$$N_e = \text{tr}(L). \quad (3.5)$$

Dimostrazione. Dato lo stato $x_e = \delta_{2^n}^i$ risulta che $Lx_e = \text{Col}_i(L)$, dunque x_e è un punto fisso se e solo se

$$Lx_e = \text{Col}_i(L) = \delta_{2^n}^i, \quad (3.6)$$

ovvero se e solo se $[L]_{i,i} = 1$. □

I punti fissi possono essere considerati cicli di lunghezza unitaria. Il prossimo teorema espone un metodo induttivo per calcolare il numero di cicli di lunghezza $s \leq 2^n$, partendo proprio dalla conoscenza dei cicli di lunghezza unitaria.

Prima di esporre il teorema risulta utile richiamare la definizione di fattore proprio di un numero intero non negativo.

Definizione 3.2. Dati due numeri interi non negativi $a, b \in \mathbb{Z}_+$, b è detto *fattore proprio* di a se $a/b \in \mathbb{Z}_+$ e $b < a$.

L'insieme dei fattori proprio di un intero non negativo a verrà indicato con $\mathcal{P}(a)$. Ad esempio l'insieme dei fattori propri di 18 è $\mathcal{P}(18) = \{1, 2, 3, 6, 9\}$.

Teorema 3.1.2. *Data la Boolean network (3.1), il numero di cicli di lunghezza s è determinato ricorsivamente da:*

$$\begin{cases} N_1 = N_e = \text{tr}(L) \\ N_s = \frac{\text{tr}(L^s) - \sum_{k \in \mathcal{P}(s)} k N_k}{s}, \quad 2 \leq s \leq 2^n. \end{cases} \quad (3.7)$$

Dimostrazione. Dato uno stato $\delta_{2^n}^i$, se esso appartiene ad un ciclo di lunghezza s si ha che $L^s \delta_{2^n}^i = \delta_{2^n}^i$ ovvero $[L^s]_{i,i} = 1$, come dimostrato nel Teorema 3.1.1. Dunque $\text{Col}_i(L^s)$ contribuisce ad aumentare di una unità $\text{tr}(L^s)$.

D'altra parte dato uno stato $\delta_{2^n}^j$ appartenente ad un ciclo di lunghezza $k \in \mathcal{P}(s)$, si avrà anche che $L^s \delta_{2^n}^j = L^k \cdots L^k \delta_{2^n}^j = \delta_{2^n}^j$, ovvero $[L^s]_{j,j} = 1$ e di conseguenza $\text{Col}_j(L^s)$ contribuisce ad aumentare di uno $\text{tr}(L^s)$. Risulta quindi necessario sottrarre tutti gli elementi diagonali che contribuiscono a questo fenomeno da $\text{tr}(L^s)$. Infine dividendo il numero di stati appartenenti ai cicli di lunghezza s per la lunghezza dei cicli stessi, si ottiene il numero di cicli di lunghezza s , come avviene nella formula (3.7).

Il numero di stati appartenenti ad una BN con n nodi è 2^n e di conseguenza la lunghezza massima di un ciclo è 2^n . \square

Esempio 3.1.1. Si consideri una Boolean network composta da 3 nodi, descritta in forma algebrica da:

$$x(t+1) = L x(t), \quad (3.8)$$

dove $L = \delta_8[1, 3, 7, 5, 6, 5, 2, 1]$.

Il calcolo dei punti fissi risulta immediato, infatti si ha che $\text{tr}(L) = 1$ ed in particolare $[L^s]_{1,1} = 1$, ovvero x_1 è l'unico punto fisso ($N_e = N_1 = 1$) della BN.

Per ottenere il numero dei cicli di lunghezza $s \leq 8$, si riportano innanzitutto le potenze fino all'ottava della matrice L , dato che il ciclo di lunghezza maggiore potrebbe avere lunghezza pari ad 8:

$$\begin{aligned} L^2 &= \delta_8[1, 7, 2, 6, 5, 6, 3, 1] \\ L^3 &= \delta_8[1, 2, 3, 5, 6, 5, 7, 1] \\ L^4 &= \delta_8[1, 3, 7, 6, 5, 6, 2, 1] \\ L^5 &= \delta_8[1, 7, 2, 5, 6, 5, 3, 1] \\ L^6 &= \delta_8[1, 2, 3, 6, 5, 6, 7, 1] \\ L^7 &= \delta_8[1, 3, 7, 5, 6, 5, 2, 1] \\ L^8 &= \delta_8[1, 7, 2, 6, 5, 6, 3, 1] \end{aligned} \quad (3.9)$$

Dall'Equazione (3.7) si ottiene che il numero di cicli di lunghezza 2 è:

$$N_2 = \frac{\text{tr}(L^2) - \sum_{k \in \mathcal{P}(2)} k N_k}{2} = \frac{3 - (1 \cdot 1)}{2} = 1. \quad (3.10)$$

In modo analogo il numero di cicli di lunghezza 3 è pari a:

$$N_3 = \frac{\text{tr}(L^3) - \sum_{k \in \mathcal{P}(3)} k N_k}{3} = \frac{4 - (1 \cdot 1)}{3} = 1. \quad (3.11)$$

Il numero di cicli di lunghezza 4 è:

$$N_4 = \frac{\text{tr}(L^4) - \sum_{k \in \mathcal{P}(4)} k N_k}{4} = \frac{3 - (1 \cdot 1 + 2 \cdot 1)}{4} = 0. \quad (3.12)$$

Il fatto che non esistano cicli di lunghezza 4 era già deducibile dal fatto che gli unici stati non appartenenti a nessun ciclo sono x_4 ed x_8 , non sufficienti a comporre un ciclo di lunghezza 4. Per lo stesso motivo risulta inutile la ricerca di cicli di lunghezza superiore. \triangle

Si vuole ora ottenere una procedura rigorosa che permetta di identificare gli stati presenti in un ciclo di lunghezza s .

Si considerino le potenze s di L per cui

$$\text{tr}(L^s) - \sum_{k \in \mathcal{P}(s)} k N_k > 0, \quad (3.13)$$

tali potenze vengono definite *potenze non banali*.

Considerando le potenze non banali di L si definiscono gli insiemi

$$C_s = \{i \mid [L^s]_{i,i} = 1\}, \quad s = 1, 2, \dots, 2^n \quad (3.14)$$

e

$$D_s = C_s \cap \bigcap_{i \in \mathcal{P}(s)} C_i^c, \quad (3.15)$$

in cui C_i^c identifica l'insieme complementare di C_i .

La seguente proposizione risulta ovvia.

Proposizione 3.1.3. *Dato $x_i = \delta_{2^n}^i$, allora $\{x_i, L x_i, \dots, L^s x_i\}$ è un ciclo di lunghezza s se e solo se $i \in D_s$.*

Esempio 3.1.2. Riprendendo l'Esempio 3.1.1, gli insiemi C_1 e D_1 coincidono ($C_1 = D_1 = \{1\}$) e il punto fisso risulta essere x_1 come già calcolato.

Per quanto riguarda gli stati appartenenti al ciclo di lunghezza 2 si ottiene che

$$C_2 = \{i \mid [L^2]_{i,i} = 1\} = \{1, 5, 6\} \quad (3.16)$$

e

$$D_2 = C_2 \cap C_1^c = \{1, 5, 6\} \cap \{2, 3, 4, 5, 6, 7, 8\} = \{5, 6\}, \quad (3.17)$$

dunque gli stati appartenenti al ciclo di lunghezza 2 sono x_5 e x_6 .

Allo stesso modo per il ciclo di lunghezza 3 si ha che

$$C_3 = \{i | [L^3]_{i,i} = 1\} = \{1, 2, 3, 7\} \quad (3.18)$$

e

$$D_3 = C_3 \cap C_1^c = \{1, 2, 3, 7\} \cap \{2, 3, 4, 5, 6, 7, 8\} = \{2, 3, 7\}. \quad (3.19)$$

Dunque gli stati appartenenti al ciclo di lunghezza 3 sono x_2 , x_3 e x_7 .

In Figura 3.1 viene riportato il grafo di stato della BN, nel quale sono evidenziati il punto fisso ed i due cicli di lunghezza 2 e 3. \triangle

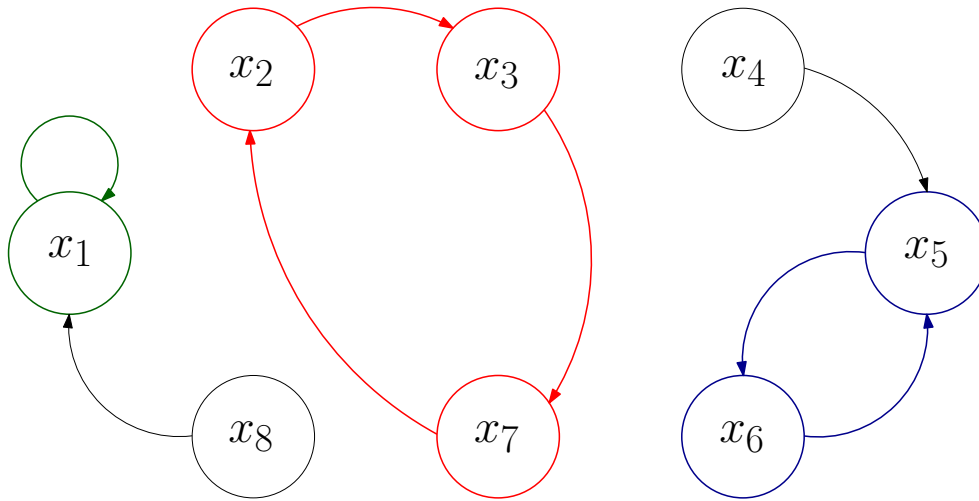


Figura 3.1: Grafo di stato della BN dell'Esempio 3.1.1: sono evidenziati in blu e rosso rispettivamente i cicli di lunghezza 2 e 3, mentre in verde l'unico punto fisso.

3.1.1 Analisi temporale

Punti fissi e cicli limite vengono detti attrattori, in quanto gli altri stati della BN convergono verso di essi in un numero di passi finito, come verrà dimostrato in seguito. Dato uno stato iniziale risulta quindi interessante capire quale sarà il ciclo a cui convergerà e dopo quanti istanti si può affermare che la BN è entrata in un ciclo.

Osservazione 7. Nel proseguo ci si riferirà solo a cicli limite evitando di specificare il caso particolare dei punti fissi, in quanto considerabili cicli di lunghezza unitaria.

Indicando con \mathcal{C}_i , $i = 1, \dots, q$, l'insieme degli stati che appartengono all' i -esimo ciclo, si definisce insieme degli attrattori (o insieme limite) l'insieme $\Omega = \bigcup_{i=1}^q \mathcal{C}_i$.

Data la natura deterministica delle BNs, si ha che partendo da ogni possibile stato $x(0) = x_0 = \delta_{2^n}^i$, $i = 1, 2, \dots, 2^n$, dopo un numero finito di passi $T_t(x_0)$ si ottiene che $x(T_t(x_0)) \in \Omega$. $T_t(x_0)$ è detto tempo di transizione (o assorbimento) dello stato x_0 e coincide con il più piccolo k tale che $x(0) = x_0$ e $x(k) \in \Omega$.

Definizione 3.3. Il tempo di transizione di una Boolean network, indicato con T_t , è definito come:

$$T_t = \max_{x \in \mathcal{L}_{2^n}} (T_t(x)). \quad (3.20)$$

Partendo da un qualsiasi stato della BN si ha dunque che dopo T_t istanti la rete si trova sicuramente in un ciclo.

Essendo L una matrice logica, essa può assumere solamente $r := 2^n \times 2^n$ valori distinti; di conseguenza nella sequenza di $r + 1$ elementi

$$I_{2^n}, L, L^2, \dots, L^r, \quad (3.21)$$

devono necessariamente esserci almeno 2 matrici uguali. Sia r_0 il più piccolo intero i per cui L^i appare nuovamente nella sequenza, ovvero $L^i = L^k$ per qualche $k > i$. Si ha che $r_0 < r$ e in particolare

$$r_0 = \operatorname{argmin}_{0 \leq i < r} \{L^i \in \{L^{i+1}, \dots, L^r\}\}. \quad (3.22)$$

Data l'esistenza di r_0 , vale la seguente proposizione.

Proposizione 3.1.4. *Sia r_0 il valore definito da (3.22). Partendo da un qualsiasi stato iniziale $x(0)$, la traiettoria della BN si trova sicuramente in un ciclo dopo r_0 iterazioni.*

Dimostrazione. Dalla definizione di r_0 si ha che $\exists k$ tale che $L^{r_0} = L^{r_0+k}$. Partendo da un qualsiasi stato iniziale $x(0)$ si ottiene che

$$x(r_0) = L^{r_0} x(0) = L^{r_0+k} x(0), \quad (3.23)$$

dunque $x(r_0) = x(r_0 + k)$ e $x(r_0)$ appartiene evidentemente ad un ciclo. \square

Per come sono stati definiti T_t e r_0 , essi sono sicuramente legati l'uno all'altro. Il seguente lemma sarà d'aiuto per identificarne il legame.

Lemma 3.1.5. *Sia τ il più piccolo intero positivo tale che $L^{r_0} = L^{r_0+\tau}$. Se nella Boolean network è presente un ciclo di lunghezza t , allora t è un fattore di τ .*

Dimostrazione. Si noti innanzitutto che, per come sono stati definiti r_0 e τ , si ha che $\tau + r_0 \leq r$. Si supponga per assurdo che la divisione tra τ e t dia resto s diverso da zero, ovvero che $\tau = tk + s$, $1 \leq s < t$; dato uno stato x_i appartenente al ciclo di lunghezza t si ha che anche lo stato $L^{r_0}x_i$ vi appartiene e dalla definizione di τ anche $L^{r_0+\tau}x_i$ appartiene al medesimo ciclo. Sviluppando i calcoli si ottiene:

$$L^{r_0+\tau}x_i = L^{r_0}L^\tau x_i = L^{r_0}L^s L^{tk}x_i = L^{r_0}L^s x_i = L^{r_0+s}x_i. \quad (3.24)$$

Si ha evidentemente che $L^{r_0+s}x_i \neq L^{r_0}x_i$ mentre $L^{r_0+\tau}x_i = L^{r_0}x_i$, il che porta alla contraddizione. \square

Confrontando la definizione di r_0 e T_t si può già constatare che $T_t \leq r_0$. Supponendo che $T_t < r_0$, dalla definizione di tempo di assorbimento si ha che per ogni stato x_i , $i = 1, 2, \dots, 2^n$, lo stato $L^{T_t}x_i$ appartiene ad un ciclo la cui lunghezza è un divisore di τ , dunque:

$$L^\tau x_i = L^\tau L^{T_t}x_i = L^{T_t+\tau}x_i, \quad (3.25)$$

ma dalla definizione di r_0 il fatto che $L^\tau = L^{T_t+\tau}$ è in contraddizione con la supposizione che $T_t < r_0$. Di conseguenza deve essere $T_t = r_0$. Il risultato appena ottenuto funge da dimostrazione al seguente teorema.

Teorema 3.1.6. *In una Boolean network tempo di transizione e r_0 coincidono, ovvero:*

$$r_0 = T_t. \quad (3.26)$$

Esempio 3.1.3. Con riferimento all'Esempio 3.1.1 si ha che

$$L^1 = L^7, \quad (3.27)$$

dunque

$$r_0 = \operatorname{argmin}_{0 \leq i < 64} \{L^i \in \{L^{i+1}, \dots, L^{64}\}\} = 1. \quad (3.28)$$

Il tempo di transizione T_t è quindi pari ad 1, infatti analizzando il grafo della BN in Figura 3.1 è immediato notare che partendo da un qualsiasi stato è sufficiente al massimo un passo per entrare in un ciclo. Inoltre si ha che $\tau = 7 - 1 = 6$ e grazie al Lemma 3.1.5 si può affermare che i cicli presenti nella BN possono avere lunghezza pari ai fattori di 6, ovvero 1, 2, 3 e 6 (dai risultati ottenuti precedentemente sappiamo che i cicli hanno lunghezza 1, 2 e 3). \triangle

3.2 Divisione per classi

Nella sezione precedente si è visto come gli stati di una Boolean network possano essere divisi in attrattori e non; in realtà questa divisione è soltanto una delle molte possibili, infatti gli stati delle BNs si prestano in maniera molto naturale ad una suddivisione per classi: ciascuna classificazione degli stati in particolari insiemi risulta molto utile nell'analisi di diversi aspetti delle reti Booleane. In questa sezione verranno presentate diverse tecniche di suddivisione degli stati e verranno analizzate le informazioni ricavabili da tali rappresentazioni.

3.2.1 Attrattori e bacini di attrazione

L'insieme limite Ω è esprimibile come unione degli stati appartenenti ai cicli presenti nel sistema, ovvero $\Omega = \bigcup_{i=1}^p \mathcal{C}_i$, dove \mathcal{C}_i rappresenta l'insieme degli stati appartenenti all' i -esimo ciclo e p è il numero di cicli presenti.

Una suddivisione degli stati molto utile è realizzata creando degli insiemi, detti bacini di attrazione, contenenti i soli stati che portano ad un determinato ciclo. La seguente definizione chiarisce in modo formale la composizione di tali insiemi.

Definizione 3.4. Viene definito *bacino di attrazione* di \mathcal{C}_i , l'insieme \mathcal{S}_i contenente tutti gli stati che convergono a \mathcal{C}_i . Uno stato x_i appartiene a \mathcal{S}_i se e solo se

$$x(t) = L^t x_i \in \mathcal{C}_i, \quad \forall t \geq T_t. \quad (3.29)$$

Evidentemente il bacino di attrazione di \mathcal{C}_i è composto da \mathcal{C}_i stesso più gli stati che portano in un certo numero di passi (sicuramente minore o uguale al tempo di attrazione T_t) a \mathcal{C}_i .

Per determinare quali sono gli stati che entrano in un ciclo risulta utile il concetto di stato genitore.

Definizione 3.5. Dati due stati $v, w \in \mathcal{L}_{2^n}$, w è detto stato genitore di v se $v = Lw$.

L'insieme di tutti gli stati genitore di v è indicato con $L^{-1}(v)$.

La definizione di insieme di stati genitore può essere estesa agli insiemi $\mathcal{C} \in \mathcal{L}_{2^n}$ come segue:

$$L^{-1}(\mathcal{C}) = \{w \mid Lw \in \mathcal{C}\}. \quad (3.30)$$

Infine con $L^{-k}(v)$ vengono indicati gli stati genitore in k passi dello stato v .

Proposizione 3.2.1. Dato l'insieme \mathcal{C}_i contenente gli stati appartenenti all' i -esimo ciclo di una Boolean network, il suo bacino di attrazione è

$$\mathcal{S}_i = \mathcal{C}_i \cup L^{-1}(\mathcal{C}_i) \cup L^{-2}(\mathcal{C}_i) \cup \dots \cup L^{-T_t}(\mathcal{C}_i). \quad (3.31)$$

Per calcolare l'insieme degli stati genitore di uno stato v è sufficiente esaminare le colonne della matrice di transizione L o della matrice L^k nel caso si ricerchino gli stati genitori in k passi, come descritto nella seguente proposizione.

Proposizione 3.2.2. *Gli stati genitore in k passi di uno stato $v \in \mathcal{L}_{2^n}$ sono ottenibili tramite:*

$$L^{-k}(v) = \{\delta_{2^n}^i \mid \text{Col}_i(L^k) = v\}, \quad k = 1, 2, \dots, T_t. \quad (3.32)$$

Esempio 3.2.1. Con riferimento all'Esempio 3.1.1, gli insiemi contenenti gli stati appartenenti ai cicli sono:

$$\begin{aligned} \mathcal{C}_1 &= \{x_1\}, & \mathcal{C}_2 &= \{x_5, x_6\} \\ \mathcal{C}_3 &= \{x_2, x_3, x_7\} \end{aligned} \quad (3.33)$$

Come visto il tempo di assorbimento è $T_t = 1$ ed è dunque sufficiente calcolare $L^{-1}(\mathcal{C}_i)$, $i = 1, 2, 3$, per ottenere i bacini di attrazione. Applicando la Proposizione 3.2.1 e la Proposizione 3.2.2 si ottiene:

$$\begin{aligned} \mathcal{S}_1 &= \{x_1, x_8\}, & \mathcal{S}_2 &= \{x_5, x_6, x_4\} \\ \mathcal{S}_3 &= \{x_2, x_3, x_7\}. \end{aligned} \quad (3.34)$$

△

Si conclude osservando che in una Boolean network l'unione dei bacini di attrazione \mathcal{S}_i crea l'intero spazio di stato \mathcal{L}_{2^n} ed essendo i bacini d'attrazione disgiunti essi formano una partizione completa di \mathcal{L}_{2^n} .

3.2.2 Accessibilità e classi di comunicazione

Un modo naturale per dividere gli stati è in base alla possibilità (o non) di transitare da uno stato ad un altro. Tale tipo di suddivisione è applicata comunemente nello studio delle catene di Markov, ma trova un'utile applicazione anche nelle Boolean networks.

Definizione 3.6. In una Boolean network descritta da (3.1) uno stato x_i è detto *accessibile* dallo stato x_j se esiste un k , $0 \leq k \leq 2^{n-1}$ tale che $x_i = L^k x_j$.

Basandosi sul grafo di stato della BN, lo stato x_i è accessibile dallo stato x_j se nel grafo esiste un cammino orientato che congiunge x_j a x_i . Essendo il grafo di stato un grafo orientato si deduce che la relazione di accessibilità non è simmetrica, ovvero il fatto che lo stato x_i sia accessibile dallo stato x_j non implica che x_j sia accessibile dallo stato x_i . Nel caso due stati siano accessibili l'un l'altro si parla di comunicabilità.

Definizione 3.7. In una Boolean network descritta da (3.1) si dice che uno stato x_i comunica con lo stato x_j se esistono $k, h, 0 \leq k, h \leq 2^{n-1}$ tali che $x_i = L^k x_j$ e $x_j = L^h x_i$, ovvero se x_i è accessibile da x_j e x_j è accessibile da x_i .

A differenza dell'accessibilità, la relazione di comunicabilità risulta essere simmetrica. Inoltre tale relazione è riflessiva, infatti ponendo $h = 0$ e $k = 0$ risulta che uno stato è sempre in comunicazione con se stesso, e gode della proprietà transitiva, infatti se x_i comunica con x_l e x_l comunica con x_j allora x_i comunica con x_j . Ne consegue che la relazione di comunicabilità è una relazione di equivalenza e permette di partizionare l'intero spazio di stato \mathcal{L}_{2^n} in classi di equivalenza, dette classi di comunicazione.

Definizione 3.8. In una BN una *classe di comunicazione* \mathcal{K} è composta da tutti gli stati che comunicano tra di loro, ovvero $x_i \in \mathcal{K}$ se e solo se x_i comunica con tutti gli stati appartenenti alla classe \mathcal{K} .

Si ha dunque che due stati x_i e x_j possono trovarsi in solo uno dei seguenti casi:

- x_i e x_j appartengono alla stessa classe di comunicazione, cioè x_i comunica con x_j .
- x_i e x_j appartengono a classi di comunicazione diverse, ovvero almeno uno dei due stati non è accessibile dall'altro.

Data la natura deterministica delle BNs è già possibile notare che classi di comunicazione composte da più di uno stato corrispondono evidentemente ad un ciclo. Per classificare le classi composte da solo uno stato risulta utile introdurre il concetto di classe chiusa e di classe transitoria.

