

# Capitolo 3

## Automi Cellulari

### 3.1 Introduzione

Gli Automi Cellulari (AC) sono modelli di calcolo parallelo la cui evoluzione è regolata da leggi puramente locali. Nella sua definizione essenziale, un AC può essere descritto come uno spazio suddiviso in celle regolari, ognuna delle quali può trovarsi in un numero finito di stati. Ogni cella dell'AC ingloba, infatti, un automa finito, uno dei modelli di calcolo più semplici e noti in Informatica. Al tempo  $t = 0$  le celle sono in uno stato arbitrario e l'AC evolve cambiando gli stati degli automi finiti a passi discreti di tempo applicando simultaneamente a ognuna la stessa legge, o funzione, di transizione. L'input per ciascuna cella è dato dagli stati delle celle vicine e le condizioni di vicinato sono determinate da una relazione geometrica, invariante nel tempo e nello spazio. Si noti che l'AC, in alternativa, può essere visto come un reticolo regolare i cui nodi contengono l'automa finito.

A discapito della loro semplice definizione, gli AC possono dar luogo a comportamenti estremamente complessi [191]. A livello microscopico, infatti, le leggi che regolano la dinamica del sistema sono perfettamente note, ma questo non significa che da esse si possa dedurre in ogni caso il comportamento del sistema a livello macroscopico [30, 171]. In altri termini, la dinamica del sistema emerge in maniera non banale dalla mutua interazione delle sue componenti elementari e, anche a fronte di leggi d'interazione semplici, il comportamento macroscopico può risultare estremamente complesso.

L'equivalenza computazionale con la Macchina di Turing [35, 174] collocano gli AC tra i modelli di calcolo universali. Questo rende teoricamente possibile risolvere con gli AC qualsiasi problema "computabile" dato che tutto ciò che è computabile

è computabile tramite una Macchina di Turing (tesi di Church-Turing), quindi anche tramite un AC.

Nello specifico, gli AC si prestano particolarmente bene alla modellizzazione e simulazione di quei sistemi caratterizzati da numerosi costituenti elementari in mutua interazione. Un esempio particolarmente esplicativo è rappresentato dallo studio del comportamento dei fluidi (considerati a livello microscopico come sistemi di particelle) tramite modelli computazionali noti con il nome di Gas Reticolari [171]. Non meno importanti sono gli studi teorici che vedono gli AC come sistemi di calcolo parallelo [176, 46].

## 3.2 Breve storia degli Automi Cellulari

Nel '47 John von Neumann, matematico americano di origine ungherese, aveva intrapreso lo studio su quali fossero le caratteristiche e la complessità di un sistema che lo rendano capace di autoriproduzione. von Neumann morì prematuramente nel '57 e non ebbe tempo di completare il suo *Theory of self reproducing automata*, uscito comunque postumo nel '66 a cura di A. W. Burks [182]. La strada inizialmente intrapresa fu quella di un modello continuo, basato su un sistema di equazioni differenziali, per descrivere uno spazio in cui fluttuavano liberamente una sorta di robot assemblatore e innumerevoli copie dei pezzi di cui egli stesso era composto; il robot era programmato ad “agganciare” i pezzi nello spazio e ad assemblarli opportunamente per costruire una copia di se stesso. Le difficoltà a gestire un tale complesso modello portarono von Neumann nel '51, su suggerimento di Stanislaw Ulam, a cambiare radicalmente approccio: una scacchiera infinita, le cui celle quadrate inglobavano un automa finito (detto poi automa elementare), sostituì lo spazio tridimensionale continuo con uno bidimensionale discreto; il tempo divenne anch'esso discreto nel senso che le celle cambiavano stato in maniera sincrona in un “passo di calcolo”, in relazione al proprio stato e a quello delle celle “vicine”, dove la “vicinanza” era definita da una relazione geometrica fissa nello spazio e nel tempo (una vicinanza a croce fu quella scelta da von Neumann). Lo stato della cella individuava lo stato funzionale di un pezzo dell'assemblatore oppure era “quiescente”, cioè individuava una porzione di spazio inattiva che poteva assumere una funzionalità (passare dallo stato quiescente a uno diverso) solo se “stimolata” da una o più celle vicine non in stato quiescente; le funzioni di transizione degli automi finiti erano le stesse dappertutto nello spazio e non variavano

nel tempo. A questo punto l'assemblatore diventava un modificatore degli stati quiescenti delle celle, da cui era circondato, fino a costruire una copia di se stesso.