Definizione 3.9. In una Boolean network descritta da (3.1), una classe di comunicazione \mathcal{K} viene definita:

- chiusa (o ergodica) se non esistono transizioni da stati di \mathcal{K} verso stati di altre classi, cioè se nessuno stato esterno a \mathcal{K} è accessibile da uno stato di \mathcal{K} ;
- transitoria se sono possibili transizioni da stati di \mathcal{K} verso stati di altre classi.

Esempio 3.2.2. Con riferimento all'Esempio 3.1.1, gli stati della BN possono essere divisi nelle seguenti classi di comunicazione:

$$\begin{aligned} \mathcal{K}_1 &= \{x_1\}, & \mathcal{K}_2 &= \{x_2, x_3, x_7\}, \\ \mathcal{K}_3 &= \{x_4\}, & \mathcal{K}_4 &= \{x_5, x_6\}, \\ \mathcal{K}_5 &= \{x_8\}. \end{aligned} \tag{3.35}$$

Sono classi chiuse le classi \mathcal{K}_1 , \mathcal{K}_2 e \mathcal{K}_4 , mentre le classi \mathcal{K}_3 e \mathcal{K}_5 sono transitorie, dal momento che da \mathcal{K}_3 è possibile accedere a \mathcal{K}_4 e da \mathcal{K}_5 è possibile accedere a \mathcal{K}_1 . In Figura 3.2 è riportato il grafo di stato con evidenziate le diverse classi di comunicazione.

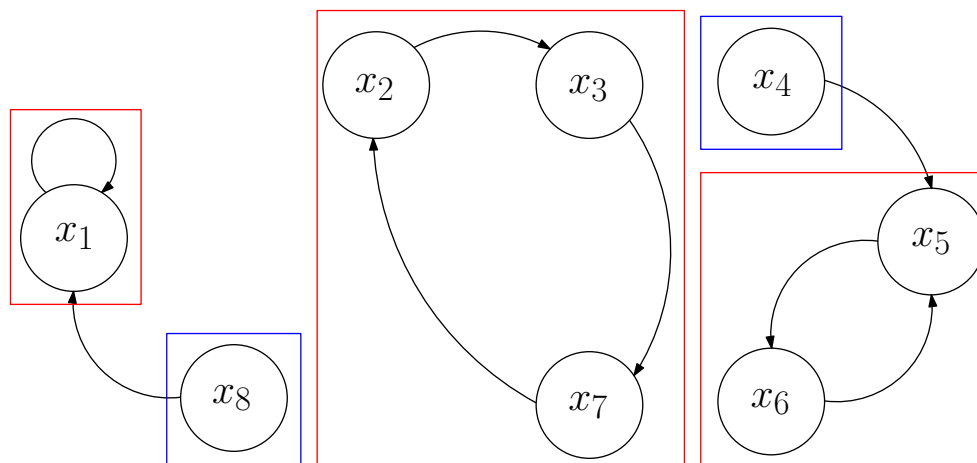


Figura 3.2: Grafo di stato della BN dell'Esempio 3.2.2: sono evidenziate in blu le classi transitorie ed in rosso le classi chiuse.

△

È ora possibile esprimere il legame che collega classi di comunicazione e cicli in una Boolean network.

Proposizione 3.2.3. *Data la Boolean network descritta da (3.1) si ha che:*

- *Una classe di comunicazione chiusa è composta da stati appartenenti ad un ciclo. Nel caso in cui la classe chiusa sia composta da un solo stato essa corrisponde ad un punto fisso della rete.*
- *Una classe di comunicazione transitoria è composta da uno solo stato, il quale non appartiene a nessun ciclo. Gli stati che compongono le classi transitorie sono detti stati transitori.*

Come detto la descrizione tramite classi di comunicazione è uno strumento applicato alle catene di Markov, ovvero in un ambito probabilistico. Le Boolean network trattate in questa tesi sono deterministiche, ma esistono anche reti Booleane probabilistiche, nelle quali la transizione da un nodo all'altro è legata ad una certa probabilità; risulta evidente quindi che la descrizione tramite classi di comunicazione è uno strumento particolarmente utile se applicato alle BNs probabilistiche.

3.3 Stabilità nelle BNs

In questa sezione viene esposto il concetto di stabilità in una Boolean network. Data la possibilità di poter esprimere la dinamica di una BN tramite l'equazione algebrica (3.1), risulta naturale l'applicazione degli strumenti della Teoria dei Sistemi per la ricerca delle condizioni che portano alla stabilità della rete.

Dalla definizione di punto fisso, si deduce che gli unici stati stabili in una Boolean network sono i punti fissi.

Definizione 3.10. Una BN si dice *globalmente stabile* se il suo insieme limite Ω è composto da un unico punto fisso.

Si deduce che per studiare la convergenza globale di una BN è sufficiente verificare che ci sia un unico attrattore e che esso sia un punto fisso. Tuttavia, conoscendo la forma algebrica di una BN, la stabilità globale può essere studiata direttamente indagando la matrice di transizione, come esposto nella seguente proposizione.

Proposizione 3.3.1. *La Boolean network descritta da (3.1) è globalmente stabile se e solo se una delle seguenti condizioni equivalenti sono soddisfatte:*

1. $\exists i$ tale che $L^{T_t} = \delta_{2^n}[i, i, \dots, i]$, dove T_t è il tempo di transizione della rete;
2. $\text{tr}(L) = 1$ e l'unica potenza non banale di L è 1, ovvero l'unica potenza s per cui $(\text{tr}(L^s) - \sum_{k \in \mathcal{P}(s)} k N_k) > 0$, è $s = 1$.

Dimostrazione. Per la dimostrazione del primo punto, se $L^{T_t} = \delta_{2^n}[i, i, \dots, i]$ significa che partendo da qualsiasi stato iniziale la rete ha raggiunto lo stato $\delta_{2^n}^i$ dopo T_t iterazioni, dunque la rete è stabile. Viceversa se la BN è globalmente stabile, dopo il tempo di transizione T_t essa si trova nell'unico punto fisso presente $\delta_{2^n}^i$, quindi L^{T_t} deve avere la forma $L^{T_t} = \delta_{2^n}[i, i, \dots, i]$.

Per il secondo punto, se $\text{tr}(L) = 1$ significa che esiste un unico ciclo di lunghezza uno, mentre la non esistenza di cicli di lunghezza superiore è assicurata dal fatto che l'unica potenza non banale è 1; ne consegue che la BN è globalmente stabile. Viceversa la globale stabilità della rete assicura la presenza di un solo ciclo di lunghezza 1, quindi $\text{tr}(L) = 1$ e l'unica potenza non banale è 1. \square

La condizione per cui la rete risulta globalmente stabile se L^{T_t} ha tutte le colonne uguali richiede implicitamente la conoscenza del tempo di transizione T_t . In realtà è possibile verificare la stabilità di una BN anche senza conoscere T_t .

Corollario 3.3.2. *Una Boolean network è globalmente stabile se e solo se $\exists i$ tale che $L^{2^n} = \delta_{2^n}[i, i, \dots, i]$.*

Dimostrazione. Se il sistema è globalmente stabile, allora esiste un indice i tale che $L^k = \delta_{2^n}[i, i, \dots, i]$, $\forall k \geq T_t$ ed essendo $2^n \geq T_t$ risulta che $L^{2^n} = \delta_{2^n}[i, i, \dots, i]$. Viceversa se L^{2^n} ha tutte le colonne uguali e pari a $\delta_{2^n}^i$ significa che esiste un intero $r \leq 2^n$ tale che

$$L^k = \delta_{2^n}[i, i, \dots, i], \quad \forall k \geq r. \quad (3.36)$$

Dalla definizione di tempo di transizione si deduce che il più piccolo r per cui si verifica (3.36) è proprio T_t e dunque la BN è globalmente stabile. \square

Esempio 3.3.1. Si considerino quattro Boolean networks composte da un solo nodo ($n = 1$), le cui matrici di transizione nella forma algebrica sono rispettivamente:

$$\begin{aligned} L_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & L_2 &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ L_3 &= \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, & L_4 &= \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}. \end{aligned} \quad (3.37)$$

Essendo le BNs composte da un solo nodo, gli stati di ciascuna rete sono 2. Per quanto riguarda la prima BN si ha che $\text{tr}(L_1) = 2$, si hanno dunque due punti fissi e il sistema non è globalmente stabile. In questo caso qualsiasi sia lo stato iniziale del nodo esso vi rimarrà costantemente, quindi lo stato finale del sistema dipende dalle condizioni iniziali.

La seconda BN risulta anch'essa instabile, infatti $\text{tr}(L_2) = 0$. In questo caso il nodo della rete continua a cambiare il proprio valore, il ché corrisponde, nello spazio di stato, alla presenza un unico ciclo che comprende entrambi gli stati.

Per la terza e la quarta BN si ha rispettivamente che $L_3^2 = \delta_2[1, 1]$ e $L_4^2 = \delta_2[2, 2]$, dunque entrambe le reti sono globalmente stabili. Infatti per la BN descritta da L_3 , a prescindere dallo stato iniziale, l'unico stato in un passo assumerà il valore 1, mentre nella rete descritta da L_4 assumerà il valore 0.

I grafi di stato delle Boolean networks prese in esame sono riportati in Figura 3.3. \triangle

3.3.1 Cicli globalmente attrattivi

Talvolta la richiesta che una Boolean network sia globalmente stabile può essere troppo stringente e risulta sufficiente che a regime la rete si trovi in un determinato insieme di stati (e non soltanto uno). Tali stati devono naturalmente appartenere ad un ciclo per garantire che la BN possa trovarsi indistintamente in uno di essi e non in altri stati e questo ciclo deve essere unico; questo ciclo prende il nome di ciclo attrattivo globalmente. In questo paragrafo verranno analizzati degli strumenti algebrici atti a studiare la presenza ed il numero di cicli in una BN, con particolare

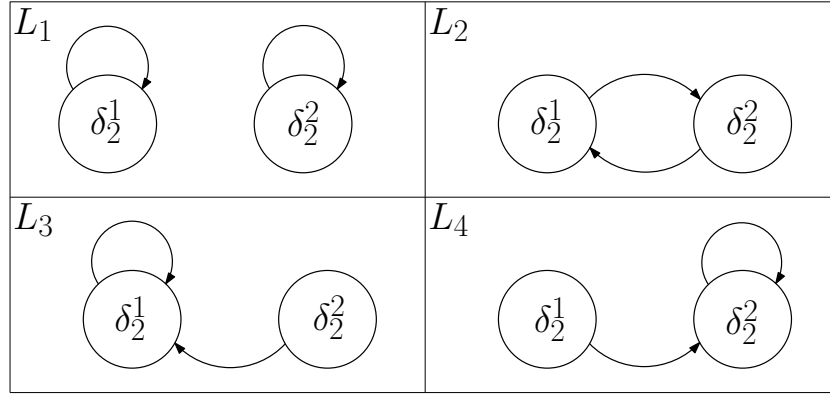


Figura 3.3: Grafi di stato delle BNs dell'Esempio 3.3.1, con indicate le rispettive matrici di transizione.

attenzione al caso in cui sia presente un solo ciclo.

Per facilitare l'esposizione viene data una definizione alternativa di ciclo limite e successivamente la definizione formale di ciclo globalmente attrattivo.

Definizione 3.11. Una sequenza ordinata di vettori $(\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \dots, \delta_{2^n}^{i_k})$ è un ciclo limite \mathcal{C} della BN (3.1) se $x(0) = \delta_{2^n}^{i_l}$ per qualche $l \in [1, k]$, assicura che la corrispondente traiettoria di stato sia periodica di periodo k e $\forall t \in \mathbb{Z}_+, x(t) = \delta_{2^n}^{i_j}$, dove $j \equiv (t + l) \pmod{k}$.

Definizione 3.12. Data una BN composta da n nodi, una ciclo limite \mathcal{C} della BN è globalmente attrattivo se per ogni stato iniziale $x(0)$ della rete esiste $T \in \mathbb{Z}_+$ tale che $x(t)$ appartiene a \mathcal{C} per ogni $t \in \mathbb{Z}_+, t \geq T$.

Da quanto visto nella Sezione 3.1.1 si deduce la seguente proposizione.

Proposizione 3.3.3. *Se una Boolean network possiede un solo ciclo, allora esso è un ciclo globalmente attrattivo. Il tempo T della Definizione 3.12 esiste sempre e corrisponde al tempo di attrazione T_t .*

Dimostrazione. L'esistenza di T_t è già stata provata. Per dimostrare che il ciclo deve essere unico, si supponga per assurdo che esistano due cicli, indicati con \mathcal{C}_i e \mathcal{C}_j , e che uno di essi sia un ciclo globalmente attrattivo. Se $x(0)$ appartiene al bacino di attrazione \mathcal{S}_i di \mathcal{C}_i allora $x(t) \in \mathcal{C}_i, \forall t \geq T_t$, mentre se $x(0)$ appartiene al bacino di attrazione \mathcal{S}_j di \mathcal{C}_j allora $x(t) \in \mathcal{C}_j, \forall t \geq T_t$. Ne consegue che nessuno dei due cicli è un ciclo globalmente attrattivo. \square

Spesso per caratterizzare i cicli presenti in una Boolean network risulta utile manipolare la matrice di transizione L in modo da portarla in una forma che renda evidenti gli stati presenti in ciascun ciclo ed anche i loro rispettivi bacini

di attrazione. Tale forma permette naturalmente di individuare la presenza di un ciclo globalmente attrattivo.

Proposizione 3.3.4. *Data una BN descritta da (3.1), esistono un intero r e una matrice di permutazione $\Pi \in \mathcal{L}_{2^n \times 2^n}$ tali che*

$$\Pi^T L \Pi = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_r \end{bmatrix}, \quad (3.38)$$

$$\text{con } D_i = \begin{bmatrix} N_i & 0 \\ T_i & C_i \end{bmatrix} \in \mathcal{L}_{2^{n_i} \times 2^{n_i}}, \quad (3.39)$$

dove N_i è una matrice nilpotente di dimensione $(n_i - k_i) \times (n_i - k_i)$, e C_i è una matrice ciclica di dimensione $k_i \times k_i$.

Dimostrazione. Siano $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_r$, i cicli limite del sistema (compresi i punti fissi) e siano $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_r$ i corrispondenti bacini d'attrazione. Esiste sicuramente una matrice di permutazione Π tale che $\Pi^T L \Pi$ sia diagonale a blocchi come in (3.38). Si possono poi ordinare gli stati appartenenti a ciascun bacino \mathcal{S}_i , $i = 1, 2, \dots, r$, in modo tale che gli stati appartenenti a \mathcal{C}_i siano gli ultimi. Allora, senza perdita di generalità, si assuma che ciascun blocco diagonale abbia la forma (3.39), in cui C_i è la matrice ciclica associata a \mathcal{C}_i . Tutti gli stati transitori appartenenti a \mathcal{S}_i , producono traiettorie che dopo un numero di passi uguale o inferiore a T_t appartengono a \mathcal{C}_i ; questo comporta che per $k \geq T_t$ si ha che

$$D_i^k = \begin{bmatrix} N_i^k & 0 \\ \star & C_i^k \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \star & C_i^k \end{bmatrix}, \quad (3.40)$$

ovvero N_i è una matrice nilpotente. □

Nella matrice $\Pi^T L \Pi$ ad ogni blocco D_i corrisponde dunque un solo ciclo, ed in particolare l'intero bacino di attrazione del ciclo stesso. Se la matrice presenta r blocchi, allora sono presenti r cicli nella rete, i cui stati sono deducibili dalla matrici C_i . Un'altra informazione ricavabile da $\Pi^T L \Pi$ riguarda il tempo di transizione della BN: si ha infatti che $T_t \leq \max_{i=1, \dots, r} (n_i - k_i)$, dunque è possibile limitare superiormente il tempo di transizione.

Date le considerazioni fatte precedentemente risulta ovvia la seguente proposizione che caratterizza una rete con un ciclo globalmente attrattivo.

Proposizione 3.3.5. *Data la Boolean network descritta da (3.1), una sequenza ordinata di vettori logici $\mathcal{C} = (\delta_{2^n}^{i_1}, \delta_{2^n}^{i_2}, \dots, \delta_{2^n}^{i_k})$ è un ciclo globalmente attrattivo della BN se e solo se esiste una matrice di permutazione $\Pi \in \mathcal{L}_{2^n \times 2^n}$ tale che $\Pi^T L \Pi$ abbia la forma descritta in (3.38) e (3.39) con $r = 1$ e con C_1 matrice circolare di dimensione $k \times k$, ovvero pari alla lunghezza del ciclo globalmente attrattivo.*

Un ultimo metodo per constatare la presenza di un ciclo globalmente attrattivo è mediante l'analisi del polinomio caratteristico della matrice di transizione. È risaputo che due matrici che differiscono solo per un cambiamento di base hanno lo stesso polinomio caratteristico, dunque è possibile affermare che L e $\Pi^T L \Pi$ posseggono lo stesso polinomio caratteristico. Segue direttamente dalla struttura di $\Pi^T L \Pi$ che il polinomio caratteristico della matrice di transizione associata ad una BN con n nodi ha la forma:

$$\Delta_L(z) := \det(zI_{2^n} - L) = \left(z^{2^n - \sum_{i=1}^r k_i} \right) \prod_{i=1}^r (z^{k_i} - 1). \quad (3.41)$$

La seguente proposizione discende direttamente dalla struttura di $\Delta_L(z)$.

Proposizione 3.3.6. *Una Boolean network descritta da (3.1) possiede un ciclo globalmente attrattivo (di lunghezza k) se e solo se $\Delta_L(z) = z^{2^n - k} (z^k - 1)$.*

Vengono riportati due esempi in cui grazie alla forma $\Pi^T L \Pi$ vengono individuati i cicli della Boolean network, verificando anche la presenza di un ciclo globalmente attrattivo.

Esempio 3.3.2. Si consideri una BN composta da 4 nodi la cui forma algebrica è descritta da

$$x(t+1) = Lx(t), \quad (3.42)$$

dove $x(t) \in \mathcal{L}_{16}$ e

$$L = \delta_{16}[2, 3, 4, 13, 6, 6, 8, 7, 8, 9, 16, 12, 3, 6, 9, 11]. \quad (3.43)$$

Utilizzando la matrice di permutazione

$$\Pi = \delta_{16}[1, 2, 3, 4, 13, 5, 14, 6, 15, 10, 9, 8, 7, 11, 16, 12], \quad (3.44)$$

si ottiene la forma (3.38), ovvero

$$\begin{aligned} \Pi^T L \Pi &= \delta_{16}[2, 3, 4, 5, 3, 8, 8, 8, 11, 11, 12, 13, 12, 15, 14, 16] \\ &= \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & D_5 \end{bmatrix}. \end{aligned} \quad (3.45)$$

I blocchi D_i , $i = 1, \dots, 5$, hanno la forma (3.39), in particolare

$$D_1 = \left[\begin{array}{cc|ccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array} \right] \quad D_2 = \left[\begin{array}{cc|c} 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 1 & 1 & 1 \end{array} \right] \quad (3.46)$$

$$D_3 = \left[\begin{array}{ccc|cc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{array} \right] \quad D_4 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad D_5 = [1].$$

In ciascun blocco si può riconoscere la matrice ciclica C_i , rappresentante gli stati appartenenti al ciclo i -esimo, e la matrice nilpotente N_i (se presente). Indicando con \tilde{x}_i , $i = 1, 2, \dots, 16$, gli stati della rete nel nuovo spazio di stato relativo al cambiamento di base indotto da Π , risulta immediato indicare cicli e relativi bacini di attrazione, in particolare risulta che:

$$\begin{aligned} \tilde{\mathcal{C}}_1 &= \{\tilde{x}_3, \tilde{x}_4, \tilde{x}_5\}, & \tilde{\mathcal{S}}_1 &= \tilde{\mathcal{C}}_1 \cup \{\tilde{x}_1, \tilde{x}_2\}; \\ \tilde{\mathcal{C}}_2 &= \{\tilde{x}_8\}, & \tilde{\mathcal{S}}_2 &= \tilde{\mathcal{C}}_2 \cup \{\tilde{x}_6, \tilde{x}_7\}; \\ \tilde{\mathcal{C}}_3 &= \{\tilde{x}_{12}, \tilde{x}_{13}\}, & \tilde{\mathcal{S}}_3 &= \tilde{\mathcal{C}}_3 \cup \{\tilde{x}_9, \tilde{x}_{10}, \tilde{x}_{11}\}; \\ \tilde{\mathcal{C}}_4 &= \{\tilde{x}_{14}, \tilde{x}_{15}\}, & \tilde{\mathcal{S}}_4 &= \tilde{\mathcal{C}}_4; \\ \tilde{\mathcal{C}}_5 &= \{\tilde{x}_{16}\}, & \tilde{\mathcal{S}}_5 &= \tilde{\mathcal{C}}_5. \end{aligned} \quad (3.47)$$

Il polinomio caratteristico di L , calcolato grazie all'ausilio di `Matlab`, è

$$\Delta_z(L) = z^{16} - 2z^{15} - z^{14} + 3z^{13} + z^{12} - z^{11} - 3z^{10} + 2z^9 + 2z^8 - z^7; \quad (3.48)$$

naturalmente non è necessario ricorrere all'utilizzo di un software esterno, infatti sappiamo che il polinomio caratteristico di una matrice di transizione ha la forma (3.41). Procedendo con i calcoli si ottiene

$$\begin{aligned} \Delta_z(L) &= z^{16-(3+1+2+2+1)} (z^3 - 1) (z^2 - 1) (z^2 - 1) (z - 1) (z - 1) \\ &= z^7 (z^3 - 1) (z^2 - 1)^2 (z - 1)^2, \end{aligned} \quad (3.49)$$

che coincide con il polinomio calcolato precedentemente.

Sia dalla forma della matrice $\Pi^T L \Pi$, sia dal polinomio caratteristico $\Delta_z(L)$ risulta evidente la non esistenza di un ciclo globalmente attrattivo. In Figura 3.4 è riportato il grafo di stato della BN, nel quale si può constatare la presenza di 3 cicli e 2 punti fissi, in accordo con i risultati ottenuti (si presti attenzione al fatto che il grafo presenta gli stati originari x_i , $i = 1, \dots, 16$, e non quelli ottenuti dopo il cambiamento di base indotto da Π). \triangle

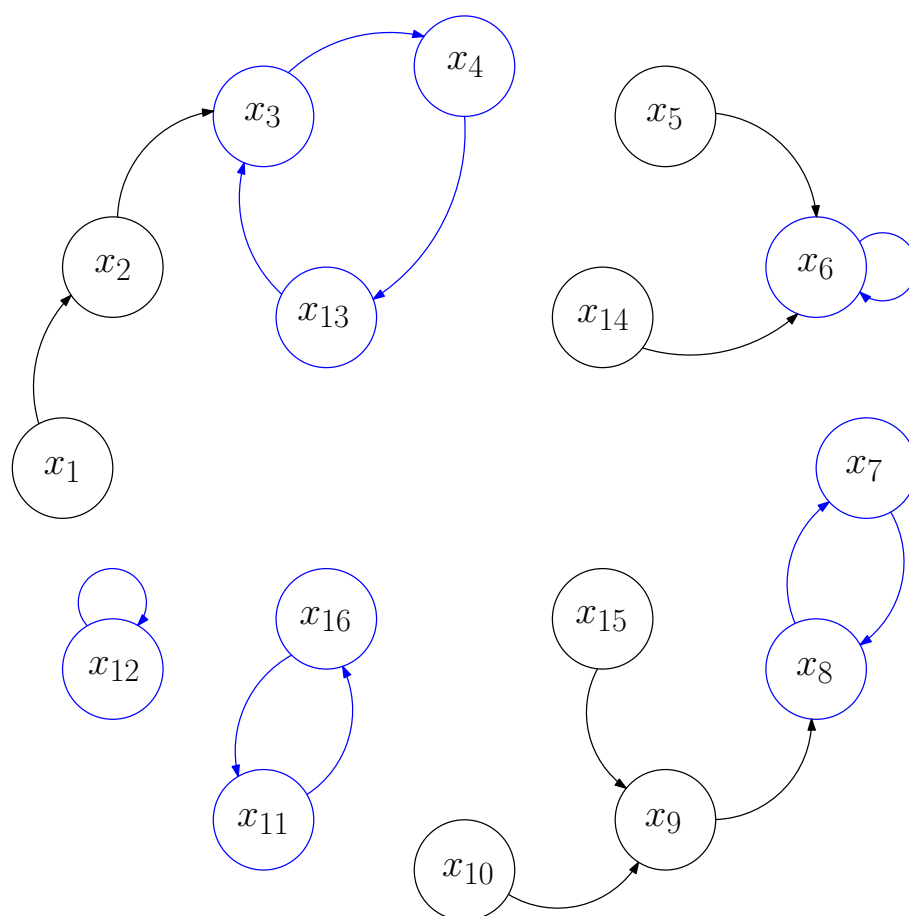


Figura 3.4: Grafo di stato della BN dell'Esempio 3.3.2: in blu sono evidenziati i cicli della rete.

Esempio 3.3.3. Si consideri una BN composta da 2 nodi la cui forma algebrica è descritta da

$$x(t+1) = Lx(t), \quad (3.50)$$

dove $x(t) \in \mathcal{L}_4$ e $L = \delta_4[3, 3, 2, 2]$. Utilizzando la matrice di permutazione $\Pi = \delta_4[1, 4, 2, 3]$ si ottiene la forma (3.38), ovvero:

$$\begin{aligned} \Pi^T L \Pi &= \left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{array} \right] \\ &= \left[\begin{array}{c|c} N_1 & 0 \\ \hline \star & C_1 \end{array} \right]. \end{aligned} \quad (3.51)$$

Si può notare che $\Pi^T L \Pi$ è composta da un unico blocco della forma (3.39), in particolare la matrice circolare C_1 ha dimensione 2×2 , il che comporta la presenza di un ciclo di lunghezza 2; la matrice nilpotente N_1 ha anch'essa dimensione 2×2 , dunque gli stati appartenenti al bacino di attrazione del ciclo ma non al ciclo stesso sono due. Osservando nel dettaglio $\Pi^T L \Pi$ si conclude che l'unico ciclo presente (riferito alla nuova base indotta da Π) è $\tilde{\mathcal{C}}_1 = \{\tilde{x}_3, \tilde{x}_4\}$, mentre il suo bacino di attrazione è $\tilde{\mathcal{C}}_1 = \{\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4\}$. Grazie alla Proposizione 3.3.5 si può affermare che la BN ha un ciclo globalmente attrattivo che nello spazio di stato originario è $\mathcal{C}_1 = \{x_2, x_3\}$. Un altro modo per constatare la presenza del ciclo globalmente attrattivo è analizzare il polinomio caratteristico di L : procedendo con i calcoli si ottiene

$$\begin{aligned} \Delta_z(L) &= z^4 - z^2 \\ &= z^2(z^2 - 1), \end{aligned} \quad (3.52)$$

che risulta avere proprio la forma descritta nella Proposizione 3.3.6 per $k = 2$.

I risultati ottenuti sono confermati dal grafo di stato della BN riportato in Figura 3.5. △

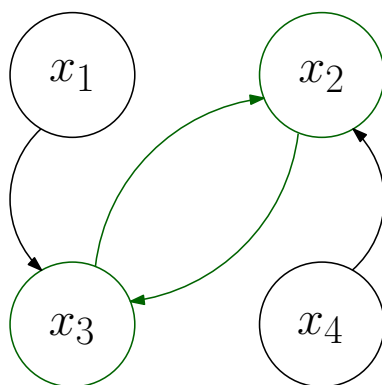


Figura 3.5: Grafo di stato della BN dell'Esempio 3.3.3: in verde è evidenziato il ciclo globalmente attrattivo della rete.

3.4 Raggiungibilità e controllabilità nelle BCNs

Le proprietà delle Boolean control networks sono fortemente legate alla capacità degli ingressi di portare la rete in determinati stati. Acquisiscono una grande importanza dunque quelle proprietà che riescono a descrivere come gli ingressi riescano ad influenzare la rete. A differenza delle BNs, in cui lo studio effettuato è stato per lo più di carattere asintotico, per le BCNs ci si concentrerà maggiormente

sull'analisi degli effetti degli ingressi, cercando di analizzare come sia possibile controllare la rete mediante essi. In particolare verranno approfonditi i concetti di raggiungibilità di uno stato, di controllabilità e di stabilizzabilità dell'intera rete.

3.4.1 Raggiungibilità

Definizione 3.13. Data una Boolean control network descritta da (3.2), lo stato $x_i = \delta_{2^n}^i$ si dice raggiungibile in τ passi dallo stato $x_j = \delta_{2^n}^j$ se esiste una sequenza di ingressi $u(0), u(1), \dots, u(\tau - 1)$, tali da portare la traiettoria dello stato da $x(0) = x_i$ a $x(\tau) = x_j$.

Definizione 3.14. Data una Boolean control network descritta da (3.2), lo stato $x_i = \delta_{2^n}^i$ si dice raggiungibile dallo stato $x_j = \delta_{2^n}^j$ se esistono $\tau \in \mathbb{Z}_+$ ed una sequenza di ingressi $u(0), u(1), \dots, u(\tau - 1)$, tali da portare la traiettoria dello stato da $x(0) = x_i$ a $x(\tau) = x_j$.

L'insieme degli stati raggiungibili in τ passi dallo stato x_i viene indicato con $R_\tau(x_i) \subseteq \mathcal{L}_{2^n}$; l'insieme di tutti gli stati raggiungibili da x_i , senza imporre un preciso numero di passi, viene indicato con $R(x_i)$ e risulta essere

$$R(x_i) = \bigcup_{\tau=0}^{+\infty} R_\tau(x_i). \quad (3.53)$$

Dato che gli stati in una BCN sono in numero finito, ovvero 2^n , per definire $R(x_i)$ non è necessario conoscere gli insiemi di raggiungibilità in τ passi per ogni $\tau \geq 1$, bensì è sufficiente fermarsi al passo 2^n , come dimostrato nella seguente proposizione.

Proposizione 3.4.1. *Data una BCN descritta da (3.2), se uno stato x_i è raggiungibile da uno stato x_j allora x_i è raggiungibile da x_j in $\tau \leq 2^n$ passi.*

Dimostrazione. Si supponga x_i raggiungibile da x_j e si consideri la più breve sequenza di stati tale che $x(0) = x_j, x(1), \dots, x(\tau) = x_i$. Se, per assurdo, risultasse $\tau > 2^n$, significherebbe che la rete è transitata per un numero di stati maggiore di quelli da cui è composta. Di conseguenza il sistema deve essere transitato per un ciclo ed averlo percorso interamente almeno una volta: ciò implica che esiste un percorso più breve per raggiungere x_i , il che dimostra l'assurdo. \square

L'insieme degli stati raggiungibili dallo stato x_i può quindi essere riscritto come

$$R(x_i) = \bigcup_{\tau=1}^{2^n} R_\tau(x_i). \quad (3.54)$$

Definizione 3.15. Se l'insieme degli stati raggiungibili da uno stato x_i coincide con lo spazio di stato, ovvero se $R(x_i) \equiv \mathcal{L}_{2^n}$, la BCN viene detta globalmente raggiungibile da x_i .

Osservazione 8. Nel seguito con la scrittura x_i si intenderà implicitamente il riferimento allo stato $\delta_{2^n}^i$. In ogni caso che possa presentare ambiguità verrà specificato esplicitamente lo stato a cui ci si riferisce.

Tramite l'analisi della matrice L_{tot} è possibile acquisire tutte le informazioni sulla raggiungibilità degli stati di una rete, come descritto nel seguente teorema.

Teorema 3.4.2. *Data una BCN descritta da (3.2) e la matrice L_{tot} ad essa associata, si ha che lo stato x_i è raggiungibile in τ passi da x_j se e solo se*

$$[L_{tot}^\tau]_{i,j} = 1. \quad (3.55)$$

Dimostrazione. Si dimostra solamente che (3.55) implica che x_i è raggiungibile da x_j in τ passi, essendo il risultato più utile. La dimostrazione dell'implicazione inversa è del tutto simile.

Si consideri il caso base in cui $[L_{tot}^1]_{i,j} = 1$, allora esiste un ingresso $u(0) = \delta_{2^m}^k$ per cui $\text{Col}_j(L_k) = \delta_{2^n}^i$ e di conseguenza si ha che $L u(0) x_j = L_k x_j = x_i$, quindi x_i risulta raggiungibile in un passo da x_j .

Si assuma per ipotesi induttiva che $[L_{tot}^{\tau-1}]_{i,j} = 1$ implichi che $x_i \in R_{\tau-1}(x_j)$.

Si consideri quindi $[L_{tot}^\tau]_{i,j} = 1$; la stessa espressione può essere scritta come

$$[L_{tot}^\tau]_{i,j} = \bigvee_{h=1}^{2^n} [L_{tot}]_{i,h} [L_{tot}^{\tau-1}]_{h,j} = 1, \quad (3.56)$$

e quindi deve esistere un h per cui $[L_{tot}]_{i,h} = 1$ e $[L_{tot}^{\tau-1}]_{h,j} = 1$.

$[L_{tot}^{\tau-1}]_{h,j} = 1$ implica che lo stato $x_h = \delta_{2^n}^h$ è raggiungibile in $\tau - 1$ passi da x_j , mentre $[L_{tot}]_{i,h} = 1$ comporta che x_i è raggiungibile in 1 passo da x_h . In conclusione x_i risulta raggiungibile in τ passi da x_j . \square

I seguenti corollari, che discendono direttamente dalle considerazioni fatte precedentemente, vanno a caratterizzare il concetto di raggiungibilità e raggiungibilità globale mediante la matrice L_{tot} .

Corollario 3.4.3. *Lo stato x_i è raggiungibile dallo stato x_j se e solo se $\exists \tau \in [1, 2^n]$ tale che $[L_{tot}^\tau]_{i,j} = 1$, o equivalentemente se e solo se*

$$\left[\bigvee_{\tau=1}^{2^n} L_{tot}^\tau \right]_{i,j} = 1. \quad (3.57)$$

Corollario 3.4.4. *La BCN descritta da (3.2) è globalmente raggiungibile da x_i se e solo se*

$$\bigvee_{\tau=1}^{2^n} Col_i(L_{tot}^\tau) \gg 0. \quad (3.58)$$

Osservazione 9. Tutte le considerazioni fatte sulla raggiungibilità risultano più intuitive se fatte mediante analisi del grafo di stato della BCN: risulta evidente che lo stato x_i è raggiungibile dallo stato x_j se e solo se nel grafo di stato esiste un cammino orientato con origine in x_j e arrivo in x_i . Similmente la BCN è globalmente raggiungibile da x_i se e solo se per ogni stato x_j , $j = 1, \dots, 2^n$, esiste un cammino orientato che va dallo stato x_i allo stato x_j .

3.4.2 Controllabilità

Definizione 3.16. Una BCN si dice controllabile allo stato x_i se esso risulta raggiungibile da ogni altro stato del sistema.

Definizione 3.17. Una BCN viene detta globalmente controllabile se x_i risulta raggiungibile da x_j , per ogni scelta della coppia di stati $x_i, x_j \in \mathcal{L}_{2^n}$.

Dato il legame tra raggiungibilità e controllabilità, anche per quest'ultima proprietà risulta utile l'analisi della matrice L_{tot} . In questo caso l'informazione più interessante è contenuta nelle righe della matrice: gli elementi unitari della i -esima riga della matrice L_{tot} corrispondono agli stati dai quali è possibile giungere allo stato x_i in un passo. Analizzando le righe di L_{tot}^τ è possibile estendere il discorso al caso in cui siano richiesti τ passi. Il seguente teorema caratterizza la controllabilità ad uno stato mediante la matrice L_{tot} .

Teorema 3.4.5. *La BCN descritta da (3.2) è controllabile a x_i se e solo se*

$$\bigvee_{\tau=1}^{2^n} Row_i(L_{tot}^\tau) \gg 0, \quad (3.59)$$

dove la matrice L_{tot} è definita da (3.4).

Allo scopo di semplificare l'analisi della controllabilità globale di una BCN risulta utile introdurre la seguente matrice:

$$\mathbb{L} = \bigvee_{k=0}^{2^n-1} L_{tot}^k. \quad (3.60)$$

Dalla teoria delle matrici positive si deduce il seguente risultato.

Teorema 3.4.6. *La BCN descritta da (3.2) è globalmente controllabile se e solo la matrice L_{tot} ad essa associata è irriducibile, o equivalentemente se e solo se la matrice \mathbb{L} definita da (3.60) ha tutti gli elementi unitari.*

L'analisi della matrice \mathbb{L} permette dunque di sapere in modo immediato se una BCN è globalmente controllabile e, nel caso non lo fosse, permette di identificare gli stati non controllabili; tali stati corrispondono alle righe di \mathbb{L} non strettamente positive.

Lo svantaggio della matrice \mathbb{L} consta nel fatto che essa non porta alcuna informazione relativa al numero di passi necessari al controllo. \mathbb{L} risulta dunque inutile nell'analisi di proprietà quali la controllabilità globale in senso forte, della quale si riporta la definizione.

Definizione 3.18. Una BCN composta da n nodi è globalmente controllabile in senso forte in τ passi se esiste una sequenza di ingresso $u(0), u(1), \dots, u(\tau - 1)$ tale da portare la traiettoria dello stato da $x(0) = x_i$ a $x(\tau) = x_j$ per ogni scelta di $x_i, x_j \in \mathcal{L}_{2^n}$.

Definizione 3.19. Una BCN composta da n nodi è globalmente controllabile in senso forte se esiste un $\tau \in \mathbb{Z}_+$ tale che la BCN sia globalmente controllabile in senso forte in τ passi.

Si presti attenzione al fatto che la controllabilità globale in senso forte richiede che la BCN sia controllabile ad ogni stato del sistema in esattamente τ passi, ed è dunque una condizione molto più stringente della controllabilità classica. Ne consegue che anche la condizione relative alla matrice L_{tot} per garantire la controllabilità globale in senso forte risulti più stringente, come esposto nel seguente teorema.

Teorema 3.4.7. *La BCN descritta da (3.2) è globalmente controllabile in senso forte se e solo la matrice L_{tot} ad essa associata è una matrice primitiva.*

Dimostrazione. Perché la BCN sia globalmente controllabile in senso forte in τ passi la matrice L_{tot}^τ deve avere tutti gli elementi pari ad 1, di conseguenza se la rete è globalmente controllabile in senso forte deve esistere un $s \in \mathbb{Z}_+ : L_{tot}^s \gg 0$, ovvero L_{tot} deve essere una matrice primitiva. Il viceversa è ovvio. \square

Esempio 3.4.1. Si consideri una BCN composta da 2 nodi sulla quale agiscono 2 ingressi, la cui matrice di transizione è

$$\begin{aligned} L &= \delta_4[3, 1, 4, 2, 3, 4, 1, 3, 2, 1, 2, 3, 3, 2, 3, 1] \\ &= [L_1, L_2, L_3, L_4], \end{aligned} \quad (3.61)$$

dove $L_1 = \delta_4[3, 1, 4, 2]$, $L_2 = \delta_4[3, 4, 1, 3]$, $L_3 = \delta_4[2, 1, 2, 3]$ e $L_4 = \delta_4[3, 2, 3, 1]$ sono le matrici di transizione delle BNs associate alla BCN quando il vettore di

ingresso $u(t)$ vale rispettivamente δ_4^1 , δ_4^2 , δ_4^3 e δ_4^4 .

Risulta che

$$\begin{aligned} L_{tot} &= \bigvee_{j=1}^4 L_j \\ &= \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \end{aligned} \quad (3.62)$$

dunque la BCN non risulta globalmente raggiungibile in un passo da x_1 , x_2 e x_4 . D'altra parte si ha che

$$L_{tot}^2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad (3.63)$$

quindi la BCN risulta globalmente raggiungibile da tutti gli stati in $\tau \geq 2$ passi. Per quanto riguarda la controllabilità si ha che

$$\begin{aligned} \mathbb{L} &= \bigvee_{k=0}^2 L_{tot}^k \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \end{aligned} \quad (3.64)$$

di conseguenza la rete è globalmente controllabile. Quest'ultimo passaggio era evitabile, infatti considerando che L_{tot}^2 ha tutti gli elementi unitari, risulta che L_{tot} è una matrice primitiva, dunque la rete è globalmente controllabile in senso forte (in $\tau \geq 2$ passi). Nelle Figure 3.6 e 3.7 è rappresentato il grafo di stato della BCN (la divisione in due figure ha il solo scopo di migliorarne la chiarezza). \triangle

3.5 Stabilizzabilità

Generalmente quando si parla di stabilizzabilità si intende la capacità di portare un sistema in uno stato e di potercelo mantenere. In questo paragrafo verrà affrontato il problema di poter stabilizzare una BCN ad un ciclo: si tratta di una naturale estensione della stabilizzabilità ad uno stato che, come già visto, può essere considerato un ciclo di lunghezza unitaria.

Definizione 3.20. Una BCN descritta da (3.2) è stabilizzabile al ciclo $\mathcal{C} = (x_{i_1}, x_{i_1}, \dots, x_{i_k})$ se per ogni scelta dello stato iniziale $x(0) \in \mathcal{L}_{2^n}$ esistono $t \in \mathbb{Z}_+$, $\tau \in \mathbb{Z}_+$ e una sequenza di ingressi $u(0), u(1), \dots, u(t)$, tali che $x(t) = x_{i_l}$ per ogni $t \geq \tau$, dove $l \in [1, k]$ e $l \equiv (t - \tau + 1) \pmod{k}$.

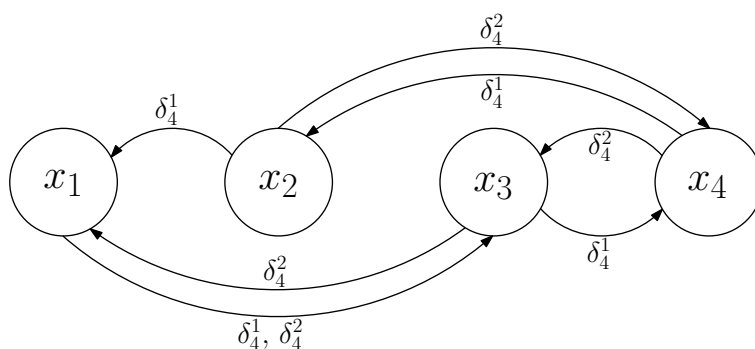


Figura 3.6: Grafo di stato della BCN dell'Esempio 3.4.1 relativo all'ingresso $u(t) = \delta_4^1$ e $u(t) = \delta_4^2$.

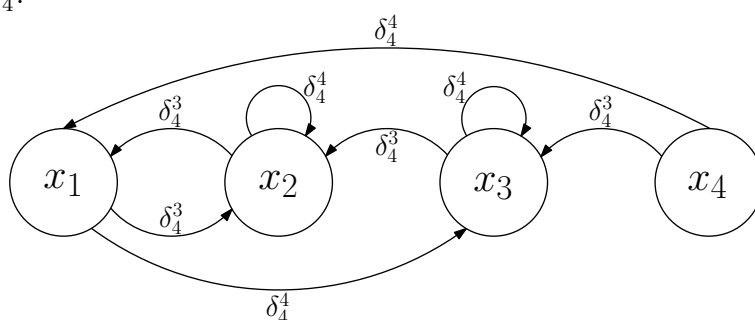


Figura 3.7: Grafo di stato della BCN dell'Esempio 3.4.1 relativo all'ingresso $u(t) = \delta_4^3$ e $u(t) = \delta_4^4$.

Osservazione 10. La definizione di rete stabilizzabile richiama parzialmente la definizione di ciclo globalmente attrattivo, infatti una BN che presenta un ciclo di quel tipo può essere vista come una BCN con un ingresso costante, il quale permette l'ingresso nel ciclo e naturalmente anche la permanenza.

Teorema 3.5.1. *Una BCN descritta da (3.2) è stabilizzabile al ciclo $\mathcal{C} = (x_{i_1}, x_{i_1}, \dots, x_{i_k})$ se e solo se sono rispettate le seguenti condizioni:*

1. per ogni coppia (i_l, i_{l+1}) , dove $l \in [1, k]$ e $i_{k+1} = i_1$, esiste un ingresso $\delta_{2^m}^j$ tale che

$$x_{l+1} = L \delta_{2^m}^j x_l = L_j x_l; \quad (3.65)$$

2. la BCN è controllabile a x_{i_1} .

Dimostrazione. Se la BCN risulta stabilizzabile al ciclo \mathcal{C} significa che esiste una sequenza di ingressi in grado di portare qualsiasi stato iniziale ad uno stato interno al ciclo e, una volta giunti all'interno del ciclo, x_{i_1} risulta sicuramente raggiungibile, dunque la BCN deve essere controllabile a x_{i_1} . Inoltre la stabilizzabilità assicura

che esiste una sequenza di ingressi tali che $x(t) \in \mathcal{C}$ per ogni $t \geq \tau$, il ché garantisce il verificarsi della prima condizione.

Viceversa il fatto che la BCN sia controllabile a x_{i_1} assicura l'ingresso della rete in \mathcal{C} , mentre l'esistenza di un ingresso $\delta_{2^m}^j$ tale che $x_{l+1} = L \delta_{2^m}^j x_l$, assicura la permanenza all'interno del ciclo, quindi la BCN risulta stabilizzabile. \square

Esempio 3.5.1. Si consideri una BCN composta da tre nodi ($n = 3$) sulla quale agisce un ingresso ($m = 1$), la cui matrice di transizione è $L = [L_1, L_2]$, dove

$$\begin{aligned} L_1 &= \delta_8[5, 6, 3, 3, 5, 7, 4, 7], \\ L_2 &= \delta_8[1, 6, 2, 3, 6, 5, 6, 7]. \end{aligned} \quad (3.66)$$

Un possibile ciclo della rete è $\mathcal{C} = (x_6, x_7)$, come osservabile dal grafo di stato riportato in Figura 3.8. Risulta che $x_6 = L_2 x_7$ e $x_7 = L_1 x_6$, dunque è rispettata la prima condizione del Teorema 3.5.1. Per verificare la seconda condizione è sufficiente controllare se la BCN è controllabile ad uno degli stati appartenenti a \mathcal{C} : prendendo in esame ad esempio x_6 risulta che

$$\bigvee_{\tau=1}^8 \text{Row}_6(L_{tot}^\tau) \gg 0, \quad (3.67)$$

dunque la rete è controllabile a x_6 . Essendo rispettate entrambe le condizioni del Teorema 3.5.1, la BCN risulta stabilizzabile a \mathcal{C} . \triangle

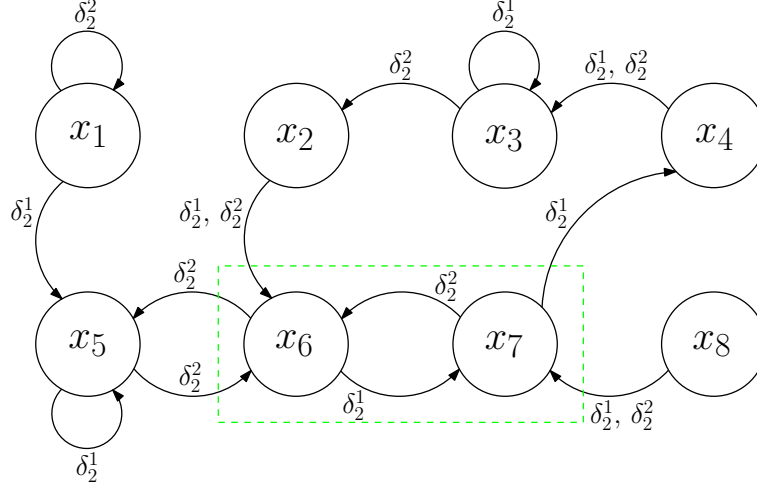


Figura 3.8: Grafo di stato della BCN dell'Esempio 3.5.1: nel riquadro verde gli stati appartenenti a \mathcal{C} .