Successivamente agli studi di von Neumann sull'autoriproduzione, Codd [35] e Thatcher [174] hanno studiato le proprietà computazionali degli AC, dimostrandone, tra l'altro, l'equivalenza con la Macchina di Turing.

Grande interesse intorno agli AC si ebbe negli anni '70 grazie al Gioco della Vita (Life) di John Horton Conway [71]. Si tratta di un AC bidimensionale con celle quadrate e vicinato della cella costituito dalla cella stessa e dalle 8 celle che la circondano. Ogni cella può assumere solo due stati: lo stato morto (stato 0) e lo stato vivo (stato 1). La funzione di transizione è data dalle seguenti regole: 1) una cella nello stato vivo passa allo stato morto se è in contatto con meno di due celle nello stato vivo (muore per isolamento), oppure se è in contatto con più di tre celle in stato vivo (muore per sovrappopolazione); 2) una cella nello stato morto passa allo stato vivo (nasce) se è in contatto con esattamente tre celle nello stato vivo. Applicando queste semplici regole la popolazione di celle vive evolve continuamente assumendo configurazioni imprevedibili: in qualche caso la popolazione si estingue, in altri raggiunge configurazioni stabili o oscillanti, in altri ancora dà vita a strutture, i glider, in grado di muoversi nello spazio cellulare e interagire, in modo anche molto complesso, con altri glider o con altre strutture stabili o oscillanti. L'emergenza di comportamenti tanto complessi e imprevedibili in un sistema estremamente semplice come Life misero in luce le enormi potenzialità degli AC.

Alcune tra le ricerche teoriche e applicative più interessanti sugli AC per gli argomenti discussi in questo lavoro sono sinteticamente descritti nel paragrafo 3.6.

### 3.3 Definizione informale di Automa Cellulare

Una prima definizione informale di Automa Cellulare, quale si può dedurre dal primo lavoro di von Neumann, può essere data elencandone le proprietà fondamentali.

**Definizione 3.3.1 (Definizione informale di Automa Cellulare).** Un Automa Cellulare è caratterizzato dalle seguenti proprietà fondamentali:

- è formato da uno spazio  $d$ -dimensionale suddiviso in celle regolari (triangoli, quadrati, esagoni, cubi, ecc.) o, equivalentemente, da un reticolo regolare  $d$ -dimensionale;
- il numero di stati della cella è finito;

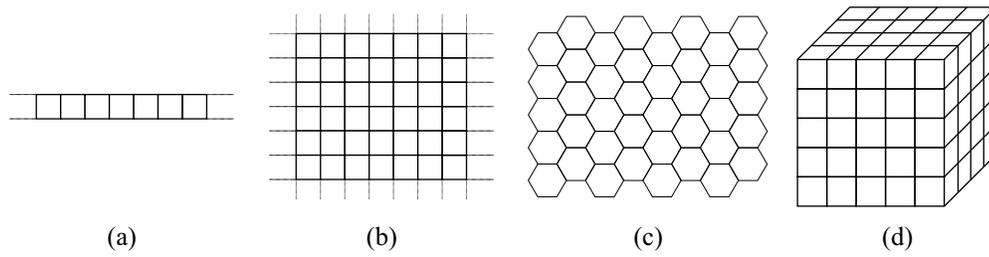


Figura 3.1: Esempi di spazi cellulari (a) unidimensionale, (b) bidimensionale con celle quadrate, (c) bidimensionali con celle esagonali e (d) tridimensionale con celle cubiche.

- l'evoluzione avviene a passi discreti;
- ogni cella cambia di stato simultaneamente a tutte le altre in accordo alla stessa regola di transizione;
- la regola di transizione dipende dallo stato della cella stessa e dallo stato delle celle vicine;
- la relazione di vicinanza è locale, uniforme e invariante nel tempo.