Per quanto riguarda la possibilità di stabilizzare la rete ad uno stato è possibile applicare il seguente corollario, del tutto equivalente al Teorema 3.5.1 per $k = 1$.

Corollario 3.5.2. *Una BCN descritta da (3.2) è stabilizzabile allo stato $x_i = \delta_{2^n}^i$ se e solo se sono verificate le seguenti condizioni:*

1. x_i è un punto di equilibrio della BN associata al blocco L_j di L , per qualche $j \in [1, 2^m]$;
2. la BCN è controllabile a x_i .

Per comprendere quali sono gli stati ai quali è possibile stabilizzare una BCN, è inizialmente necessario trovare tutti i punti di equilibrio dei sottosistemi della rete descritti da L_j , ovvero identificare tutti quegli stati x_i tali che esiste un $j \in [1, 2^m]$ per i quali $[L_j]_{i,i} = 1$. Una volta identificato l'insieme di tutti i punti di equilibrio, che indichiamo con \mathcal{X}_e , bisogna identificare il sottoinsieme di \mathcal{X}_e contenente tutti quegli stati ai quali la BCN è controllabile. Per compiere questa operazione è possibile servirsi della matrice \mathbb{L} , infatti le righe di \mathbb{L} che presentano solo elementi unitari corrispondono agli stati a cui la rete è controllabile.

3.6 Controllo tramite retroazione dallo stato

Nella sezione precedente è stato affrontato il problema della stabilizzazione della rete e, in particolare, sono stati analizzati gli strumenti che consentono di capire quando una BCN è stabilizzabile. Per risolvere tale problema si è supposto che il vettore d'ingresso potesse essere scelto liberamente, con la sola restrizione che fosse un vettore logico.

In questa sezione viene affrontato il problema del controllo in retroazione e nello specifico viene indagato se e come un controllo di questo tipo sia in grado di stabilizzare la BCN.

Quando si parla di controllo in retroazione dallo stato si intende un controllo in cui il valore dell'ingresso $u(t)$ è combinazione lineare del valore dello stato $x(t)$, ovvero l'ingresso può essere espresso come

$$u(t) = K x(t), \quad (3.68)$$

dove se $x(t) \in \mathcal{L}_{2^n}$ e $u(t) \in \mathcal{L}_{2^m}$, allora $K \in \mathcal{L}_{2^m \times 2^n}$.

Il problema di stabilizzare la rete ad un ciclo tramite la retroazione descritta da (3.68) si riduce dunque al design della matrice K .

Supponendo che la BCN sia stabilizzabile al ciclo $\mathcal{C} = (x_{i_1}, x_{i_2}, \dots, x_{i_k})$, si vuole definire le colonne di K una ad una. Dalla stabilizzabilità della rete a \mathcal{C} sappiamo che per ogni x_{i_l} , $i_l \in \{i_1, i_2, \dots, i_k\}$, esiste un ingresso $u = \delta_{2^m}^j$ tale che $x_{i_{l+1}} = L \delta_{2^m}^j x_{i_l}$ ($x_{i_{k+1}} = x_{i_1}$), quindi imponendo che l'ingresso rispetti (3.68) si ottiene

$$u = \delta_{2^m}^j = K x_{i_l} = K \delta_{2^n}^{i_l}, \quad (3.69)$$

che corrisponde ad imporre che $[K]_{j,i_l} = 1$. Essendo K una matrice logica di dimensione $2^m \times 2^n$ risulta dunque che $\text{Col}_{i_l}(K) = \delta_{2^m}^j$. In questo modo, sfruttando la conoscenza degli stati appartenenti al ciclo, sono state definite k colonne di K . Per definire le $2^m - k$ colonne rimanenti risulta utile partizionare lo spazio di stato negli insiemi \mathcal{E}_τ , $\tau \in \mathbb{Z}_+$, tali che \mathcal{E}_τ indica l'insieme di tutti gli stati la cui distanza minima da \mathcal{C} è τ , dove con distanza minima si intende il minimo numero di passi necessari a raggiungere uno stato qualsiasi appartenente al ciclo. L'insieme \mathcal{E}_0 indica l'insieme contenente gli stati appartenenti al ciclo \mathcal{C} , mentre per ogni $\tau > 2^n - k$ risulta $\mathcal{E}_\tau = \emptyset$. Chiaramente essendo la rete controllabile a tutti gli stati di \mathcal{C} risulta che

$$\mathcal{E}_0 \cup \mathcal{E}_1 \cup \mathcal{E}_2 \cup \dots \cup \mathcal{E}_{2^n - k} = \mathcal{L}_{2^n} \quad (3.70)$$

e

$$\mathcal{E}_i \cap \mathcal{E}_j = \emptyset, \quad \forall i \neq j. \quad (3.71)$$

Dato che per ogni stato $x_i \in \mathcal{E}_{\tau+1}$ esiste un ingresso $u = \delta_{2^m}^h$ tale che $Lu x_i \in \mathcal{E}_\tau$, risulta evidente che imponendo $\text{Col}_i(K) = \delta_{2^m}^h$, $\forall x_i \in \mathcal{E}_1 \cup \dots \cup \mathcal{E}_{2^n - k}$, si fa in modo che ad ogni passo il controllo porti la rete verso il ciclo \mathcal{C} . Avendo assegnato tutte le 2^n colonne, la matrice K risulta completamente determinata e il controllo (3.68) permette di convergere a \mathcal{C} e di rimanervi.

Il metodo appena proposto per il calcolo di K funge da prova costruttiva al seguente teorema.

Teorema 3.6.1. *Se una Boolean control network descritta da (3.2) è stabilizzabile ad un ciclo $\mathcal{C} = (x_{i_1}, x_{i_2}, \dots, x_{i_k})$, allora è anche stabilizzabile tramite retroazione dallo stato.*

Dalla procedura descritta per il calcolo di K si deduce che la matrice K non è unica. Mentre tutte le colonne con indice $i_l \in \{i_1, i_2, \dots, i_k\}$ risultano determinate univocamente, le colonne rimanenti possono assumere valori differenti: basti pensare ad uno stato x_i appartenente a $\mathcal{E}_{\tau+1}$ il quale, in base al valore dell'ingresso, può transitare verso 2 o più stati appartenenti a \mathcal{E}_τ ; in questo caso è possibile assegnare 2 o più valori distinti alla i -esima colonna di K .

Esempio 3.6.1. Si consideri una BCN composta da tre nodi ($n = 3$) sulla quale agisce un ingresso ($m = 1$), la cui matrice di transizione è $L = [L_1, L_2]$, dove

$$\begin{aligned} L_1 &= \delta_8[2, 3, 4, 4, 6, 7, 8, 4], \\ L_2 &= \delta_8[2, 3, 4, 8, 6, 3, 8, 7]. \end{aligned} \quad (3.72)$$

Il grafo di stato è riportato in Figura 3.9. Un possibile ciclo della rete è $\mathcal{C} = (x_4, x_8)$: la rete risulta stabilizzabile a \mathcal{C} in quanto x_4 (e quindi x_8) è raggiungibile

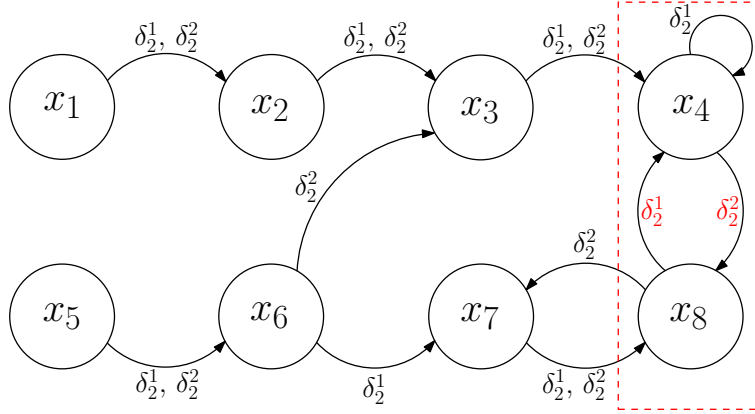


Figura 3.9: Grafo di stato della BCN dell'Esempio 3.6.1: in rosso il ciclo \mathcal{C} e gli ingressi necessari a mantenere la rete in \mathcal{C} .

da tutti gli altri stati e inoltre $x_4 = L_1 x_8$ e $x_8 = L_2 x_4$. Si desidera applicare un controllo nella forma $u(t) = K x(t)$, $K \in \mathcal{L}_{2 \times 8}$, per stabilizzare la rete a \mathcal{C} .

Il ciclo ha lunghezza 2, dunque è possibile determinare le 2 colonne di K associate agli stati appartenenti a \mathcal{C} . In particolare risulta $x_4 = L \delta_2^1 x_8$ e $x_8 = L \delta_2^2 x_4$ e quindi bisogna imporre $\text{Col}_8(K) = \delta_2^1$ e $\text{Col}_4(K) = \delta_2^2$; in questo modo si avrà che

$$Kx_4 = \delta_2^2, \quad Kx_8 = \delta_2^1. \quad (3.73)$$

Per ricavare le altre colonne di K è necessario definire gli insiemi \mathcal{E}_τ , $\tau \geq 1$. Naturalmente $\mathcal{E}_0 = \{x_4, x_8\}$, mentre per ottenere le informazioni riguardo al numero di passi minimo necessari a ciascuno stato per raggiungere \mathcal{C} è possibile analizzare il grafo di stato o ispezionare la matrice L_{tot}^τ . Seguendo quest'ultima strada si ha che

$$L_{tot} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.74)$$

ed analizzando la quarta e ottava riga di L_{tot} è possibile dedurre gli stati che raggiungono x_4 e x_8 in un passo. Prestando attenzione ad escludere gli stati appartenenti a \mathcal{E}_0 si ottiene $\mathcal{E}_1 = \{x_3, x_7\}$. Allo stesso modo analizzando la quarta e ottava riga di L_{tot}^2 e prestando attenzione a non includere gli stati appartenenti a \mathcal{E}_0 e \mathcal{E}_1 si ottiene $\mathcal{E}_2 = \{x_2, x_6\}$. Applicando lo stesso procedimento si ha che $\mathcal{E}_3 =$

$\{x_1, x_5\}$. Dal grafo di stato si può osservare che il passaggio dagli stati dell'insieme $\mathcal{E}_{\tau+1}$ agli stati dell'insieme \mathcal{E}_τ , $\tau = 0, 1, 2$, avviene indistintamente per $u = \delta_2^1$ o $u = \delta_2^2$, con conseguente arbitrarietà nella scelta di $\text{Col}_i(K)$, $i = 1, 2, 3, 5, 6, 7$. Esistono dunque diverse matrici di retroazione dallo stato che stabilizzano la BCN a \mathcal{C} e per la precisione la loro struttura è

$$K = \left[\star, \star, \star, \delta_2^2, \star, \star, \star, \delta_2^1 \right], \quad (3.75)$$

dove \star denota le colonne che possono essere sia δ_2^1 che δ_2^2 . Dato che sono presenti 6 colonne che possono avere 2 valori distinti, le matrici K che soddisfano la richiesta di stabilizzazione sono $2^6 = 64$. \triangle

Dal Teorema 3.6.1 discende il seguente risultato.

Corollario 3.6.2. *Se una Boolean control network descritta da (3.2) è stabilizzabile ad un punto di equilibrio x_i , allora è anche stabilizzabile tramite retroazione dallo stato.*

Si consideri ora l'equazione di aggiornamento dello stato della BCN sulla quale agisce un controllo in retroazione:

$$x(t+1) = L K x(t) x(t) = L K x^2(t). \quad (3.76)$$

Nel Capitolo 2 è stata introdotta la matrice di riduzione di potenza M_r , la quale permette di "ridurre" da 2 a 1 l'esponente di un vettore appartenente a \mathcal{L}_2 . Tramite l'utilizzo di M_r è possibile definire la matrice logica $\Phi_{2^h} \in \mathcal{L}_{2^{2^h} \times 2^h}$ che permette di ridurre alla stesso modo l'esponente di un vettore logico di dimensione 2^h , $h \in \mathbb{Z}_+$, ovvero $x^2 = \Phi_{2^h} x$, $x \in \mathcal{L}_{2^h}$. La matrice Φ_{2^h} è ricavabile come segue:

$$\Phi_{2^h} = \prod_{i=1}^{2^h} I_{2^{i-1}} \otimes \left[(I_2 \otimes W_{[2, 2^{2^h-i}]}) M_r \right], \quad (3.77)$$

dove W è una matrice swap. A questo punto è possibile riscrivere (3.76) come

$$\begin{aligned} x(t+1) &= L K x^2(t) \\ &= L K \Phi_{2^n} x(t) \\ &= \bar{L} x(t), \end{aligned} \quad (3.78)$$

dove $\bar{L} = L K \Phi_{2^n} \in \mathcal{L}_{2^n \times 2^n}$. Si può concludere quindi che una BCN stabilizzata ad un ciclo tramite retroazione può essere considerata una BN la cui matrice di transizione è \bar{L} .

Se K stabilizza la rete ad un ciclo allora (3.78) descrive una BN che presenta un

ciclo globalmente attrattivo; per quanto visto nella Sezione 3.3.1 è allora possibile trovare una matrice di permutazione Π tale che

$$\Pi^T L \Pi = \begin{bmatrix} N_1 & 0 \\ \star & C_1 \end{bmatrix}, \quad (3.79)$$

dove C_1 è una matrice ciclica e N_1 è una matrice nilpotente. Dalle considerazioni fatte segue la seguente proposizione.

Proposizione 3.6.3. *Una matrice K stabilizza ad un ciclo la BCN (3.2) tramite retroazione se e solo se la BN la cui matrice di transizione è $\bar{L} = L K \Phi_{2^n}$ ha un ciclo globalmente attrattivo o, equivalentemente, se e solo se esiste una matrice di permutazione Π tale che $\Pi^T L \Pi$ ha la forma (3.79).*

Esempio 3.6.2. Con riferimento alla BCN dell'Esempio 3.6.1, si consideri una nuova matrice di retroazione

$$K = \delta_2[1, 1, 1, 2, 1, 1, 1, 2], \quad (3.80)$$

che ha una struttura differente dalle matrici di retroazione che stabilizzano la BCN al ciclo $\mathcal{C} = (x_4, x_8)$ calcolate nell'esercizio precedente. Si desidera conoscere gli effetti di un controllo in retroazione fatto tramite K sulla rete.

Procedendo con i calcoli si ha che

$$\begin{aligned} \Phi_{2^3} &= \prod_{i=1}^{2^3} I_{2^{i-1}} \otimes \left[(I_2 \otimes W_{2,2^{2^3-i}}) M_r \right] \\ &= \delta_{64}[1, 10, 19, 28, 37, 46, 55, 64] \end{aligned} \quad (3.81)$$

e

$$\begin{aligned} \bar{L} &= L K \Phi_{2^3} \\ &= \delta_8[2, 3, 4, 8, 6, 7, 8, 7]. \end{aligned} \quad (3.82)$$

Si ottiene così la BN descritta da $x(t+1) = \bar{L}x(t)$, il cui grafo di stato è rappresentato in Figura 3.10. Analizzando il grafo si può notare la presenza di un ciclo $\bar{\mathcal{C}} = (x_7, x_8)$ che è raggiungibile da tutto il sistema e dunque è globalmente attrattivo. Si poteva arrivare alla stessa conclusione notando che \bar{L} presenta già la forma (3.79).

Dai risultati ottenuti si può infine affermare che il controllo in retroazione dallo stato eseguito mediante K stabilizza la BCN al ciclo $\bar{\mathcal{C}}$.

△

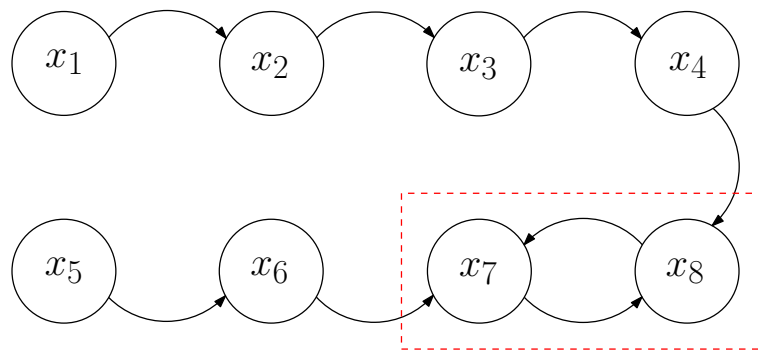


Figura 3.10: Grafo di stato della BN dell'Esempio 3.6.2: nel riquadro rosso il ciclo globalmente attrattivo \bar{C} .

Capitolo 4

Applicazione alla teoria dei giochi

La Teoria dei giochi è uno di quei campi che risulta particolarmente adatto ad essere modellato (e quindi studiato) mediante le Boolean control networks. Tale teoria è una disciplina che studia, mediante modelli matematici, l'interazione strategica tra dei decisori razionali, che possono essere persone, aziende, computer ecc. I decisori vengono chiamati giocatori e ciascun giocatore può compiere un'azione che viene chiamata strategia. Le strategie che ogni giocatore può adottare sono in un numero ben definito e ciò fa sì che un gioco possa essere modellato mediante Boolean networks nel caso le strategie siano due o, più in generale, mediante k -valued networks nel caso in cui le strategie siano k . In questo capitolo verrà dapprima definito il concetto di k -valued network, verranno poi introdotte le nozioni fondamentali della Teoria dei giochi ed infine verranno modellati e studiati dei particolari tipi di giochi usufruendo di tutti gli argomenti e gli strumenti approfonditi nei capitoli precedenti.

4.1 K -valued networks

Nelle sezioni successive vedremo che nell'analisi di particolari tipi di giochi, detti Networked Evolutionary Games (o semplicemente NEG), è possibile modellare i giocatori come nodi di una rete che assumono un valore diverso in base alla strategia adottata. Nel caso in cui ciascun giocatore possa adottare solo 2 mosse (che possiamo pensare essere rappresentate dai valori 0 o 1), risulta naturale modellare la dinamica del gioco tramite una Boolean Network, ma che cambiamenti avvengono nel caso in cui le mosse disponibili siano più di due? Come vedremo nel corso di questo paragrafo i cambiamenti sono marginali e permettono di applicare tutti gli strumenti analizzati nei capitoli precedenti.

4.1.1 Sistemi di equazioni logiche a k valori

Dato l'insieme \mathcal{D}_k contenete tutti i numeri interi da 1 a k ovvero $\mathcal{D}_k = \{1, 2, \dots, k\}$, risulta possibile associare ad ogni variabile $\chi \in \mathcal{D}_k$ un vettore logico $x \in \mathcal{L}_k$, in particolare per ogni $i = 1, 2, \dots, k$ si ha

$$\chi = i \sim \delta_k^i = x. \quad (4.1)$$

Una funzione logica a k valori nelle variabili $\chi_1, \chi_2, \dots, \chi_n \in \mathcal{D}_k$ è una mappa $f : \mathcal{D}_k^n \rightarrow \mathcal{D}_k$. Anche in questo caso è possibile rappresentare una funzione logica a k valori in forma algebrica mediante l'utilizzo di matrici di struttura:

$$f(\chi_1, \chi_2, \dots, \chi_n) = M_f x, \quad (4.2)$$

dove $x = \times_{i=1}^n x_i \in \mathcal{L}_{k^n}$, $x_i \in \mathcal{L}_k$, $i = 1, \dots, n$, sono i vettori logici associati alle variabili $\chi_i \in \mathcal{D}_k$ e $M_f \in \mathcal{L}_{k \times k^n}$ è la matrice di struttura di f .

Si consideri ora il sistema

$$\begin{cases} f_1(\chi_1, \chi_2, \dots, \chi_n) = \beta_1 \\ f_2(\chi_1, \chi_2, \dots, \chi_n) = \beta_2 \\ \vdots \\ f_m(\chi_1, \chi_2, \dots, \chi_n) = \beta_m, \end{cases} \quad (4.3)$$

dove f_j , $j = 1, 2, \dots, m$ sono equazioni logiche a k valori, $\chi_i \in \mathcal{D}_k$, $i = 1, \dots, n$, sono variabili e $\beta_j \in \mathcal{D}_k$, $j = 1, \dots, m$, sono costanti. Un sistema di questo tipo viene definito sistema di equazioni logiche a k valori. Associando alle variabili χ_i , $i = 1, \dots, n$, e alle costanti β_j , $j = 1, \dots, m$, i rispettivi vettori logici x_i , $b_j \in \mathcal{L}_k$, anche per un sistema di questo tipo è possibile ottenere la seguente rappresentazione mediante equazioni algebriche

$$\begin{cases} M_{f_1} x = b_1 \\ M_{f_2} x = b_2 \\ \vdots \\ M_{f_m} x = b_m, \end{cases} \quad (4.4)$$

dove $x = \times_{i=1}^n x_i$ e $M_{f_j} \in \mathcal{L}_{k \times k^n}$, $j = 1, \dots, m$, sono le matrici di struttura delle funzioni f_j .

Utilizzando il prodotto di Khatri-Rao si ottiene infine la rappresentazione del sistema (4.3) tramite un'unica equazione algebrica:

$$L x = b, \quad (4.5)$$

dove $b = \times_{j=1}^m b_j$ ed $L = M_{f_1} * M_{f_2} * \dots * M_{f_m}$ è una matrice logica di dimensione $k^m \times k^n$, ovvero $L \in \mathcal{L}_{k^m \times k^n}$.

4.1.2 Reti logiche a k valori

Una rete logica a k valori, o k -valued network, consiste in un insieme di variabili (o nodi) $\chi_1, \chi_2, \dots, \chi_n \in \mathcal{D}_k$, che interagiscono tra loro. L'evoluzione della rete è determinata dal seguente sistema di equazioni

$$\begin{cases} \chi_1(t+1) = f_1(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \chi_2(t+1) = f_2(\chi_1(t), \chi_2(t), \dots, \chi_n(t)) \\ \vdots \\ \chi_n(t+1) = f_n(\chi_1(t), \chi_2(t), \dots, \chi_n(t)), \end{cases} \quad (4.6)$$

dove le f_i , $i = 1, 2, \dots, n$, sono funzioni logiche a k valori. Per quanto visto nel precedente paragrafo, la dinamica della rete può essere espressa mediante la seguente equazione algebrica:

$$x(t+1) = Lx(t), \quad (4.7)$$

dove $x(t) = \times_{i=1}^n x_i \in \mathcal{L}_{k^n}$ è detto vettore di stato della rete al tempo t e $L \in \mathcal{L}_{k^n \times k^n}$ è la matrice di transizione della rete.