### 3.3.1 Dimensione e geometria dell'Automa Cellulare

La definizione di Automa Cellulare richiede, dunque, la discretizzazione dello spazio in celle. Per gli automi cellulari unidimensionali l'unica possibilità è una sequenza di celle allineate una a fianco all'altra, ovvero un reticolo unidimensionale. Per automi cellulari di dimensioni superiori esistono diverse alternative; per automi cellulari bidimensionali, per esempio, si possono adottare spazi cellulari con celle triangolari, quadrate o esagonali, mentre per automi cellulari tridimensionali si scelgono, solitamente, celle cubiche. La figura 3.1 illustra alcuni esempi di spazi cellulari in una, due e tre dimensioni.

Per quanto riguarda gli automi cellulari bidimensionali, sebbene la tassellazione quadrata sia facilmente rappresentabile attraverso una matrice e non presenti problemi nella rappresentazione grafica (per esempio ogni elemento della matrice può essere visualizzato utilizzando un pixel della matrice dello schermo), in alcune applicazioni può presentare problemi di anisotropia (l'argomento è ripreso più nei dettagli nel paragrafo 3.6.2). Quando questo si verifica si preferisce adottare una tassellazione esagonale che, per AC bidimensionali, è quella con anisotropia più

bassa [190], può rendere le simulazioni più realistiche e, in alcuni casi, è indispensabile per modellare correttamente alcuni fenomeni [184]. Purtroppo non esiste un equivalente tridimensionale dello spazio cellulare esagonale; come si vedrà (ancora nel paragrafo 3.6.2) questo può richiedere il ricorso a spazi di dimensione quattro.

### 3.3.2 Numero di stati della cella

Il numero di stati della cella deve essere finito ed è determinato in relazione al particolare contesto di studio o d'applicazione. Nei primi studi teorici successivi a von Neumann che hanno visto gli AC come modelli computazionali astratti [35, 174] il numero di stati della cella era, solitamente, abbastanza piccolo. Con soli due stati, per esempio, è possibile rappresentare, nella configurazione iniziale (specificazione dello stato, in questo caso 0 o 1, di tutte le celle dello spazio cellulare al tempo  $t = 0$ ), l'informazione che l'AC deve elaborare.

Anche quando l'automa cellulare è utilizzato per descrivere sistemi di particelle e modellarne le interazioni, il numero di stati è abbastanza contenuto [171, 184].

Al contrario, quando si studiano sistemi che possono trovarsi in un continuum di possibili stati, può essere necessario un numero di stati della cella abbastanza grande perchè il modello sia significativo [59].

### 3.3.3 Relazione di vicinanza

La relazione di vicinanza della cella, cui ci si riferirà con il nome di cella centrale, dipende dalla geometria delle celle. Secondo la definizione data, deve godere delle seguenti proprietà: 1) deve essere locale, cioè deve coinvolgere solo un numero limitato di celle in prossimità della cella centrale; 2) deve essere omogenea, cioè la stessa per ogni cella dello spazio cellulare; 3) deve essere invariante nel tempo.

Per AC unidimensionali si è soliti riferirsi al vicinato in termini di raggio,  $r$ , che definisce un vicinato composto da  $n = 2r + 1$  celle [189]. Per esempio, un raggio  $r = 1$  identifica un vicinato di  $n = 2r + 1 = 3$  celle: la cella centrale, la cella adiacente a sinistra e la cella adiacente a destra. La figura 3.2 illustra due esempi di vicinato con raggio  $r = 1$  ed  $r = 2$  per un automa cellulare unidimensionale.

Nel caso di AC bidimensionali con tassellazione quadrata i vicinati più utilizzati sono quello di von Neumann e quello di Moore. Il primo è composto dalla cella centrale e dalle celle a nord, est, ovest e sud, mentre il secondo contiene anche quelle di nord-ovest, nord-est, sud-ovest e sud-est. Un tipico vicinato per automi cellulari bidimensionali esagonali è, invece, composto dalle celle a nord, nord-est, sud-est,

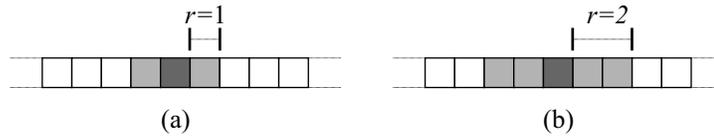


Figura 3.2: Esempio di vicinato con raggio (a)  $r = 1$  e (b)  $r = 2$  per un automa cellulare unidimensionale. Le celle in grigio scuro identificano la cella centrale, quelle in grigio chiaro le vicine.