Nel caso nella rete siano presenti degli ingressi appartenenti a \mathcal{D}_k si parla di rete logica a k valori controllata o k -valued control network, e l'evoluzione della rete è rappresentato dal sistema di equazioni logiche a k valori

$$\begin{cases} \chi_1(t+1) = f_1(\chi_1(t), \chi_2(t), \dots, \chi_n(t), \mu_1(t), \mu_2(t), \dots, \mu_m(t)) \\ \chi_2(t+1) = f_2(\chi_1(t), \chi_2(t), \dots, \chi_n(t), \mu_1(t), \mu_2(t), \dots, \mu_m(t)) \\ \vdots \\ \chi_n(t+1) = f_n(\chi_1(t), \chi_2(t), \dots, \chi_n(t), \mu_1(t), \mu_2(t), \dots, \mu_m(t)), \end{cases} \quad (4.8)$$

dove $\mu_1, \mu_2, \dots, \mu_m \in \mathcal{D}_k$ sono gli ingressi della rete. Anche in questo caso è possibile ottenere la rappresentazione tramite equazione algebrica, ovvero

$$x(t+1) = Lu(t)x(t), \quad (4.9)$$

dove $L \in \mathcal{L}_{k^n \times k^{n+m}}$ e $u(t) \in \mathcal{L}_{k^m}$ è il vettore di ingresso ricavato dalle versioni vettoriali degli ingressi $\mu_1(t), \mu_2(t), \dots, \mu_m(t)$. Dividendo la matrice L in k^m blocchi di dimensione $k^n \times k^n$, ovvero $L = [L_1, L_2, \dots, L_{k^m}]$, è possibile considerare la BCN come un Boolean switched system descritto dalla seguente espressione:

$$x(t+1) = L_j x(t) \quad (4.10)$$

dove L_j è il blocco di L selezionato quando $u(t) = \delta_{k^m}^j$, $j = 1, 2, \dots, k^m$.

Dall'analisi fatta è immediato constatare che le differenze tra reti Booleane e reti

logiche a k valori sono minime; le BNs possono infatti essere considerate un caso particolare delle k -valued networks per $k = 2$. Nella forma algebrica l'unica cosa che le distingue è la dimensione di matrici e vettori che descrivono la loro dinamica, ma trattandosi sempre di matrici logiche e vettori logici l'analisi delle reti resta immutata. Tutte le proprietà studiate per BNs e BCNs si mantengono anche per le reti a k -valori, controllate o non. Una differenza apprezzabile è che lo stato della rete passa da poter assumere valori in \mathcal{L}_{2^n} per le BNs a poter assumere valori in \mathcal{L}_{k^n} per le k -valued networks (nel caso nelle reti siano presenti n nodi); di conseguenza nella rappresentazione tramite grafo di stato le k -valued networks saranno composte da k^n nodi. L'unica vera differenza tra i due tipi di rete è che gli operatori su cui si basano le funzioni logiche a k valori che regolano la dinamica della rete delle k -valued networks naturalmente differiscono dagli operatori su cui si basano le funzioni logiche (i.e NOT, OR, AND), dovendo i primi operare su \mathcal{D}_k e gli ultimi su \mathcal{B} .

Si conclude riportando un breve esempio di k -valued network il cui studio viene effettuato mediante gli stessi strumenti utilizzati per lo studio delle BNs.

Esempio 4.1.1. Si consideri una k -valued network composta da 2 nodi ($n = 2$) che possono assumere valori in $\mathcal{D}_3 = \{1, 2, 3\}$. La dinamica della rete è descritta nella sua forma algebrica da

$$\begin{aligned} x(t+1) &= Lx(t) \\ &= \delta_9[4, 3, 6, 7, 5, 5, 8, 7, 6]x(t) \end{aligned} \quad (4.11)$$

dove $x(t)$ assume valori in \mathcal{L}_9 . Il grafo di stato delle rete è riportato in Figura 4.1.

Essendo $\text{tr}(L) = 1$ si deduce che esiste un unico punto fisso ($N_1 = 1$) e dato che $[L]_{5,5} = 1$ tale punto è $x_5 = \delta_9^5$. Per la ricerca di un ciclo di lunghezza 2 si calcola la matrice $L^2 = \delta_9[7, 6, 5, 8, 5, 5, 7, 8, 5]$ ed il numero di cicli di lunghezza 2 risulta essere

$$N_2 = \frac{\text{tr}(L^2) - \sum_{i \in \mathcal{P}(2)} i N_i}{2} = \frac{3 - (1 \cdot 1)}{2} = 1. \quad (4.12)$$

Dato che $[L]_{5,5} = 1$, $[L]_{7,7} = 1$ e $[L]_{8,8} = 1$ e x_5 è un punto fisso, il ciclo è composto dagli stati x_7 e x_8 . Procedendo allo stesso modo si ottiene che non esistono cicli di lunghezza superiore. Si conclude che la rete non è stabile e non presenta un ciclo globalmente attrattivo essendo presenti sia un punto fisso che un ciclo. \triangle

4.2 Breve Introduzione alla teoria dei giochi

La Teoria dei Giochi è la disciplina che studia come i “giocatori” si comportano dal punto di vista strategico, ovvero analizza come diversi giocatori interagiscono

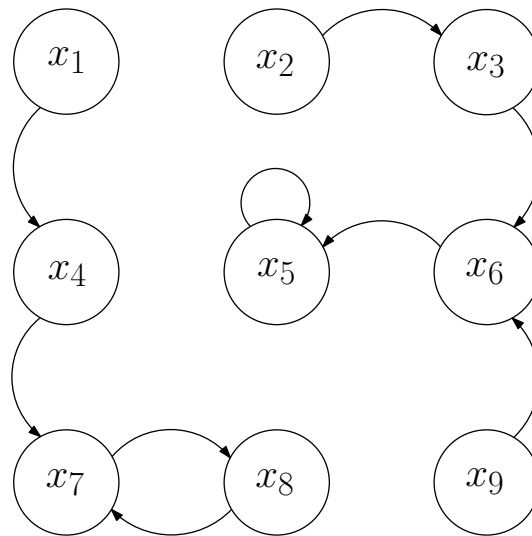


Figura 4.1: Grafo di stato della k -valued network dell'Esempio 4.1.1.

avendo degli obiettivi che possono essere comuni, diversi o conflittuali. Si tratta di una materia che trova applicazione in svariati campi, tra cui il campo economico e finanziario, il campo ingegneristico o sociologico e la politica. Lo scopo della Teoria dei Giochi è sia spiegare in termini matematici perché i giocatori (o decisori) utilizzino determinate strategie in determinate situazioni, sia prevedere quali casistiche possono generarsi dall'interazione dei giocatori.

L'ipotesi che sta alla base della Teoria dei Giochi è l'ipotesi di razionalità secondo la quale i decisori sono supposti essere intelligenti e razionali. Un giocatore intelligente ha la capacità logica di riconoscere le azioni necessarie a massimizzare il proprio guadagno mentre un giocatore razionale è in grado di ordinare mediante un insieme di assiomi le sue preferenze su un insieme di risultati.

I giochi possono essere suddivisi in varie tipologie in base al tipo di iterazione tra i giocatori o in base al tipo di vincite che possono realizzarsi, ma i parametri su cui basarsi per la categorizzazione dei giochi possono essere molteplici. Di seguito vengono elencate le tipologie più usate per descrivere un gioco.

- **Giochi cooperativi e non cooperativi:** se i giocatori non hanno interesse ad opporsi l'uno all'altro e stipulano accordi vincolanti tra loro si parla di gioco cooperativo, in caso contrario si parla di gioco non cooperativo.
- **Giochi simultanei e sequenziali:** se i giocatori applicano le loro strategie simultaneamente, o comunque l'ultimo giocatore del turno non conosce le azioni effettuate da tutti gli altri giocatori in quel turno, si parla di gioco simultaneo. In un gioco sequenziale invece i giocatori possono avere delle

informazioni sulle mosse eseguite dagli altri giocatori nello stesso turno. Il più classico esempio di gioco simultaneo è la morra cinese.

- **Giochi ad informazione completa e perfetta:** nel caso in cui i giocatori siano a conoscenza delle possibili strategie adottabili da ogni giocatore (e del guadagno ricavabile da esse) si parla di gioco ad informazione completa. Quando ogni giocatore è a conoscenza delle mosse precedentemente compiute da tutti gli altri giocatori, il gioco viene detto ad informazione perfetta. Da come sono stati descritti giochi simultanei e sequenziali si deduce che solo gli ultimi possono essere ad informazione perfetta.
- **Giochi a somma zero:** nel caso al termine di ogni turno la somma delle vincite di tutti i giocatori sia pari a zero si parla di gioco a somma zero, in caso contrario di gioco a somma non zero. Si deduce che nei giochi a somma zero ad ogni guadagno positivo di un giocatore deve corrispondere la perdita (guadagno negativo) di almeno un altro giocatore.

Nel seguito verranno trattati solo giochi non cooperativi, simultanei, ad informazione completa ma non perfetta e a somma zero o non.

4.2.1 Definizione e rappresentazione di gioco

Un gioco strategico è costituito da tre componenti fondamentali:

1. n giocatori indicati dall'insieme $N = \{1, 2, \dots, n\}$;
2. l'insieme dei profili $S = \prod_{i=1}^n S_i$, dove $S_i = \{1, 2, \dots, k_i\}$ rappresenta le strategie adottabili dal i -esimo giocatore e \prod è il prodotto cartesiano.;
3. l'insieme delle funzioni di payoff $C = \{c_1, c_2, \dots, c_n\}$, dove $c_i : S \rightarrow \mathbb{R}$ è la funzione di payoff dell' i -esimo giocatore che associa ad ogni possibile combinazione delle strategie adottate dai giocatori un valore reale che viene detto payoff o pagamento del giocatore i .

Un gioco strategico può essere dunque espresso mediante la terna $G = (N, S, C)$. In questo capitolo verranno presi in considerazione solamente giochi in cui le strategie adottabili sono le stesse per ogni giocatore, ovvero $S_i = S_j, \forall j \neq i, i, j = 1, 2, \dots, n$. Tra i giochi che rispettano questa condizione, una particolare importanza è rivestita dai Fundamental Network Games.

Definizione 4.1. Un gioco strategico composto da due giocatori ($N = \{1, 2\}$) in cui

$$S_1 = S_2 := S_0 = \{1, 2, \dots, k\} \quad (4.13)$$

è detto *Fundamental Network Game* (FNG). Se è rispettata la condizione

$$c_1(x_1, x_2) = c_2(x_2, x_1), \quad \forall x_1, x_2 \in S_0, \quad (4.14)$$

allora il FNG è simmetrico.

La rappresentazione più usata per descrivere i FNGs è tramite una tabella, detta payoff bi-matrix, i cui elementi sono il codominio delle funzioni di payoff dei 2 giocatori. La payoff bi-matrix di un FNG ha la struttura riportata in Tabella 4.1, nella quale P_1 e P_2 sono i giocatori le cui funzioni di payoff sono rispettivamente c_1 e

$P_1 \setminus P_2$	1	2	\dots	k
1	$c_1(1, 1), c_2(1, 1)$	$c_1(1, 2), c_2(1, 2)$	\dots	$c_1(1, k), c_2(1, k)$
2	$c_1(2, 1), c_2(2, 1)$	$c_1(2, 2), c_2(2, 2)$	\dots	$c_1(2, k), c_2(2, k)$
\vdots	\vdots	\vdots	\ddots	\vdots
k	$c_1(k, 1), c_2(k, 1)$	$c_1(k, 2), c_2(k, 2)$	\dots	$c_1(k, k), c_2(k, k)$

Tabella 4.1: Payoff bi-matrix di un FNG

c_2 e l'insieme delle strategie di ciascun giocatore è $S_0 = \{1, 2, \dots, k\}$. Il vantaggio di questa rappresentazione sta nel riuscire a sintetizzare in un'unica tabella tutte le informazioni del gioco, ma ovviamente non è applicabile nel caso siano presenti più di due giocatori.

Per dare una rappresentazione anche nel caso siano presenti più di due giocatori si consideri allora $G = (N, S, C)$ in cui $N = \{1, 2, \dots, n\}$, $S_i = \{1, 2, \dots, k\}$, $i = 1, 2, \dots, n$ e $C = \{c_1, c_2, \dots, c_n\}$. Si ordinino i profili $s^j \in S$, $j = 1, 2, \dots, k^n$ in ordine alfabetico, ovvero:

$$\begin{aligned} s^1 &= (1, 1, \dots, 1, 1); & s^2 &= (1, 1, \dots, 1, 2); & \dots; \\ \dots; & & s^{k^n-1} &= (k, k, \dots, k, k-1); & s^{k^n} &= (k, k, \dots, k, k). \end{aligned}$$

Ad ogni profilo s^j si associa il vettore colonna C_j i cui elementi sono le funzioni di payoff di ciascun giocatore per il profilo s^j , ovvero:

$$C_j = [c_1(s^j), c_2(s^j), \dots, c_n(s^j)]^T, \quad j = 1, 2, \dots, k^n. \quad (4.15)$$

La matrice $P_G \in \mathbb{R}^{n \times k^n}$ è detta *matrice di payoff* di G se

$$Col_j(P_G) = C_j, \quad j = 1, 2, \dots, k^n. \quad (4.16)$$

La matrice di payoff non è sintetica quanto la payoff bi-matrix, ma è applicabile al caso generale in cui siano presenti $n \geq 2$ giocatori ed esprime tutta l'informazione necessaria a comprendere il gioco G . La matrice P_G può essere riportata da sola oppure essere incorporata in una tabella per migliorarne la leggibilità, ma in entrambi i casi si parla di matrice di payoff.

4.2.2 Esempi di gioco strategico

In questo paragrafo vengono riportati alcuni dei più classici esempi di gioco non cooperativo e simultaneo; per ciascun gioco vengono riportate sia la payoff bi-matrix che la matrice di payoff.

Esempio 4.2.1 (La morra cinese). Ciascun giocatore può scegliere la propria strategia dall'insieme $S_0 = \{1, 2, 3\}$, dove 1 indica la mossa “sasso”, 2 la mossa “carta” e 3 la mossa “forbice”. Le famose regole del gioco dicono che sasso batte forbice, forbice batte carta e carta batte sasso, mentre si verifica un pareggio nel caso in cui venga utilizzata la stessa mossa da entrambi i giocatori. Per costruire una funzione di payoff è possibile, ad esempio, attribuire il valore 1 alla vittoria, il valore 0 al pareggio ed il valore -1 alla sconfitta. In Tabella 4.2 e in Tabella 4.3 sono riportate rispettivamente la payoff bi-matrix e la matrice di payoff nel caso ci siano due giocatori P_1 e P_2 . Si noti che il gioco della morra cinese a due giocatori

$P_1 \setminus P_2$	1	2	3
1	(0, 0)	(-1, 1)	(1, -1)
2	(1, -1)	(0, 0)	(-1, 1)
3	(-1, 1)	(1, -1)	(0, 0)

Tabella 4.2: Payoff bi-matrix del gioco della morra cinese.

Profilo	11	12	13	21	22	23	31	32	33
c_1	0	-1	1	1	0	-1	-1	1	0
c_2	0	1	-1	-1	0	1	1	-1	0

Tabella 4.3: Matrice di payoff del gioco della morra cinese.

è un FNG simmetrico. △

Esempio 4.2.2 (Il dilemma del prigioniero). Due persone (i giocatori) sono accusate di aver commesso un reato e ognuno può scegliere la propria strategia dall'insieme $S_0 = \{1, 2\}$, dove 1 significa “confessare” e 2 significa “non confessare”. La scelta di ciascuno dei due influenza anche il destino dell'altro, infatti se solo uno dei due confessa, accusando l'altro, chi ha confessato evita la pena mentre l'altro viene condannato a 7 anni di carcere, se entrambi confessano, tutti e due sono condannati a 5 anni mentre se entrambi decidono di non confessare a ciascuno viene dato 1 anno di prigione. Il gioco è schematizzato tramite payoff bi-matrix e matrice di payoff rispettivamente in Tabella 4.4 e in Tabella 4.5. Dato che i due

$P_1 \setminus P_2$	1	2
1	$(-5, -5)$	$(0, -7)$
2	$(-7, 0)$	$(-1, -1)$

Tabella 4.4: Payoff bi-matrix del gioco Il dilemma del prigioniero.

Profilo	11	12	21	22
c_1	5	0	7	1
c_2	5	7	0	1

Tabella 4.5: Matrice di payoff del gioco Il dilemma del prigioniero.

giocatori sono razionali e non possono cooperare, entrambi sono portati a confessare, anche se, potendo collaborare, è evidente che la scelta migliore diverrebbe quella di non confessare. \triangle

Esempio 4.2.3 (Il gioco di Benoit-Krishna). Si tratta di una versione modificata del dilemma del prigioniero. È stata introdotta da J. P. Benoit e V. Krishna per analizzare il comportamento dei giochi ripetuti un numero finito di volte, argomento su cui focalizzeremo l'attenzione nella prossima sezione. In questo caso le strategie adottabili dai due prigionieri sono $S_0 = \{1, 2, 3\}$, in cui 1 sta per “negare”, 2 significa “attendere” (o non far niente) e 3 significa “confessare”. In questo caso i prigionieri sono già stati condannati e attraverso un nuovo interrogatorio (il gioco) devono decidere che strategia adottare per avere uno sconto di pena che corrisponde al payoff nelle varie situazioni; uno sconto di pena negativo corrisponde ad un aumento degli anni di carcere da scontare. Il gioco è schematizzato tramite payoff bi-matrix e matrice di payoff rispettivamente in Tabella 4.6 e in Tabella 4.7. \triangle

$P_1 \setminus P_2$	1	2	3
1	$(10, 10)$	$(-1, -12)$	$(-1, 15)$
2	$(-12, -1)$	$(8, 8)$	$(-1, -1)$
3	$(15, -1)$	$(-1, -1)$	$(0, 0)$

Tabella 4.6: Payoff bi-matrix del gioco di Benoit-Krishna.

Esempio 4.2.4 (Il gioco dei maiali nel recinto). Ci sono due maiali in un recinto, uno grande ed uno piccolo. Ad un'estremità del recinto è posta una mangiatoia

Profilo	11	12	13	21	22	23	31	32	33
c_1	10	-1	-1	-12	8	-1	15	-1	0
c_2	10	-12	15	-1	8	-1	-1	-1	0

Tabella 4.7: Matrice di payoff del gioco di Benoit-Krishna.

il cui rifornimento di foraggio è controllato tramite una pedana posta all'altra estremità del recinto. Se il maiale piccolo sale sulla pedana ed il maiale grande attende, quest'ultimo può mangiare una quantità di cibo pari ad 7, mentre non ne rimane per il maiale piccolo. Se il maiale grande sale sulla pedana e quello piccolo attende, quest'ultimo può mangiare una quantità di cibo pari a 6, mentre quello grande ha a disposizione 1 unità di cibo. Se entrambi i due maiali salgono sulla pedana, il maiale grande può mangiare 5 unità di cibo mentre a quello piccolo ne rimangono 2 unità. Se nessun maiale sale sulla pedana non viene erogato cibo. Data l'asimmetria del gioco, in questo caso è necessario specificare l'identità dei giocatori: con P_1 indichiamo il maiale piccolo e con P_2 quello grande. L'insieme delle strategie adottabili è $S_0 = \{1, 2\}$ in cui 1 significa "salire sulla pedana" e 2 significa "aspettare". In Tabella 4.8 e in Tabella 4.9 sono riportate rispettivamente la payoff bi-matrix e la matrice di payoff del gioco. \triangle

$P_1 \setminus P_2$	1	2
1	(2, 5)	(0, 7)
2	(6, 1)	(0, 0)

Tabella 4.8: Payoff bi-matrix del gioco dei maiali nel recinto.

Profilo	11	12	21	22
c_1	2	0	6	0
c_2	5	7	1	0

Tabella 4.9: Matrice di payoff del gioco dei maiali nel recinto.

Osservazione 11. Negli esempi esposti sono sempre stati forniti dei valori per i payoff di ciascun giocatore. Per un'analisi generale è possibile mantenere delle variabili al posto di quei valori, in modo da studiare la dinamica del gioco al variare di queste variabili.

4.2.3 Equilibrio di Nash

L'equilibrio di Nash è il concetto di soluzione più significativo nei giochi non cooperativi: tale equilibrio definisce uno stato stazionario del gioco, in cui nessun giocatore è interessato a cambiare strategia.

Definizione 4.2. Indicando con s_i la strategia adottata dal i -esimo giocatore, un profilo $\bar{s} = (\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n) \in S$ è un equilibrio di Nash se per ogni giocatore i si ha

$$c_i(\bar{s}_i, \bar{s}^{-i}) \geq c_i(s_i, \bar{s}^{-i}), \quad \forall s_i \in S_i, \quad (4.17)$$

dove \bar{s}^{-i} indica il profilo contenente le strategie degli $n - 1$ giocatori escluso i .

Quando si presenta un equilibrio di Nash ogni giocatore sta applicando la strategia che gli fornisce il maggior payoff considerando il profilo di tutti gli altri giocatori e dunque non ha interesse a modificare la strategia attuale.

In un gioco possono esistere uno o più equilibri di Nash, così come può non esistere alcuno.

Esempio 4.2.5. Nel gioco della morra cinese a due giocatori esposto nell'Esempio 4.2.1 la condizione (4.17) non si verifica per nessun profilo, dunque non è presente nessun equilibrio di Nash.

Nel dilemma del prigioniero si è già accennato al fatto che entrambi i giocatori sono portati a confessare, ovvero a scegliere la strategia 1: nello specifico, al giocatore P_1 conviene confessare indipendentemente dalla scelta di P_2 e lo stesso accade per il giocatore P_2 . Si ha dunque che il profilo $\bar{s} = (1, 1)$ è un equilibrio di Nash, infatti per si ha che

$$c_1(1, 1) = c_2(1, 1) \geq c_i(s_i, 1), \quad \forall s_i \in S_0 = \{1, 2\}. \quad (4.18)$$

Nel gioco dei maiali nel recinto illustrato nell'Esempio 4.2.4 è facile notare la presenza di due equilibri di Nash, uno è costituito dal profilo $\bar{s}^1 = (1, 2)$ e l'altro dal profilo $\bar{s}^2 = (2, 1)$. △

4.3 Networked Evolutionary Games

Uno degli interessi principali nella Teoria dei giochi è l'analisi della dinamica di un gioco quando questo viene ripetuto più volte; a questo proposito ricoprono una particolare importanza i Networked Evolutionary Games che, come deducibile dal nome, trattano l'evoluzione nel tempo (o turni) di un gioco. Un Networked Evolutionary Game (NEG) è composto da tre ingredienti fondamentali. Il primo è un FNG, la cui funzione è descrivere il gioco che intercorre tra due giocatori.

La seconda componente di un NEG è un grafo che descrive topologicamente l'interazione tra i giocatori. Il concetto di grafo è già stato trattato, tuttavia risultano utili le seguenti definizioni che vanno a descrivere più approfonditamente determinate caratteristiche dei grafi.

Definizione 4.3. Un grafo è detto omogeneo se si verifica una delle seguenti condizioni:

- è un grafo orientato e tutti i suoi nodi hanno lo stesso grado entrante e lo stesso grado uscente;
- è un grafo non orientato e tutti i suoi nodi hanno lo stesso grado.

Un grafo non omogeneo è chiamato grafo eterogeneo.