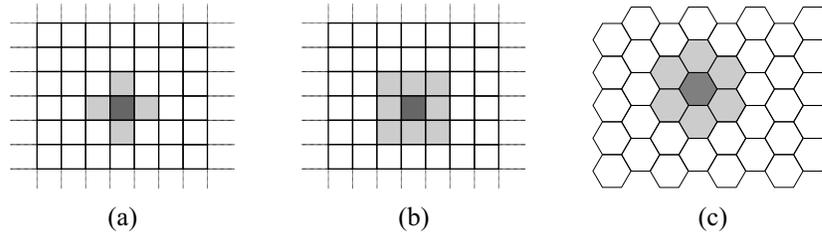


Figura 3.3: Vicinati di von Neumann (a) e di Moore (b) per un automa cellulare bidimensionale con tassellazione quadrata e vicinato esagonale (c) per un automa cellulare bidimensionale con tassellazione esagonale. Le celle in grigio scuro identificano la cella centrale, quelle in grigio chiaro le vicine.

sud, sud-ovest e nord-ovest. La figura 3.3 illustra (a) il vicinato di von Neumann e (b) quello di Moore per automi cellulari con tassellazione quadrata e (c) il tipico vicinato per automi cellulari con tassellazione esagonale. Ovviamente è possibile definire relazioni di vicinanza differenti da quelle illustrate. Nella simulazione della diffusione di gas in un ambiente, ad esempio, è possibile utilizzare la relazione di vicinanza di Margolus<sup>1</sup> [176].

### 3.3.4 Funzione di transizione di stato della cella

A ogni passo dell'AC, la funzione di transizione è applicata simultaneamente a tutte le celle dello spazio cellulare, determinando il nuovo stato di ognuna in funzione

<sup>1</sup>La relazione di vicinanza di Margolus non gode della proprietà d'invarianza temporale. Infatti il vicinato della cella cambia a seconda del passo, pari o dispari, dell'AC. Ai passi pari il vicinato è formato dalla cella centrale e dalle celle a nord, est, e nord-est; ai passi dispari, invece, il vicinato è formato dalla cella centrale e dalle celle a sud, ovest, e sud-ovest. Si noti, tuttavia, che un AC che utilizzi tale relazione di vicinanza è perfettamente "legale". E' infatti possibile dimostrare che, dato un AC con relazione di vicinanza di Margolus, si può costruire un AC perfettamente equivalente che soddisfi tutte le proprietà richieste dalla definizione.

dello stato delle celle del vicinato. In tal modo la computazione dell'AC assume caratteristiche di parallelismo e di decentralizzazione.

Quando il numero di stati è piccolo, si è soliti definire la regola di transizione tramite una tabella (look-up table) che specifica il nuovo stato della cella centrale per ogni possibile configurazione del vicinato [191]. Al contrario, quando il numero di stati dell'AC è troppo grande, la funzione di transizione viene solitamente definita tramite l'esplicitazione di un algoritmo [59].

## 3.4 Definizione formale di Automa Cellulare

In letteratura è possibile incontrare numerose definizioni di AC, ognuna delle quali si adatta meglio delle altre ai particolari contesti d'applicazione. La definizione formale presentata in questo paragrafo coincide con la definizione di AC omogeneo deterministico. Questo, del resto, non è limitativo poichè la maggior parte delle altre definizioni possono essere a essa ricondotte.

Come accennato in precedenza, ogni cella dell'AC ingloba un identico automa finito. Nel caso specifico dell'AC omogeneo deterministico, si tratta di un automa finito anch'esso omogeneo e deterministico. E' opportuno, quindi, premetterne la definizione.