Definizione 4.4. Dato un grafo $\Sigma = \{\mathcal{V}, \mathcal{E}\}$, dove \mathcal{V} è l'insieme dei vertici ed \mathcal{E} è l'insieme degli archi, si definiscono:

1. insieme dei vicini di \mathcal{V}_i e viene indicato con $U(i)$, l'insieme di tutti i nodi $\mathcal{V}_j \in \mathcal{V}$ tali che $(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E}$ o $(\mathcal{V}_j, \mathcal{V}_i) \in \mathcal{E}$. Si assume che $\mathcal{V}_i \in U(i)$;
2. insieme dei vicini in r -passi di \mathcal{V}_i e viene indicato con $U_r(i)$ l'insieme di tutti i nodi $\mathcal{V}_j \in \mathcal{V}$ per i quali esiste un cammino di lunghezza minore o uguale ad r che parte da \mathcal{V}_i ed arriva in \mathcal{V}_j .

In un NEG i nodi di un grafo rappresentano i giocatori e dunque attraverso il grafo è possibile conoscere con chi gioca ciascun giocatore. Due grafi che verranno spesso utilizzati nel seguito sono il ciclo elementare a n nodi, indicato con O_n e la linea a n nodi, indicata con R_n . Due rappresentazioni di O_n orientato e R_n non orientato per $n = 4$ sono raffigurate in Figura 4.2. Si presti attenzione al fatto che O_n e R_∞ sono grafi omogenei, mentre R_n è eterogeneo.

L'ultima componente di un NEG è una legge di aggiornamento delle strategie. Lo scopo di una legge di aggiornamento delle strategie è quello di stabilire con quale criterio un giocatore sceglie la strategia da adottare al turno successivo sulla base delle strategie adottate nei turni precedenti o degli esiti dei turni precedenti. Per ricondursi alla simbologia usata per descrivere reti Booleane e k -valued networks, nel seguito la strategia adottata dal i -esimo giocatore al tempo t verrà indicata con $\chi_i(t)$; dato che vengono considerati solo giochi in cui le strategie adottabili sono le stesse per tutti i giocatori, ovvero $S_i = S_0$, $i = 1, \dots, n$, allora $\chi_i(t)$ può assumere solo valori appartenenti a S_0 . In un gioco con k strategie si ha che $S_0 = \mathcal{D}_k$, dunque è possibile esprimere $\chi_i(t)$ tramite il vettore $x_i(t)$ che assume valori in \mathcal{L}_k . Fin'ora si è sottinteso che un giocatore scelga una ben precisa strategia tra quelle a disposizione, ovvero scelga una strategia pura. Esiste anche la possibilità che

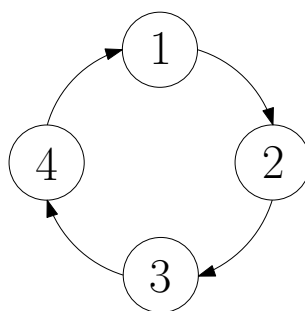


Figura 4.2: In alto un grafo orientato la cui struttura è O_4 , in basso un grafo non orientato la cui struttura è R_4 .

i giocatori possano adottare delle strategie miste: con questo termine si intende una distribuzione di probabilità sull'insieme delle strategie pure. In questo caso il vettore $x_i(t)$ diventa un vettore probabilistico la cui j -esima componente indica la probabilità che il giocatore i scelga la strategia $j \in S_0$. In questa tesi tuttavia verrà trattato solo il caso in cui i giocatori possono scegliere strategie pure.

Definizione 4.5. Una legge di aggiornamento delle strategie di un NEG, indicata con Φ , è descritta dal seguente insieme di funzioni:

$$\begin{aligned} x_1(t+1) &= f_1(\{x_j(t), c_j(t) | j \in U(1)\}), \\ x_2(t+1) &= f_2(\{x_j(t), c_j(t) | j \in U(2)\}), \\ &\vdots \\ x_n(t+1) &= f_n(\{x_j(t), c_j(t) | j \in U(n)\}), \end{aligned} \tag{4.19}$$

dove $f_i : \mathcal{L}_k^{|U(i)|} \times \mathbb{R}^{|U(i)|} \rightarrow \mathcal{L}_k$, $i = 1, 2, \dots, n$, $x_i(t) \in \mathcal{L}_k$, n è il numero dei giocatori e k è il numero delle strategie adottabili dai giocatori. Nel caso in cui il grafo del NEG sia omogeneo, tutte le f_i , $i = 1, \dots, n$, sono uguali.

Osservazione 12. Nel caso generale le f_i possono essere delle mappe probabilistiche, il che porterebbe ad avere come risultato la possibile adozione di strategie miste da parte dei giocatori. In questo elaborato tuttavia vengono considerate f_i deterministiche in modo da ottenere solamente strategie pure. Questa semplificazione permette di utilizzare gli strumenti introdotti nei capitoli precedenti.

Ciò che esprime la definizione di legge di aggiornamento delle strategie è che la strategia di ciascun giocatore al tempo $t + 1$ dipende dalle strategie e dai payoff dei giocatori vicini (incluso se stesso) al tempo t . Esistono svariate leggi di aggiornamento dello stato, sia deterministiche che probabilistiche, che vengono scelte in base al contesto in cui vengono applicate. In questo capitolo verranno applicate due delle leggi di aggiornamento delle strategie tra le più comunemente utilizzate in campo deterministico, ovvero l'Imitazione incondizionata a priorità fissa e la Risposta ottima (myopic best response).

1. **Imitazione incondizionata a priorità fissa:** la strategia del giocatore i al tempo $t + 1$ è scelta selezionando la miglior strategia (la strategia che ha ottenuto il payoff maggiore) tra le strategie scelte dai propri vicini al tempo t , ovvero se

$$\bar{j} = \operatorname{argmax}_{j \in U(i)} c_j(x(t)), \quad (4.20)$$

allora

$$x_i(t + 1) = x_{\bar{j}}(t). \quad (4.21)$$

Se il miglior payoff non è unico è possibile sceglierne uno in base ad una priorità prestabilita, ad esempio:

$$\bar{j} = \min\{d : d \in \operatorname{argmax}_{j \in U(i)} c_j(x(t))\}. \quad (4.22)$$

2. **Risposta ottima:** il giocatore i assume che al tempo $t + 1$ i suoi vicini scelgano la stessa strategia che hanno utilizzato al tempo t . Basandosi su questa assunzione il giocatore i sceglie la propria strategia in modo da massimizzare il proprio payoff. In particolare si ha che

$$x_i(t + 1) = \operatorname{argmax}_{s_j \in S_0} c_i(s_j, s^{-i}(t)), \quad (4.23)$$

dove $s^{-i}(t)$ indica il profilo dei giocatori appartenenti a $U(i)$ al tempo t esclusa la strategia del giocatore i . Anche in questo caso la strategia che massimizza il payoff potrebbe non essere unica, quindi si può effettuare la scelta in base ad una priorità, ad esempio:

$$x_i(t + 1) = \min\{d : d \in \operatorname{argmax}_{s_j \in S_0} c_i(s_j, s^{-i}(t))\}. \quad (4.24)$$

Sono ora disponibili tutti gli elementi necessari a definire un Networked Evolutionary Game.

Definizione 4.6. Un Networked Evolutionary Game, indicato con $((\mathcal{V}, \mathcal{E}), G, \Phi)$, è composta da:

1. un grafo, o network, $(\mathcal{V}, \mathcal{E})$;
2. un Fundamental Network Game G , tale che se $(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E}$ allora il giocatore i ed il giocatore j giocano ripetutamente a G tra di loro, usando rispettivamente le strategie $x_i(t)$ e $x_j(t)$. Nel caso G non sia simmetrico (come ad esempio il gioco dei maiali nel recinto), il grafo corrispondente deve essere orientato, in modo da evidenziare chi è il giocatore 1 e chi è il giocatore 2.
3. una legge di aggiornamento delle strategie della forma definita in (4.19).

Infine risulta necessario definire qual è il payoff complessivo di un giocatore all'interno di un NEG.

Definizione 4.7. Sia $c_{i,j}$ il payoff derivante dal FNG giocato tra i giocatori i ed j , allora si ha che:

1. se il grafo del NEG è omogeneo, il payoff complessivo del giocatore i è:

$$c_i(t) = \sum_{j \in U(i) \setminus i} c_{i,j}(t), \quad i = 1, 2, \dots, n; \quad (4.25)$$

2. se il grafo del NEG è eterogeneo, il payoff complessivo del giocatore i è:

$$c_i(t) = \frac{1}{|U(i)| - 1} \sum_{j \in U(i) \setminus i} c_{i,j}(t), \quad i = 1, 2, \dots, n. \quad (4.26)$$

4.3.1 Dinamica di un NEG

Per come è stato definito un NEG non appare immediato come esprimere la sua dinamica mediante una k -valued network, infatti la legge di aggiornamento delle strategie (4.19) è composta da funzioni che non dipendono solamente dalle strategie dei giocatori $x_j(t)$, $j \in U(i)$, ma anche dai loro payoff $c_j(t)$. Fortunatamente è possibile notare che i payoff $c_j(t)$ dipendono a loro volta dalle strategie dei giocatori vicini, ovvero da $x_h(t)$, $h \in U(j)$, e non da altri payoff. È possibile concludere che la strategia di un giocatore al tempo $t + 1$ dipende solamente dalle strategie al tempo t dei giocatori vicini raggiungibili in 2 passi. La considerazione appena fatta funge da dimostrazione alla seguente proposizione.

Proposizione 4.3.1. *Si consideri un NEG $((\mathcal{V}, \mathcal{E}), G, \Phi)$ composto da n giocatori ($|\mathcal{V}| = n$) i quali possono scegliere la propria strategia dall'insieme $S_0 = \{1, 2, \dots, k\}$; la dinamica del NEG, detta anche evoluzione del NEG, può essere espressa tramite il seguente sistema di equazioni logiche a k valori:*

$$\begin{cases} x_1(t+1) = f_1(\{x_j(t) | j \in U_2(1)\}) \\ x_2(t+1) = f_2(\{x_j(t) | j \in U_2(2)\}) \\ \vdots \\ x_n(t+1) = f_n(\{x_j(t) | j \in U_2(n)\}), \end{cases} \quad (4.27)$$

dove $x_i(t) \in \mathcal{L}_k$ rappresenta la strategia utilizzata dall' i -esimo giocatore al tempo $t \geq 0$. Nel caso $(\mathcal{V}, \mathcal{E})$ sia un grafo omogeneo, allora le funzioni logiche f_i , $i = 1, 2, \dots, n$, sono tutte uguali.

Dal sistema di equazioni (4.27) si deduce che un NEG è modellabile mediante una k -valued network, dunque è possibile esprimere la sua dinamica in forma algebrica; in particolare si ha che la dinamica evolutiva è esprimibile tramite il consueto sistema di equazioni algebriche:

$$\begin{cases} x_1(t+1) = M_{f_1} x(t) \\ x_2(t+1) = M_{f_2} x(t) \\ \vdots \\ x_n(t+1) = M_{f_n} x(t), \end{cases} \quad (4.28)$$

in cui $x(t) = \times_{i=1}^n x_i(t) \in \mathcal{L}_{k^n}$ e $M_{f_i} \in \mathcal{L}_{k \times k^n}$, $i = 1, \dots, n$. Infine si ottiene che la dinamica evolutiva è esprimibile mediante

$$x(t+1) = L x(t), \quad (4.29)$$

dove $L = M_{f_1} * M_{f_2} * \dots * M_{f_n} \in \mathcal{L}_{k^n \times k^n}$.

Nell'ambito dei NEG, il vettore $x(t)$ rappresenta il profilo di strategie al tempo t ; ne consegue che ogni profilo può essere espresso mediante il vettore logico $\delta_{k^n}^i$, $i = 1, 2, \dots, k^n$. Allo stesso modo anche le strategie possono essere espresse mediante il vettore logico δ_k^i , $i = 1, 2, \dots, k$.

Le matrici di struttura M_{f_i} , $i = 1, 2, \dots, n$ possono essere calcolate utilizzando gli strumenti approfonditi nella Sezione 2.1 o direttamente da delle tabelle in cui ad ogni profilo viene associato il corrispondente valore di f_i , come illustrato nel seguente esempio.

Esempio 4.3.1. Si consideri un Networked Evolutionary Game $((\mathcal{V}, \mathcal{E}), G, \Phi)$, in cui il grafo $(\mathcal{V}, \mathcal{E}) = R_4$ è rappresentato il Figura 4.3, G è il gioco dei maiali nel recinto e la legge di aggiornamento delle strategie Φ è la Risposta ottima. Data

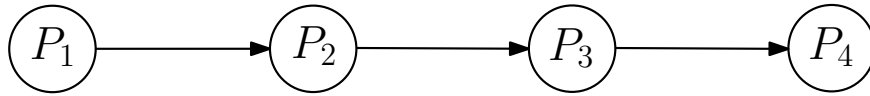


Figura 4.3: Grafo del NEG studiato nell'Esempio 4.3.1 .

l'asimmetria del FNG è necessario specificare il ruolo dei giocatori P_1, P_2, P_3, P_4 : identificando (con abuso di notazione) ogni maiale con il suo peso, risulta che $P_1 < P_2 < P_3 < P_4$. Tale ordine è anche deducibile dal grafo $(\mathcal{V}, \mathcal{E})$, in cui se un ramo è direzionato da P_i a P_j significa che nel FNG P_i e P_j sono rispettivamente

il giocatore 1 e 2. Per calcolare i valori dei payoff complessivi di ciascun giocatore è necessario prestare attenzione al fatto che il grafo è eterogeneo, dunque bisogna applicare l'equazione (4.26). Nel caso in cui $x_1(t) = \delta_2^1$, $x_2(t) = \delta_2^1$, $x_3(t) = \delta_2^1$ e $x_4(t) = \delta_2^1$, i valori dei payoff sono

$$\begin{aligned}
c_1(t) &= \frac{\sum_{j \in U(1) \setminus 1} c_{i,j}(t)}{|U(1)| - 1} = c_{1,2}(t) = 2 \\
c_2(t) &= \frac{\sum_{j \in U(2) \setminus 2} c_{i,j}(t)}{|U(2)| - 1} = \frac{c_{2,1}(t) + c_{2,3}(t)}{2} = \frac{5 + 2}{2} = 3.5, \\
c_3(t) &= \frac{\sum_{j \in U(3) \setminus 3} c_{i,j}(t)}{|U(3)| - 1} = \frac{c_{3,2}(t) + c_{3,4}(t)}{2} = \frac{2 + 5}{2} = 3.5, \\
c_4(t) &= \frac{\sum_{j \in U(4) \setminus 4} c_{i,j}(t)}{|U(4)| - 1} = c_{4,3}(t) = 5,
\end{aligned} \tag{4.30}$$

mentre i valore delle funzioni f_i , $i = 1, 2, 3, 4$, calcolate applicando Φ quando il profilo $x(t) = \delta_{16}^1$, sono

$$\begin{aligned}
f_1(x_1(t), x_2(t), x_3(t), x_4(t)) &= \delta_2^2, \\
f_2(x_1(t), x_2(t), x_3(t), x_4(t)) &= \delta_2^2, \\
f_3(x_1(t), x_2(t), x_3(t), x_4(t)) &= \delta_2^2, \\
f_4(x_1(t), x_2(t), x_3(t), x_4(t)) &= \delta_2^2.
\end{aligned} \tag{4.31}$$

Procedendo in modo analogo per tutti gli altri 15 profili si ottengono i risultati riportati in Tabella 4.10 in cui oltre alla matrice di payoff del NEG sono state aggiunte le 4 righe rappresentanti i valori di f_i , $i = 1, 2, 3, 4$, per ogni possibile profilo. Le matrici di struttura associate alle 4 funzioni logiche sono proprio le righe corrispondenti a f_1 , f_2 , f_3 e f_4 , in particolare si ottiene che

$$\begin{aligned}
M_{f_1} &= \delta_2[2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1], \\
M_{f_2} &= \delta_2[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1], \\
M_{f_3} &= \delta_2[2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 2, 1], \\
M_{f_4} &= \delta_2[2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1].
\end{aligned} \tag{4.32}$$

Infine la matrice di evoluzione è

$$\begin{aligned}
L &= M_{f_1} * M_{f_2} * M_{f_3} * M_{f_4} \\
&= \delta_{16}[16, 16, 15, 15, 8, 6, 7, 5, 16, 16, 11, 11, 8, 6, 3, 1]
\end{aligned} \tag{4.33}$$

e la dinamica evolutiva del NEG può essere scritta nella forma $x(t+1) = Lx(t)$. \triangle

Profilo	δ_{16}^1	δ_{16}^2	δ_{16}^3	δ_{16}^4	δ_{16}^5	δ_{16}^6	δ_{16}^7	δ_{16}^8
c_1	2	2	2	2	0	0	0	0
c_2	3.5	3.5	2.5	2.5	6.5	6.5	3.5	3.5
c_3	2.5	6.5	3.5	1.5	8	0.5	3	0
c_4	5	7	1	0	5	7	1	0
f_1	δ_2^2	δ_2^1	δ_2^2	δ_2^2	δ_2^1	δ_2^1	δ_2^1	δ_2^1
f_2	δ_2^2	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2
f_3	δ_2^2	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^1	δ_2^2	δ_2^1
f_4	δ_2^2	δ_2^1	δ_2^1	δ_2^1	δ_2^2	δ_2^2	δ_2^1	δ_2^1
Profilo	δ_{16}^9	δ_{16}^{10}	δ_{16}^{11}	δ_{16}^{12}	δ_{16}^{13}	δ_{16}^{14}	δ_{16}^{15}	δ_{16}^{16}
c_1	6	6	6	6	0	0	0	0
c_2	1.5	1.5	0.5	0.5	3	3	0	0
c_3	3.5	2.5	6.5	3.5	1.5	0.5	3	0
c_4	5	7	1	0	5	7	1	0
f_1	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^1	δ_2^1	δ_2^1	δ_2^1
f_2	δ_2^2	δ_2^2	δ_2^1	δ_2^1	δ_2^2	δ_2^2	δ_2^1	δ_2^1
f_3	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^1	δ_2^2	δ_2^1
f_4	δ_2^2	δ_2^2	δ_2^1	δ_2^1	δ_2^2	δ_2^2	δ_2^1	δ_2^1

Tabella 4.10: Matrice di payoff e valori delle funzioni f_i , $i = 1, 2, 3, 4$ per ogni profilo del NEG esposto nell'Esempio 4.3.1.

Una volta ottenuta la forma algebrica che descrive la dinamica evolutiva di un NEG è possibile applicare tutti gli strumenti approfonditi nel Capitolo 3 per studiare il comportamento, specialmente asintotico, del NEG. In questo contesto le proprietà di una k -valued network assumono un significato concreto, come illustrato nel seguente esempio.

Esempio 4.3.2. Con riferimento all'Esempio 4.3.1, mediante lo studio della matrice evolutiva L è possibile studiare l'andamento del NEG in base al profilo delle strategie di partenza, ovvero $x(0)$.

Partendo dalle proprietà più generali si ha che

$$L = L^3, \quad (4.34)$$

dunque il tempo di transizione è $T_t = 1$ e $\tau = 3 - 1 = 2$. Questo significa che, a prescindere dalle condizioni iniziali, dopo un solo turno di gioco, le strategie scelte

dai giocatori si ripetono ciclicamente o non variano più. Risulta poi che $[L]_{6,6}$, $[L]_{7,7}$ e $[L]_{11,11}$ sono pari ad 1, quindi i profili dei giocatori che se adottati non variano più sono δ_{16}^6 , δ_{16}^7 e δ_{16}^{11} . Essendoci tre punti di equilibrio si ha che $N_1 = 3$, mentre i cicli di lunghezza 2 sono pari a

$$N_2 = \frac{\text{tr}(L^2) - \sum_{i \in \mathcal{P}(2)} i N_i}{2} = \frac{9 - (1 \cdot 3)}{2} = 3 \quad (4.35)$$

e la non esistenza di cicli di lunghezza superiore è garantita dal valore di τ . In particolare i cicli risultano essere: $\mathcal{C}_1 = (\delta_{16}^1, \delta_{16}^{16})$, $\mathcal{C}_2 = (\delta_{16}^3, \delta_{16}^{15})$ e $\mathcal{C}_3 = (\delta_{16}^5, \delta_{16}^8)$. Si noti che il profilo che garantisce la somma di payoff maggiore è δ_{16}^{10} , che non è nè un punto di equilibrio, nè un profilo appartenente ad un ciclo, dunque quando il gioco comincia con questo profilo come condizione iniziale, dopo solo un passo non vi torna più; questa considerazione, tuttavia, è del tutto ovvia trattandosi di un gioco non cooperativo, in cui la somma dei payoff non ha alcuna importanza. Il NEG non è stabile e non presenta cicli globalmente attrattivi essendoci più cicli e più punti di equilibrio. Il ciclo \mathcal{C}_1 risulta particolarmente interessante perché mostra come i giocatori passino ciclicamente da un profilo che garantisce a tutti loro un guadagno discreto, ovvero il profilo δ_{16}^1 , all'unico profilo che nega un guadagno a tutti, ovvero δ_{16}^{16} . \triangle

Si conclude questo paragrafo mostrando come una diversa legge di aggiornamento delle strategie Φ , modifichi drasticamente l'andamento di un NEG.

Esempio 4.3.3. Con riferimento all'Esempio 4.3.1, si consideri lo stesso NEG nel quale la legge di aggiornamento delle strategie Φ non è più la Risposta ottima, ma l'Imitazione incondizionata. La matrice di payoff ovviamente rimane la stessa, mentre cambiano i valori delle funzioni logiche f_i , $i = 1, 2, \dots, n$, i cui valori sono riportati in Tabella 4.11. Utilizzando questo tipo di Φ , ogni volta che i giocatori assumono tutti la stessa strategia, essi non la cambiano più non avendo giocatori vicini che ne utilizzano una diversa; questo garantisce dei punti di equilibrio in corrispondenza a tutti quei profili composti dalla stessa strategia. Per analizzare meglio queste proprietà si calcola la matrice di evoluzione del NEG dalle matrici di struttura M_{f_i} , $i = 1, 2, 3, 4$, ottenendo:

$$\begin{aligned} L &= M_{f_1} * M_{f_2} * M_{f_3} * M_{f_4} \\ &= \delta_{16}[1, 4, 8, 8, 15, 16, 16, 16, 13, 16, 16, 16, 13, 16, 16, 16] \end{aligned} \quad (4.36)$$

A questo punto è immediato notare che $[L]_{1,1}$, $[L]_{13,13}$ e $[L]_{16,16}$ hanno valore unitario e dunque sono dei punti di equilibrio per la rete. Si ha poi che

$$L^t = \delta_{16}[1, 16, 16, 16, 16, 16, 16, 16, 13, 16, 16, 16, 13, 16, 16, 16], \quad \forall t \geq 3, \quad (4.37)$$

Profilo	δ_{16}^1	δ_{16}^2	δ_{16}^3	δ_{16}^4	δ_{16}^5	δ_{16}^6	δ_{16}^7	δ_{16}^8
c_1	2	2	2	2	0	0	0	0
c_2	3.5	3.5	2.5	2.5	6.5	6.5	3.5	3.5
c_3	2.5	6.5	3.5	1.5	8	0.5	3	0
c_4	5	7	1	0	5	7	1	0
f_1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^2
f_2	δ_2^1	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2
f_3	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2
f_4	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^1	δ_2^2	δ_2^2	δ_2^2
Profilo	δ_{16}^9	δ_{16}^{10}	δ_{16}^{11}	δ_{16}^{12}	δ_{16}^{13}	δ_{16}^{14}	δ_{16}^{15}	δ_{16}^{16}
c_1	6	6	6	6	0	0	0	0
c_2	1.5	1.5	0.5	0.5	3	3	0	0
c_3	3.5	2.5	6.5	3.5	1.5	0.5	3	0
c_4	5	7	1	0	5	7	1	0
f_1	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2
f_2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2	δ_2^2
f_3	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^1	δ_2^2	δ_2^2	δ_2^2
f_4	δ_2^1	δ_2^2	δ_2^2	δ_2^2	δ_2^1	δ_2^2	δ_2^2	δ_2^2

Tabella 4.11: Matrice di payoff e valori delle funzioni f_i , $i = 1, 2, 3, 4$ per ogni profilo del NEG studiato nell'Esempio 4.3.3.

dunque il tempo di transizione è $T_t = 3$. Dopo tre turni la dinamica del gioco non si modifica più e i profili adottati dai giocatori possono essere soltanto tre ($\delta_{16}^1, \delta_{16}^{13}, \delta_{16}^{16}$), in base a quale è il profilo iniziale. Il profilo $\delta_{16}^1 = (1, 1, 1, 1)$ viene adottato dai giocatori per ogni $t > 0$ se e solo se quello era anche il profilo iniziale, ovvero se $x(0) = \delta_{16}^1$. Il profilo $\delta_{16}^{13} = (2, 2, 1, 1)$ viene adottato sia nel caso esso sia il profilo iniziale, sia nel caso il profilo iniziale sia $x(0) = \delta_{16}^9$. Partendo da tutte le altre combinazioni di strategie invece si ha che la dinamica del gioco si assesta sul profilo $\delta_{16}^{16} = (2, 2, 2, 2)$. Tranne in tre casi dunque il gioco si risolve con tutti e 4 i maiali che scelgono la strategia 2, ovvero attendono, ottenendo in questo modo il payoff più basso.

È evidente che una legge di aggiornamento delle strategie differente ha portato ad un comportamento molto diverso del NEG: utilizzando la Risposta ottima si sono creati dei cicli in cui i giocatori alternano differenti profili, mentre applicando l'Imi-

tazione incondizionata dopo pochi turni il NEG si assesta su dei ben precisi profili. Per questo particolare NEG sembra ottenere risultati più soddisfacenti la Risposta ottima, infatti qualsiasi condizione iniziale garantisce, una volta arrivati a “regime”, un payoff costantemente non nullo a tutti i giocatori escluso P_1 , che ottiene un guadagno costantemente nullo nel caso in cui $x(0) \in \{\delta_{16}^5, \delta_{16}^6, \delta_{16}^7, \delta_{16}^8, \delta_{16}^{13}, \delta_{16}^{14}\}$, ovvero nel 37.5% dei casi (se le condizioni iniziali sono equiprobabili). Tuttavia per scegliere quale legge di aggiornamento sia la migliore è necessario venga fornito un obiettivo. \triangle

4.3.2 NEG omogenei

Nel paragrafo precedente è stato esposto un procedimento per modellizzare un NEG tramite una k -valued network, in modo da poterne studiare il comportamento e le proprietà. Il procedimento è applicabile a NEG con un numero di giocatori $|\mathcal{V}| = n$ arbitrario, tuttavia nel caso in cui n sia elevato la complessità computazionale necessaria a creare il modello e studiarlo diventa troppo elevata; basti pensare che per un NEG di dimensioni contenute composto da 8 giocatori le cui strategie a disposizione sono 3, la matrice evolutiva L ha dimensione 6561×6561 .

Nei NEG omogenei però tutte le f_i , $i = 1, 2, \dots, n$, sono uguali e le dinamiche di tutti i nodi (i giocatori) sono le stesse, dunque modellare localmente la dinamica della rete porta ad avere informazioni sul comportamento globale dell'intero NEG. Nel seguito verranno approfondite le proprietà dei NEG omogenei il cui grafo è O_n , anche se considerazioni simili possono essere fatte per tutti i tipi di NEG omogenei.

Prima di proseguire risulta utile definire il concetto di *sub-game*.

Definizione 4.8. Dato un $\text{NEG} = ((\mathcal{V}, \mathcal{E}), G, \Phi)$, se $(\mathcal{V}', \mathcal{E}')$ è un sottografo di $(\mathcal{V}, \mathcal{E})$, allora il $\text{NEG} = ((\mathcal{V}', \mathcal{E}'), G, \Phi)$ è detto sub-game del gioco di partenza.

Si consideri dunque un NEG in cui $|S_0| = k$ ed il cui grafo è O_n , dove $n \geq 5$ (le considerazioni che verranno fatte non risultano particolarmente utili per $n < 5$); dal sistema di equazioni (4.27) si deduce che l'equazione di aggiornamento della strategia del i -esimo nodo può essere riscritta come

$$x_i(t+1) = f(x_{i-2}(t), x_{i-1}(t), x_i(t), x_{i+1}(t), x_{i+2}(t)), \quad i = 1, 2, \dots, n, \quad (4.38)$$

dove $x_{n+j} = x_j$, $x_{-j} = x_{n+j}$, $1 \leq j \leq n-1$ e $n \geq 5$. Indicando con $M_f \in \mathcal{L}_{k \times k^n}$ la matrice di struttura di f , la (4.38) può essere riscritta come

$$x_i(t+1) = M_f \times_{j=-2}^2 x_{i+j}(t), \quad i = 1, 2, \dots, n. \quad (4.39)$$

La matrice M_f sta alla base della dinamica dell'intero NEG, infatti tramite essa e l'utilizzo di matrici swap W e di matrici D_f (o D_r) è possibile calcolare l'evoluzione

dei sub-games (di dimensioni arbitrarie) del NEG e naturalmente anche la dinamica di tutti i nodi; tuttavia ricavare la dinamica complessiva, ovvero la matrice di evoluzione L , non risulta interessante in questo contesto dato che l'obbiettivo è proprio quello di non dover analizzare l'intero NEG, che potrebbe avere un numero di giocatori n troppo elevato.

Considerando un sub-game di dimensione l , $5 \leq l \leq n$, il cui vettore di stato è $\bar{x}(t) = \times_{j=1}^l x_j(t)$ e il cui grafo è O_l , la dinamica del nodo i può essere scritta come

$$x_i(t+1) = M_i \bar{x}(t), \quad (4.40)$$

dove

$$M_i = M_f D_f^{k^5, k^{l-5}} W_{[k^{\alpha(i)}, k^{l-\alpha(i)}]}, \quad l \geq 5, \quad (4.41)$$

e

$$\alpha(i) = \begin{cases} i - 3 + l & i < 3 \\ i - 3 & i \geq 3. \end{cases} \quad (4.42)$$

La definizione di M_i , $i = 1, 2, \dots, n$ discende direttamente dall'equazione (4.39): essa è calcolata a partire da M_f e mediante l'utilizzo di matrici Swap per ordinare in modo corretto i nodi x_i , $i = 1, 2, \dots, l$ e mediante l'uso della matrice D_f per far comparire tutte le $l - 5$ variabili che non compaiono nell'equazione (4.39). Si noti che nel caso in cui $l = 5$ non è necessario far comparire alcuna variabile, infatti risulta che $D_f^{k^5, k^0} = I_{k^5}$. Il ruolo di $\alpha(i)$, $i = 1, 2, \dots, l$ è quello di considerare la differenza che c'è nell'ordinare le variabili se il nodo in esame è il primo o il secondo o se è un altro nodo differente da questi. Si noti che quando si prende in considerazione il terzo nodo, ovvero $i = 3$, risulta $\alpha(3) = 0$ e la matrice Swap $W_{[1, k^l]}$ è la matrice identità di dimensione k^l , ovvero non avviene nessun riordinamento delle variabili.

La matrice di evoluzione del sub-game di dimensione l , indicata con $L_l \in \mathcal{L}_{k^l \times k^l}$ è

$$L_l = M_1 * M_2 * \dots * M_l \quad (4.43)$$

e la dinamica del sub game può essere scritta nella forma algebrica:

$$\bar{x}(t+1) = L_l \bar{x}(t). \quad (4.44)$$

Naturalmente L_n è la matrice di evoluzione dell'intero NEG.

Osservazione 13. Si presti attenzione al fatto che la dinamica degli $l < n$ giocatori all'interno del NEG di partenza non è descritta solamente da (4.44), infatti essa può essere influenzata dalle strategie usate dagli altri $n - l$ giocatori. Quando si parla della dinamica di un sub-game si fa riferimento ad un NEG composto da un

sottoinsieme di giocatori del NEG di partenza, non influenzato da giocatori che non vi appartengono. Tuttavia vedremo che quando sono rispettate certe condizioni è possibile studiare il comportamento dei giocatori di un NEG sfruttando la matrice di evoluzione di un suo sub-game.

Esempio 4.3.4. Si consideri un Networked Evolutionary Game $((\mathcal{V}, \mathcal{E}), G, \Phi)$, in cui il grafo è $(\mathcal{V}, \mathcal{E}) = O_n$, G è il dilemma del prigioniero e la legge di aggiornamento delle strategie Φ è l'imitazione incondizionata. Per calcolare la matrice M_f si fa riferimento alla Tabella 4.3.4 in cui viene calcolata la funzione di aggiornamento della strategia del giocatore i che si basa sul payoff dei giocatori vicini; si presti attenzione che i profili considerati sono quelli dei vicini raggiungibili in 2 passi a partire da i , ovvero quelli appartenenti a $U_2(i)$, come descritto da (4.38). La matrice M_f risultante è:

$$M_f = \delta_2 [1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, \\ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2]. \quad (4.45)$$

Partendo da M_f siamo interessati a calcolare la matrice di evoluzione di un sub-game composto da 5 giocatori, ovvero un NEG identico a quello appena analizzato il cui grafo è O_5 . Utilizzando le equazioni (4.41) e (4.42) si calcolano innanzitutto le matrici M_i , $i = 1, 2, \dots, 5$:

$$\begin{aligned} M_1 &= M_f D_f^{2^5, 2^{5-5}} W_{[2^3, 2^{5-3}]} \\ &= \delta_2 [1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, \\ &\quad 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2], \\ M_2 &= M_f D_f^{2^5, 2^{5-5}} W_{[2^4, 2^{5-4}]} \\ &= \delta_2 [1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, \\ &\quad 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2], \\ M_3 &= M_f D_f^{2^5, 2^{5-5}} W_{[2^0, 2^{5-0}]} \\ &= \delta_2 [1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, \\ &\quad 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2], \\ M_4 &= M_f D_f^{2^5, 2^{5-5}} W_{[2^1, 2^{5-1}]} \\ &= \delta_2 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, \\ &\quad 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2, 2], \\ M_5 &= M_f D_f^{2^5, 2^{5-5}} W_{[2^2, 2^{5-2}]} \\ &= \delta_2 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \\ &\quad 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2]. \end{aligned} \quad (4.46)$$

Infine la matrice di evoluzione del sub-game è

$$\begin{aligned} L_5 &= M_1 * M_2 * M_3 * M_4 * M_5 \\ &= \delta_{32}[1, 1, 1, 1, 1, 1, 1, 29, 1, 1, 1, 1, 1, 1, 15, 13, \\ &\quad 1, 1, 1, 26, 1, 1, 1, 25, 1, 20, 1, 18, 8, 4, 7, 32]. \end{aligned} \quad (4.47)$$

Proseguendo con l'analisi dell'evoluzione del sub-game, si può notare che

$$\begin{aligned} L_5^2 &= L_5^4 = \delta_{32}[1, 1, 1, 1, 1, 1, 1, 8, 1, 1, 1, 1, 1, 1, 15, 1, \\ &\quad 1, 1, 1, 20, 1, 1, 1, 1, 1, 26, 1, 18, 29, 1, 1, 32] \end{aligned} \quad (4.48)$$

ed ispezionando gli elementi della matrice L_5^2 è possibile constatare che il profilo $\delta_{32}^1 = (1, 1, 1, 1, 1)$ è il più presente (78.1%), ovvero i giocatori nella maggior parte dei casi, considerando condizioni iniziali equiprobabili, dopo solo 2 turni sono portati tutti a confessare. \triangle

Si è ora interessati ad indagare le proprietà di un NEG analizzando la dinamica dei sub-games descritti dalle matrici L_l con $l < n$. Prima di riportare un interessante risultato in tal senso, è utile introdurre il seguente lemma [14].

Lemma 4.3.2. *Dati i vettori $x_i \in \mathcal{L}_k$, $i = 1, 2, \dots, n$ ed il vettore $x = \times_{i=1}^n x_i$, si ha che*

$$x_i = \pi_i x, \quad i = 1, 2, \dots, n, \quad (4.49)$$

dove le matrici π_i $i = 1, 2, \dots, n$, sono matrici di proiezione di dimensione $k \times k^n$ che hanno la seguente struttura:

$$\begin{aligned} \pi_1 &= \delta_k[\mathbb{1}_{k^{n-1}}^T, 2 \cdot \mathbb{1}_{k^{n-1}}^T, \dots, k \cdot \mathbb{1}_{k^{n-1}}^T], \\ \pi_2 &= \delta_k[\mathbb{1}_{k^{n-2}}^T, 2 \cdot \mathbb{1}_{k^{n-2}}^T, \dots, k \cdot \mathbb{1}_{k^{n-2}}^T, \mathbb{1}_{k^{n-2}}^T, 2 \cdot \mathbb{1}_{k^{n-2}}^T, \dots, k \cdot \mathbb{1}_{k^{n-2}}^T], \\ &\vdots \\ \pi_n &= \delta_k[\underbrace{1, 2, \dots, k}_k, \underbrace{1, 2, \dots, k}_k, \dots, \underbrace{1, 2, \dots, k}_k]. \end{aligned} \quad (4.50)$$

Si consideri ora un nodo x_h , $h \in \{1, 2, \dots, n\}$, di un NEG omogeneo il cui grafo è O_n : dall'equazione (4.27) si ha che $x_h(1) = f(\{x_j(0) | j \in U_2(h)\})$ e proseguendo si ottiene che

$$\begin{aligned} x_h(2) &= f(\{x_j(0) | j \in U_{2,2}(h)\}), \\ x_h(3) &= f(\{x_j(0) | j \in U_{2,3}(h)\}), \\ &\vdots \\ x_h(t) &= f(\{x_j(0) | j \in U_{2t}(h)\}), \quad t \in \mathbb{N}_+, \end{aligned} \quad (4.51)$$

ovvero, dopo t turni, i nodi le cui condizioni iniziali influenzano x_h sono i suoi vicini raggiungibili in $2t$ passi. Dunque la scelta della strategia del giocatore h al tempo

t dipende dalla strategia iniziale di ciascun giocatore in $U_{2t}(h)$ ed il numero di tali giocatori quando il grafo è O_n è pari a $|U_{2t}(h)| = 2t + 2t + 1 = 4t + 1$. Sia L_{4t+1} la matrice di evoluzione che descrive la dinamica di un sub-game di dimensione $4t + 1$ del NEG di partenza e siano $(x_{h-2t}(0), x_{h-2t+1}(0), \dots, x_h(0), \dots, x_{h+2t-1}(0), x_{h+2t}(0))$ le condizioni iniziali (nel NEG originario) da cui dipende $x_h(t)$. La numerazione dei nodi $(x_{h-2t}, x_{h-2t+1}, \dots, x_h, \dots, x_{h+2t-1}, x_{h+2t})$ dipende dallo loro posizione nel NEG di partenza; si consideri ora la seguente diversa numerazione

$$\begin{aligned} x'_1 &= x_{h-2t}, \\ x'_2 &= x_{h-2t+1}, \\ &\vdots \\ x'_\beta &= x_h, \\ &\vdots \\ x'_{4t} &= x_{h+2t-1}, \\ x'_{4t+1} &= x_{h+2t}, \end{aligned} \tag{4.52}$$

che corrisponde alla numerazione dei nodi in un sub-game di dimensione $4t + 1$ descritto da L_{4t+1} . Lo scopo di tale numerazione è quella di definire il valore di β ; in particolare quando il grafo è O_n si ha che $\beta = \lceil (4t + 1)/2 \rceil$, dunque β non dipende da h , ovvero non dipende dalla posizione del nodo preso in esame nel grafo del NEG a n giocatori. Infine si definisce il vettore di stato delle condizioni iniziali che influenzano $x_h(t)$:

$$\begin{aligned} v_{h,t}(0) &= \times_{j=-2t}^{2t} x_{h+j}(0) \\ &= \times_{j=1}^{4t+1} x'_j(0). \end{aligned} \tag{4.53}$$

È ora possibile calcolare il valore di $x_h(t)$ partendo dalle condizioni iniziali dei nodi appartenenti a $U_{2t}(h)$ mediante la seguente equazione algebrica

$$x_h(t) = \pi_\beta L_{4t+1}^t v_{h,t}(0), \tag{4.54}$$

dove $\pi_\beta \in \mathcal{L}_{k \times k^{4t+1}}$, $L_{4t+1} \in \mathcal{L}_{k^{4t+1} \times k^{4t+1}}$ e $v_{h,t}(0) \in \mathcal{L}_{k^{4t+1}}$. Si è dunque ottenuta un'equazione algebrica che permette di calcolare il valore di un nodo al tempo t mediante la matrice di evoluzione di un sub-game di dimensione $4t + 1$ e le condizioni iniziali dei nodi appartenenti ad un intorno di $2t$ passi del nodo preso in esame. È importante notare che il valore di $x_h(t)$ non dipende dalle strategie usate dagli $n - (4t + 1)$ giocatori non appartenenti a $U_{2t}(h)$, tuttavia per il calcolo di $x_h(t + T)$, $T > 0$, non è più sufficiente utilizzare L_{4t+1} e considerare $U_{2t}(h)$, ammenoché non si verifichi una particolare condizione, come esposto nella seguente proposizione, dimostrata da Cheng et al. in [14].

Proposizione 4.3.3. *Dato un NEG composto da n giocatori i quali hanno a disposizione k strategie, si consideri la strategia $x_h(\cdot)$ utilizzata dal giocatore $h \in [1, n]$; se esiste un $r > 0$ tale che*

$$L_{|U_{2(r+1)(\cdot)}|}^r = L_{|U_{2(r+1)(\cdot)}|}^{r+1}, \quad (4.55)$$

dove $L_{|U_{2(r+1)(\cdot)}|}$ è la matrice di evoluzione di un sub-game di dimensione $|U_{2(r+1)(\cdot)}|$ del NEG, allora si ha che

$$x_h(t) = \pi_\beta L_{|U_{2r(\cdot)}|}^r v_{h,r}(0), \quad \forall t \geq r, \quad \forall h \in [1, n], \quad (4.56)$$

in cui $v_{h,r}(0)$ è definito da (4.53) e β è l'indice del nodo x_h nell'ordinamento dei nodi definito da (4.52). Nel caso in cui il grafo del NEG è O_n si ha che $|U_{2(r+1)(\cdot)}| = 4r + 5$, $|U_{2r(\cdot)}| = 4r + 1$ e $\beta = \lceil (4r + 1)/2 \rceil$.

Ciò che afferma la proposizione ha un'importanza notevole: se per qualche $r > 0$ si verifica l'uguaglianza (4.55) allora il valore di un qualsiasi nodo del NEG può essere calcolato mediante la matrice di evoluzione di un sub-game di dimensione $|U_{2r(\cdot)}|$ e mediante le condizioni iniziali dei nodi in un intorno di $2r$ passi del nodo preso in esame. Si presti attenzione che questo vale per ogni valore di $t > 0$ e dunque permette un'analisi asintotica delle strategie usate dai giocatori a cui siamo interessati, senza dover ricorrere alla matrice di evoluzione L_n del NEG e senza dovere considerare le strategie usate da tutti gli n giocatori.

Esempio 4.3.5. Si consideri un NEG = $((\mathcal{V}, \mathcal{E}), G, \Phi)$, dove $(\mathcal{V}, \mathcal{E}) = O_n$ e il Fundamental Network Game G ha come insieme delle strategie $S_0 = \{1, 2\}$. Calcolando le varie matrici di evoluzioni dei sub-games del NEG si ottiene che

$$L_{17}^3 = L_{17}^4, \quad (4.57)$$

quindi l'uguaglianza (4.55) è verificata per $r = 3$. Per calcolare la strategia usata da un qualsiasi giocatore per $t \geq 3$ è dunque sufficiente considerare il suo intorno $U_{2r} = U_6$, composto da 13 nodi. Dato che il grafo considerato è O_n risulta che $\beta = 7$. Volendo calcolare ad esempio la strategia che il giocatore 18 utilizza nel NEG per $t \geq 3$, è necessario conoscere le seguenti condizioni iniziali

$$x_{12}(0), x_{13}(0), \dots, x_{23}(0), x_{24}(0), \quad (4.58)$$

e calcolare il vettore $v_{18,3}(0) = \times_{i=12}^{24} x_i(0)$. Infine si ottiene che

$$\begin{aligned} x_{18}(t) &= \pi_\beta L_{4r+1}^r v_{18,r}(0), \\ &= \pi_7 L_{13} v_{18,3}(0), \quad \forall t \geq 3. \end{aligned} \quad (4.59)$$

Lo stesso procedimento può essere applicato ad ogni nodo del NEG, con la ovvia modifica al vettore di stato delle condizioni iniziali. \triangle

4.4 Controllo dei NEGs

Si consideri ora il caso in cui risulti possibile scegliere liberamente la strategia adottata da uno o più giocatori in un NEG; in tal caso è dunque possibile effettuare un controllo dall'esterno sull'evoluzione del gioco.

Definizione 4.9. Dato un $\text{NEG} = ((\mathcal{V}, \mathcal{E}), G, \Phi)$, si consideri una partizione $\{\mathcal{X}, \mathcal{U}\}$ di \mathcal{V} , ovvero $\mathcal{X} \cup \mathcal{U} = \mathcal{V}$ e $\mathcal{X} \cap \mathcal{U} = \emptyset$. Se le strategie dei nodi $u_j \in \mathcal{U}$, $j = 1, 2, \dots, |\mathcal{U}|$ possono essere assegnate liberamente ad ogni istante $t \geq 0$ allora $((\mathcal{X} \cup \mathcal{U}, \mathcal{E}), G, \Phi)$ è detto NEG controllato. I nodi appartenenti a \mathcal{U} sono chiamati controlli.

Nell'ambito dei NEG controllati risulta interessante sia studiare le possibilità ottenibili dal controllo di determinati giocatori già presenti nel gioco, sia analizzare gli effetti dell'introduzione di uno o più nuovi giocatori controllati, modificando dunque il grafo del NEG. Un NEG controllato risulta modellabile mediante una controlled k -valued network ed è quindi possibile applicare tutti i risultati ottenuti nella Sezione 3.4. Si termina questo capitolo con un esempio che espone gli effetti ottenibili da un controllo su di un NEG.