### 3.4.1 L'automato finito

L'automato finito (af), o automa a stati finiti, è probabilmente il modello di calcolo più semplice in informatica. Intuitivamente, un af è un sistema che può trovarsi in un numero finito di stati differenti e, come conseguenza di qualche ingresso, può effettuare una transizione da uno stato a un altro.

Come nel caso degli AC, in letteratura è possibile incontrare numerose definizioni di af. L'af presentato di seguito corrisponde alla definizione di af come riconoscitore (accettore) di linguaggi [72]. Il modello di af utilizzato negli AC, presentato subito dopo, è un modello ulteriormente semplificato che prende il nome di automa elementare.

**Definizione 3.4.1 (Definizione formale di automa finito deterministico come riconoscitore di linguaggi).** Un automa finito deterministico come riconoscitore di linguaggi è formalmente definito come una quintupla

$$af = \langle Q, I, \sigma, q_0, F \rangle$$

dove:

$Q$  è l'insieme finito degli stati dell'af;

$I$  è l'insieme finito dei simboli d'ingresso (input);

$\sigma : I \times Q \rightarrow Q$  è la funzione di transizione che modifica gli stati in funzione dell'input;

$q_0 \in Q$  è lo stato iniziale dell'af;

$F \subset Q$  è l'insieme degli stati finali dell'af.

**Definizione 3.4.2.** Sia  $I^*$  l'insieme di tutte le possibili stringhe che si possono costruire sull'insieme dei simboli d'ingresso  $I$ . Il linguaggio  $L$  accettato dall'af è definito nel seguente modo:

$$L = \{x \in I^* \mid \sigma(q_0, x) \in F\}$$

dove  $\sigma(q_0, x)$  indica lo stato in cui si trova l'af al termine della computazione. Pertanto una stringa  $x \in I^*$  è accettata dall'af se questo, dopo aver letto tutta la stringa, si trova in uno stato finale.

*Esempio 3.4.1.* La figura 3.4 illustra un af che riconosce le stringhe binarie di lunghezza qualsiasi in cui compaiano almeno tre simboli 0 consecutivi. In questo caso  $I = \{0, 1\}$ ,  $Q = \{q_i, q_1, q_2, q_f\}$ ,  $q_0 = q_i$  ed  $F = \{q_f\}$ . Al passo  $t = 0$  l'af si trova nello stato iniziale  $q_i$  e viene letto il primo simbolo della stringa. La funzione di transizione  $\sigma$  è applicata tante volte quanti sono i simboli della stringa d'ingresso e determina il nuovo stato dell'af in base allo stato attuale e all'input. Esaurito l'input, la stringa è riconosciuta se e solo se l'af si trova nello stato finale  $q_f$ . La stringa  $x_1 = 100110$  non è riconosciuta dall'af. Infatti la sequenza di transizioni di stato  $\sigma(q_i, x_1)$  non termina nello stato finale  $q_f$ :

$$q_i \xrightarrow{\sigma(1, q_i)} q_i \xrightarrow{\sigma(0, q_i)} q_1 \xrightarrow{\sigma(0, q_1)} q_2 \xrightarrow{\sigma(1, q_2)} q_i \xrightarrow{\sigma(1, q_i)} q_i \xrightarrow{\sigma(0, q_i)} q_1$$

La stringa  $x_2 = 100011$  è invece riconosciuta dall'af poichè la sequenza di transizioni di stato  $\sigma(q_i, x_2)$  termina nello stato finale  $q_f$ :

$$q_i \xrightarrow{\sigma(1, q_i)} q_i \xrightarrow{\sigma(0, q_i)} q_1 \xrightarrow{\sigma(0, q_1)} q_2 \xrightarrow{\sigma(0, q_2)} q_f \xrightarrow{\sigma(1, q_f)} q_f \xrightarrow{\sigma(1, q_f)} q_f$$