Esempio 4.4.1. Si consideri un $\text{NEG} = ((\mathcal{V}, \mathcal{E}), G, \Phi)$ in cui G è il gioco di Benoit-Krishna, $(\mathcal{V}, \mathcal{E}) = R_3$ e Φ è l'Imitazione incondizionata a priorità fissa. La disposizione dei giocatori è illustrata in Figura 4.4. La matrice di payoff e i valori di

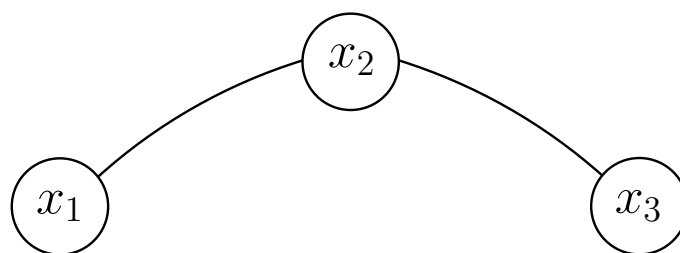


Figura 4.4: Grafo del NEG studiato nell'Esempio 4.4.1.

f_i , $i = 1, 2, 3$, sono riportati in Tabella 4.13. Le matrici di struttura delle funzione f_i , $i = 1, 2, 3$, ottenibili direttamente dalle ultime 3 righe della Tabelle 4.13, sono

rispettivamente

$$\begin{aligned}
M_{f_1} &= \delta_3[1, 1, 1, 1, 1, 1, 3, 3, 3, 1, 1, 1, 2, 2, \\
&\quad 2, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3], \\
M_{f_2} &= \delta_3[1, 1, 3, 1, 2, 1, 3, 3, 3, 1, 1, 3, 2, 2, \\
&\quad 3, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3], \\
M_{f_3} &= \delta_3[1, 1, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 1, 2, \\
&\quad 2, 3, 3, 3, 1, 1, 3, 1, 2, 2, 3, 3, 3]
\end{aligned} \tag{4.60}$$

e la matrice di evoluzione del NEG è

$$\begin{aligned}
L &= M_{f_1} * M_{f_2} * M_{f_3} \\
&= \delta_{27}[1, 1, 9, 1, 5, 3, 27, 27, 27, 1, 1, 9, 13, 14, \\
&\quad 14, 27, 15, 27, 25, 25, 27, 25, 14, 26, 27, 27, 27].
\end{aligned} \tag{4.61}$$

Analizzando L si può notare la presenza di 5 punti fissi, corrispondenti ai profili δ_{27}^1 , δ_{27}^5 , δ_{27}^{13} , δ_{27}^{14} e δ_{27}^{27} . Proseguendo l'analisi si ottiene che

$$\begin{aligned}
L^t &= \delta_{27}[1, 1, 27, 1, 5, 27, 27, 27, 27, 1, 1, 27, 13, 14, \\
&\quad 14, 27, 14, 27, 27, 27, 27, 27, 14, 27, 27, 27, 27] \quad \forall t \geq 3,
\end{aligned} \tag{4.62}$$

dunque il tempo di transizione è $T_t = 3$ e non sono presenti cicli. Si noti che, supponendo condizioni iniziali equiprobabili, nel 59.3% dei casi il profilo finale del NEG è $\delta_{27}^{27} = (3, 3, 3)$, corrispondente al caso in cui tutti i prigionieri confessano.

Si supponga di voler incrementare i casi in cui i tre prigionieri sono portati a confessare mediante l'utilizzo di un controllo. A tale scopo si introduce un giocatore P_u (che può essere considerato un infiltrato), la cui strategia al tempo t viene denotata con $u(t) \in \{1, 2, 3\}$. Il giocatore P_u può comunicare solo con P_1 e P_3 e il NEG controllato corrispondente possiede il grafo rappresentato in Figura 4.5. In Tabella 4.14 è riportata la matrice di payoff del NEG controllato ed i valori di f_i , $i = 1, 2, 3$, quando il controllo è $u = \delta_3^1$; il payoff del giocatore P_u è indicato con c_u . Allo stesso modo si ricavano le tabelle relative ai casi in cui $u = \delta_3^2$ e $u = \delta_3^3$ e da esse, mediante i calcoli visti più volte, si ottengono le matrici:

$$\begin{aligned}
L_1 &= \delta_{27}[1, 1, 9, 1, 1, 9, 27, 27, 27, 1, 1, 9, 1, 14, \\
&\quad 18, 27, 27, 27, 19, 25, 27, 25, 26, 27, 27, 27, 27],
\end{aligned} \tag{4.63}$$

$$\begin{aligned}
L_2 &= \delta_{27}[1, 1, 9, 1, 5, 3, 27, 27, 27, 1, 11, 18, 13, 14, \\
&\quad 14, 27, 14, 14, 25, 26, 27, 25, 14, 26, 27, 14, 27],
\end{aligned} \tag{4.64}$$

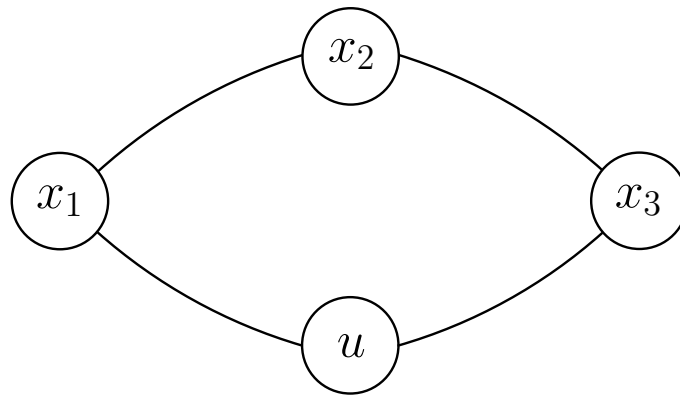


Figura 4.5: Grafo del NEG controllato studiato nell'Esempio 4.4.1.

$$L_3 = \delta_{27} [21, 21, 27, 21, 24, 27, 27, 27, 27, 21, 1, 27, 24, 14, \\ 14, 27, 15, 27, 27, 27, 27, 14, 27, 27, 27, 27], \quad (4.65)$$

dove L_i , $i = 1, 2, 3$, è la matrice di evoluzione del NEG derivante dal NEG controllato quando $u = \delta_3^i$. La matrice di evoluzione del NEG controllato è dunque $L = [L_1, L_2, L_3]$ e la dinamica del gioco è descritta da

$$x(t+1) = L u(t) x(t), \quad (4.66)$$

dove $x(t) = x_1(t)x_2(t)x_3(t)$. Per l'analisi delle proprietà più importanti del NEG controllato è utile il calcolo della matrice $L_{tot} = L_1 + L_2 + L_3$. Da essa è poi

sequenza di controllo che porta il NEG dal profilo δ_{27}^1 al profilo δ_{27}^{27} . L'unico ingresso che non fa rimanere il gioco nel profilo δ_{27}^1 è $u = \delta_3^3$ che porta il sistema nel profilo δ_{27}^{21} ; una volta giunti in questo profilo qualsiasi valore dell'ingresso porta il gioco al profilo desiderato δ_{27}^{27} . Riassumendo la sequenza di ingresso da utilizzare è

$$u(0) = \delta_3^3 \text{ e } u(1) = \delta_3^*, \quad (4.67)$$

dove \star può assumere indistintamente il valore 1, 2 e 3. In tal modo si ottiene:

$$x(2) = L_\star L_3 \delta_{27}^1 = \delta_{27}^{27}. \quad (4.68)$$

Il controllo da applicare affinché il profilo rimanga δ_{27}^{27} per $t > 2$ può valere indistintamente δ_3^1 , δ_3^2 e δ_3^3 , dunque la scelta più semplice è quella di applicare un ingresso costante per $t \geq 2$. \triangle

Un concetto particolarmente interessante nel contesto dei NEG controllati è quello di consensus.

Definizione 4.10. Sia ξ una strategia appartenente a $S_0 = \mathcal{D}_k$. Si dice che un NEG controllato composto da n giocatori che possono scegliere la propria strategia da S_0 raggiunge il *consensus* in ξ se esso è stabilizzabile al profilo $x_e = (\delta_k^\xi)^n$.

Nell'esempio precedente il NEG non raggiungeva un consensus per nessuna strategia; ciò era dovuto alla presenza di due punti fissi comuni alle matrici L_1 , L_2 e L_3 . Denotando con $L_{\mathcal{U}}$ l'insieme che comprende tutte le matrici $L_i = L \delta_{k|u|}^i$, $i = 1, 2, \dots, k^{|u|}$, è possibile esporre il seguente risultato che formalizza l'osservazione appena fatta.

Proposizione 4.4.1. *Se i punti fissi comuni a tutte le matrici appartenenti a $L_{\mathcal{U}}$ sono più di uno, allora il NEG controllato non può raggiungere un consensus. Se x_e è il solo punto fisso comune a tutte le matrici appartenenti a $L_{\mathcal{U}}$ e il NEG raggiunge il consensus in $\xi \in S_0$, allora $x_e = (\delta_k^\xi)^n$.*

Profilo	δ_{32}^1	δ_{32}^2	δ_{32}^3	δ_{32}^4	δ_{32}^5	δ_{32}^6	δ_{32}^7	δ_{32}^8
c_{i-1}	-10	-10	-10	-10	-5	-5	-5	-5
c_i	-10	-10	-5	-5	-14	-14	-8	-8
c_{i+1}	-10	-5	-14	-8	-5	0	-8	-2
f_i	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^2
Profilo	δ_{32}^9	δ_{32}^{10}	δ_{32}^{11}	δ_{32}^{12}	δ_{32}^{13}	δ_{32}^{14}	δ_{32}^{15}	δ_{32}^{16}
c_{i-1}	-14	-14	-14	-14	-8	-8	-8	-8
c_i	-5	-5	0	0	-8	-8	-2	-2
c_{i+1}	-10	-5	-14	-8	-5	0	-8	-2
f_i	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^2	δ_2^2
Profilo	δ_{32}^{17}	δ_{32}^{18}	δ_{32}^{19}	δ_{32}^{20}	δ_{32}^{21}	δ_{32}^{22}	δ_{32}^{23}	δ_{32}^{24}
c_{i-1}	-5	-5	-5	-5	0	0	0	0
c_i	-10	-10	-5	-5	-14	-14	-8	-8
c_{i+1}	-10	-5	-14	-8	-5	0	-8	-2
f_i	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^1
Profilo	δ_{32}^{25}	δ_{32}^{26}	δ_{32}^{27}	δ_{32}^{28}	δ_{32}^{29}	δ_{32}^{30}	δ_{32}^{31}	δ_{32}^{32}
c_{i-1}	-8	-8	-8	-8	-2	-2	-2	-2
c_i	-5	-5	0	0	-8	-8	-2	-2
c_{i+1}	-10	-5	-14	-8	-5	0	-8	-2
f_i	δ_2^1	δ_2^1	δ_2^1	δ_2^1	δ_2^2	δ_2^1	δ_2^2	δ_2^2

Tabella 4.12: Matrice di payoff e valori della funzioni f_i per ogni profilo nell'insieme $U_2(i)$ del NEG omogeneo studiato nell'Esempio 4.3.4.

Profilo	δ_{27}^1	δ_{27}^2	δ_{27}^3	δ_{27}^4	δ_{27}^5	δ_{27}^6	δ_{27}^7	δ_{27}^8	δ_{27}^9
c_1	10	10	10	-1	-1	-1	-1	-1	-1
c_2	10	4.5	4.5	-12	-2	-6.5	15	7	7.5
c_3	10	-12	15	-1	8	-1	-1	-1	0
f_1	δ_3^1	δ_3^1	δ_3^1	δ_3^1	δ_3^1	δ_3^1	δ_3^3	δ_3^3	δ_3^3
f_2	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^2	δ_3^1	δ_3^3	δ_3^3	δ_3^3
f_3	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^2	δ_3^3	δ_3^3	δ_3^3	δ_3^3
Profilo	δ_{27}^{10}	δ_{27}^{11}	δ_{27}^{12}	δ_{27}^{13}	δ_{27}^{14}	δ_{27}^{15}	δ_{27}^{16}	δ_{27}^{17}	δ_{27}^{18}
c_1	-12	-12	-12	8	8	8	-1	-1	-1
c_2	4.5	-1	-1	-2	8	3.5	7	-1	-0.5
c_3	10	-12	15	-1	8	-1	-1	-1	0
f_1	δ_3^1	δ_3^1	δ_3^1	δ_3^2	δ_3^2	δ_3^2	δ_3^3	δ_3^2	δ_3^3
f_2	δ_3^1	δ_3^1	δ_3^3	δ_3^2	δ_3^2	δ_3^2	δ_3^3	δ_3^2	δ_3^3
f_3	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^2	δ_3^2	δ_3^3	δ_3^3	δ_3^3
Profilo	δ_{27}^{19}	δ_{27}^{20}	δ_{27}^{21}	δ_{27}^{22}	δ_{27}^{23}	δ_{27}^{24}	δ_{27}^{25}	δ_{27}^{26}	δ_{27}^{27}
c_1	15	15	15	-1	-1	-1	0	0	0
c_2	4.5	-1	-1	-6.5	3.5	-1	7.5	-0.5	0
c_3	10	-12	15	-1	8	-1	-1	-1	0
f_1	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^2	δ_3^3	δ_3^3	δ_3^3	δ_3^3
f_2	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^2	δ_3^3	δ_3^3	δ_3^3	δ_3^3
f_3	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^2	δ_3^2	δ_3^3	δ_3^3	δ_3^3

Tabella 4.13: Matrice di payoff e valori delle funzioni f_i , $i = 1, 2, 3$ per ogni profilo del NEG studiato nell'Esempio 4.4.1.

Profilo	δ_{27}^1	δ_{27}^2	δ_{27}^3	δ_{27}^4	δ_{27}^5	δ_{27}^6	δ_{27}^7	δ_{27}^8	δ_{27}^9
c_1	10	10	10	4.5	4.5	4.5	4.5	4.5	4.5
c_2	10	4.5	4.5	-12	-2	-6.5	15	7	7.5
c_3	10	-12	15	4.5	-2	7	4.5	-6.5	7.5
c_u	10	4.5	4.5	10	4.5	4.5	10	4.5	4.5
f_1	δ_3^1	δ_3^1	δ_3^1	δ_3^1	δ_3^1	δ_3^1	δ_3^3	δ_3^3	δ_3^3
f_2	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^1	δ_3^3	δ_3^3	δ_3^3	δ_3^3
f_3	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^1	δ_3^3	δ_3^3	δ_3^3	δ_3^3
Profilo	δ_{27}^{10}	δ_{27}^{11}	δ_{27}^{12}	δ_{27}^{13}	δ_{27}^{14}	δ_{27}^{15}	δ_{27}^{16}	δ_{27}^{17}	δ_{27}^{18}
c_1	-12	-12	-12	-2	-2	-2	-6.5	-6.5	-6.5
c_2	4.5	-1	-1	-2	8	3.5	7	-1	-0.5
c_3	10	-12	15	-1	-2	7	4.5	-6.5	7.5
c_0	4.5	-1	-1	-1	-1	-1	4.5	-1	-1
f_1	δ_3^1	δ_3^1	δ_3^1	δ_3^1	δ_3^2	δ_3^2	δ_3^3	δ_3^3	δ_3^3
f_2	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^2	δ_3^3	δ_3^3	δ_3^3	δ_3^3
f_3	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^2	δ_3^3	δ_3^3	δ_3^3	δ_3^3
Profilo	δ_{27}^{19}	δ_{27}^{20}	δ_{27}^{21}	δ_{27}^{22}	δ_{27}^{23}	δ_{27}^{24}	δ_{27}^{25}	δ_{27}^{26}	δ_{27}^{27}
c_1	15	15	15	7	7	7	7.5	7.5	7.5
c_2	4.5	-1	-1	-6.5	3.5	-1	7.5	-0.5	0
c_3	10	-12	15	4.5	-2	7	4.5	-6.5	7.5
c_0	4.5	-1	-1	4.5	-1	-1	4.5	-1	-1
f_1	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3
f_2	δ_3^1	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3	δ_3^3
f_3	δ_3^1	δ_3^1	δ_3^3	δ_3^1	δ_3^2	δ_3^3	δ_3^3	δ_3^3	δ_3^3

Tabella 4.14: Matrice di payoff e valori delle funzioni f_i , $i = 1, 2, 3$ per ogni profilo del NEG controllato quando $u = \delta_3^1$.

Conclusioni

In questo elaborato sono state introdotte le Boolean networks e le Boolean control networks e sono state approfondite alcune loro proprietà. Per poter descrivere ed analizzare le BNs in spazio di stato si è reso necessario l'utilizzo di uno strumento algebrico relativamente recente, il prodotto semitensoriale, sviluppato da D. Cheng in [1]. Si è esposto un metodo per ottenere la forma algebrica delle BNs partendo dalla loro rappresentazione mediante sistemi di equazione logiche alle differenze del primo ordine ed è stata poi esaminata una procedura inversa per poter ricavare le operazioni logiche che stanno alla base di una rete Booleana descritta in spazio di stato. È stata approfondita la rappresentazione delle reti Booleane mediante grafi di stato che permettono una comprensione immediata ed intuitiva della dinamica della rete; tuttavia data la crescita esponenziale del numero di stati all'aumento dei nodi della rete questa rappresentazione può perdere di chiarezza già per reti Booleane di piccole dimensioni. Il metodo più efficace per lo studio delle proprietà di una BN è stato mediante l'analisi della matrice di transizione, tramite la quale si è resa possibile l'individuazione di punti fissi e cicli e dunque lo studio del comportamento asintotico della rete. Ci si è poi soffermati sull'analisi temporale della rete ricavando dei parametri, quali il tempo di transizione, utili a poter comprendere l'andamento di una BN senza conoscerne le condizioni iniziali. Sono state successivamente approfondite due diverse suddivisioni in classi degli stati, una basata su attrattori e bacini di attrazione e l'altra basata sul concetto di comunicazione tra stati; entrambe avevano lo scopo di dare una descrizione intuitiva della rete, specificando i diversi ruoli degli stati nell'evoluzione della BN. È stata infine analizzata la stabilità delle Boolean networks, ricavando le condizioni sufficienti e necessarie affinché si verifichi; una attenzione particolare è stata dedicata ai cicli globalmente attrattivi e ad una rappresentazione della matrice di transizione atta all'immediata individuazione di questi ultimi, ottenibile mediante una semplice permutazione della base dello spazio di stato.

Passando alle Boolean control networks è stato osservato come esse possano essere considerate dei switched systems controllati dalle sequenze di ingresso. Sono state approfondite le proprietà di maggior interesse per quanto riguarda la comprensione della dinamica di una BCN, ovvero la raggiungibilità e la controllabilità, e per

ciascuna di esse sono state ricavate le condizioni sufficienti e necessarie affinché si verifichino. Un ruolo fondamentale in questo ambito è stato rivestito dalla matrice L_{tot} che rende l'analisi di queste proprietà pressoché immediato. Ci si è poi soffermati sul problema della stabilizzabilità e dunque sulla ricerca di sequenze di ingresso adeguate alla stabilizzazione della rete; si è infine dimostrato come la stabilizzabilità ad un ciclo o un punto di equilibrio implichi l'esistenza di una matrice di retroazione che stabilizza la BCN ad un ciclo o ad uno stato rispettivamente. Si sono poi introdotte le reti logiche a k valori e, una volta ricavata la loro rappresentazione in spazio di stato, è stata evidenziata la loro quasi totale corrispondenza alle reti Booleane, con la conseguente possibilità di applicare i risultati ottenuti per le BNs e BCNs alle k -valued networks e k -valued control networks.

Tra i vari campi di applicazione delle reti Booleane è stata scelta quella della Teoria dei giochi. Dopo una breve presentazione dei fondamenti di tale teoria sono stati introdotti i Networked Evolutionary Games, particolari tipi di giochi che si prestano perfettamente ad essere modellati e studiati mediante reti logiche a k valori. Sono stati poi approfonditi i NEG omogenei, ricavando un risultato che permette di ottenere informazioni sull'evoluzione dell'intero gioco da un'analisi di carattere locale. Sono stati modellati e approfonditi diversi tipi di NEG basati sui FNGs più studiati in letteratura, quali "Il dilemma del prigioniero", per poi terminare con l'analisi di un NEG controllato.

Per quanto riguarda gli sviluppi futuri, un approfondimento delle reti Booleane probabilistiche permetterebbe un'applicazione più proficua della teoria delle catene di Markov. Sempre mediante l'utilizzo di probabilistic BNs sarebbe possibile modellare NEG nei quali i giocatori possono adottare strategie miste, permettendo così una scelta di leggi di aggiornamento delle strategie molto più varia e adattabile a situazioni reali.

Ringraziamenti

Ci sono tante persone che vorrei ringraziare: chi mi è stato accanto, chi mi ha tenuto compagnia, chi mi ha aiutato quando ero in difficoltà, chi mi ha insegnato e tanti altri. Se sono giunto a questo traguardo però, è solo grazie ai miei genitori che con tanti, tanti sacrifici e sostenendomi sempre mi hanno dato la possibilità di arrivare fin qua.

Quindi, grazie mamma, grazie papà! Spero di avervi resi orgogliosi.

Una dedica speciale va a mio nonno Aldo che avrei voluto tanto potesse essere con me in questo giorno così importante.

Bibliografia

- [1] D. Cheng, H. Qi and Z.Li, *Analysis and Control of Boolean Networks. A Semi-tensor Product Approach*, Springer (2011).
- [2] D. Cheng, H. Qi and Y. Zhao, *An Introduction to Semi-Tensor Product of Matrices and Its Applications*, World Scientific (2012).
- [3] D. Cheng, *From Dimension-Free Matrix Theory to Cross-Dimensional Dynamic Systems*, Academic Press (2019).
- [4] E. Fornasini, *Appunti di teoria dei sistemi*, Padova: Libreria Editrice Progetto (2015).
- [5] E. Fornasini, *Appunti di Sistemi Ecologici*, non pubblicato.
- [6] S. Tadelis, *Game Theory: An Introduction*, Princeton (2013).
- [7] D. Cheng and H. Qi, “Controllability and observability of Boolean control networks,” *Automatica*, vol. 45, pp. 1659-1667, (2009).
- [8] E. Fornasini and M. E. Valcher, “On the periodic trajectories of Boolean control networks,” *Automatica*, vol. 49, pp. 1506–1509, (2013).
- [9] E. Fornasini and M. E. Valcher, “Observability, Reconstructibility and State Observers of Boolean Control Networks,” *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1390–1401, (2013).
- [10] E. Fornasini and M. E. Valcher, “Optimal control of Boolean control networks,” *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1258–1270, (2014).
- [11] E. Fornasini and M. E. Valcher, “Recent developments in Boolean networks control,” *Journal of Control and Decision*, 3:1, pp. 1-18, (2016).
- [12] N. Bof, E. Fornasini and M. E. Valcher, “Output feedback stabilization of Boolean control networks,” *Automatica*, vol. 57, pp. 21-28, (2015).

- [13] D. Cheng, "On finite potential games," *Automatica*, vol. 50, pp. 1793-1801, (2014).
- [14] D. Cheng, F. He, H. Qi and T. Xu, "Modeling, Analysis and Control of Networked Evolutionary Games," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2402-2415, (2015).
- [15] Y. Zhou, X. Ding and H. Li, "Strategy Consensus for a Class of Cascading Finite Networked Evolutionary Games," *IEEE: Chinese Automation Congress*, pp. 3390-3394, (2017).
- [16] D. Jiang, "Situation Analysis of Boxed Pigs Games," *IEEE: International Conference on E-Business and E-Government*, (2011).
- [17] M. GE, J. Zhao, H. Xing and J. Wang, "Impact of Social Punishment on Networked Evolutionary Games via Semi-tensor Product Method," *IEEE: 35th Chinese Control Conference*, pp. 165-170, (2016).
- [18] K. Zhang, "A note on stationary stable profiles of networked evolutionary games," *13th IEEE International Conference on Control and Automation*, pp. 209-211, (2017).
- [19] T. Liu and D. Cheng, "Control of Networked Evolutionary Games Using Local Information," *IEEE: 35th Chinese Control Conference*, pp. 10369-10374, (2016).
- [20] J. P. Benoit and V. Krishna, "Finitely repeated games," *Econometrica*, vol. 53, no. 4, pp. 905-922, (1985).
- [21] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, pp. 124-131, (2006).
- [22] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theoretical Biology*, vol. 22, pp. 437-467, (1969).