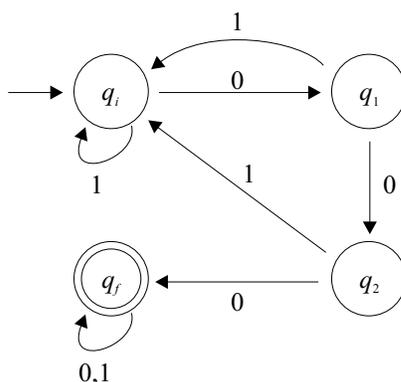


Figura 3.4: Automa finito che riconosce le stringhe binarie in cui compaiano almeno tre simboli 0 consecutivi. I nodi indicano gli stati dell'af, gli archi le transizioni. All'inizio della computazione, l'af si trova nello stato iniziale  $q_i$ ; se viene letto il simbolo d'ingresso 0, effettua una transizione verso lo stato  $q_1$ ; al contrario, se viene letto il simbolo 1, l'af rimane nello stato iniziale  $q_i$ . L'af riconosce la stringa in ingresso se alla fine della computazione si trova nello stato finale  $q_f$ .

Le celle dell'AC inglobano un tipo di af, detto elementare (ae), che è un'ulteriore semplificazione del semplice modello appena descritto: non è rilevante se l'ae termina il suo calcolo in uno stato finale o in qualsiasi altro stato. Ciò che importa è il cambiamento di stato in sè e la distinzione tra l'insieme degli stati  $Q$  e l'insieme degli stati finali  $F \subset Q$  non è presente. L'insieme finito dei simboli d'ingresso è implicitamente definito dal numero di stati dell'ae e dal numero di celle del vicinato.

**Definizione 3.4.3 (Definizione formale di automa elementare).** L'automa elementare è formalmente definito come una tripla

$$a = \langle Q, I, \sigma \rangle$$

dove:

$Q$  è l'insieme finito degli stati dell'ae;

$I \equiv Q^n$  è l'insieme finito dei simboli d'ingresso, essendo  $n$  il numero di celle del vicinato;

$\sigma : I \rightarrow Q$  è la funzione di transizione che modifica gli stati in funzione dell'input.

### 3.4.2 L'Automa Cellulare Omogeneo Deterministico

A questo punto, dopo aver premesso le definizioni di automa finito e di automa elementare, si può definire formalmente l'AC omogeneo deterministico.

**Definizione 3.4.4 (Definizione formale di Automa Cellulare Omogeneo Deterministico).** L'Automa Cellulare Omogeneo Deterministico è formalmente definito come una quadrupla

$$A = \langle \mathbb{Z}^d, Q, X, \sigma \rangle$$

dove:

$\mathbb{Z}^d = \{i \equiv (i_1, i_2, \dots, i_d) \mid i_k \in \mathbb{Z} \forall k = 1, 2, \dots, d\}$  è l'insieme dei punti del reticolo  $d$ -dimensionale che definisce lo spazio cellulare dell'AC;  $\mathbb{Z}$  è l'insieme dei numeri interi;

$Q$  è l'insieme finito degli stati dell'automa elementare;

$X = \{\xi_0, \xi_1, \dots, \xi_{m-1}\}$  è l'insieme finito degli  $m$  vettori  $d$ -dimensionali

$$\xi_j = \{\xi_{j_1}, \xi_{j_2}, \dots, \xi_{j_d}\}$$

che definiscono l'insieme

$$V(X, i) = \{i + \xi_0, i + \xi_1, \dots, i + \xi_{m-1}\}$$

delle coordinate delle celle vicine alla generica cella  $i$  di coordinate  $(i_1, i_2, \dots, i_d)$ .  $X$  è detto indice o relazione di vicinanza<sup>2</sup>.

$\sigma : Q^m \rightarrow Q$  è la funzione di transizione deterministica dell'automa elementare.

*Esempio 3.4.2.* Si consideri un automa cellulare bidimensionale con vicinato di von Neumann (figura 3.3a) e la cella  $i$  di coordinate  $(7, 7)$ . La relazione di vicinanza che definisce il vicinato di von Neumann è

$$X = \{\xi_0, \xi_1, \xi_2, \xi_3, \xi_4\} = \{(0, 0), (0, -1), (1, 0), (0, 1), (-1, 0)\}$$

---

<sup>2</sup>Per il seguito imponiamo che  $\xi_0$  sia il vettore nullo e, di conseguenza, che ogni cella appartenga al proprio vicinato. Ci si riferirà a tale cella con il nome di cella centrale. Tuttavia è senz'altro pensabile un AC in cui la cella non faccia parte del proprio vicinato.

Pertanto, l'insieme delle coordinate delle celle vicine alla cella  $i$  sono definite dall'insieme

$$\begin{aligned} V(X, i) &= \{i + \xi_0, i + \xi_1, i + \xi_2, i + \xi_3, i + \xi_4\} = \{(7, 7) + (0, 0), \\ &(7, 7) + (0, -1), (7, 7) + (1, 0), (7, 7) + (0, 1), (7, 7) + (-1, 0)\} = \\ &= \{(7, 7), (7, 6), (8, 7), (7, 8), (6, 7)\} \end{aligned}$$

**Definizione 3.4.5.** Sia  $C = \{c | c : \mathbb{Z}^d \rightarrow Q\}$  l'insieme delle possibili assegnazioni di stato ad  $A$  (insieme delle configurazioni dell'AC) e  $c(i)$  lo stato della cella  $i$  nella configurazione  $c$ . La funzione di transizione globale dell'AC è definita come segue:

$$\begin{aligned} \tau : C &\longrightarrow C \\ c &\mapsto \tau(c) \end{aligned}$$

essendo

$$\tau(c)(i) = \sigma(c(V(X, i))) = \sigma(c(i + \xi_0), c(i + \xi_1), \dots, c(i + \xi_{m-1}))$$

**Definizione 3.4.6.** Si definisce stato quiescente uno stato  $q_0 \in Q$  tale che

$$\sigma(q_0, q_0, \dots, q_0) = q_0$$

.

## 3.5 Studi teorici sugli Automi Cellulari

Di seguito sono illustrate alcune interessanti ricerche teoriche sugli AC. Poichè la maggior parte di esse riguardano gli AC unidimensionali è utile, per una migliore comprensione degli argomenti trattati, premettere alcune definizioni e convenzioni ormai divenute di uso comune.

### 3.5.1 Automi Cellulari unidimensionali

Gli AC più semplici, almeno dal punto di vista della loro costruzione, sono gli AC elementari [189]. Si tratta di AC unidimensionali di  $N$  celle con  $k = 2$  stati (0 e 1), raggio del vicinato  $r = 1$  e condizioni periodiche al contorno (lo spazio cellulare unidimensionale è visto come un anello in cui la prima e l'ultima cella sono adiacenti). La figura 3.5 illustra un esempio di AC con condizioni periodiche

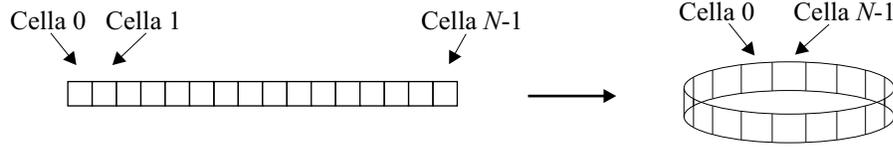


Figura 3.5: Esempio di automa cellulare unidimensionale con condizioni periodiche al contorno. La prima e l'ultima cella dello spazio cellulare risultano vicine nella rappresentazione ad anello.

al contorno. La scelta di adottare uno spazio cellulare così fatto nasce, in generale, dall'impossibilità di gestire spazi cellulari infiniti. Una soluzione ad anello, pur essendo finita, definisce comunque uno spazio illimitato in cui l'AC può evolversi.

La regola di transizione  $\sigma$  della cella è espressa sotto forma di tabella (look-up table). Per esempio, se si indica con  $\eta$  una generica configurazione del vicinato (il numero di configurazioni del vicinato è dato da  $k^{2r+1} = k^n$ , nel caso degli AC elementari  $2^3 = 8$ ), la seguente regola di transizione determina il nuovo stato,  $s = \sigma(\eta)$ , della cella centrale:

$\eta$	000	001	010	011	100	101	110	111
$s$	0	0	1	1	0	1	1	0

Si noti che le otto possibili configurazioni del vicinato sono elencate in ordine crescente secondo i valori binari che rappresentano. Infatti il numero binario 000 corrisponde al numero decimale 0, il numero binario 001 corrisponde al numero decimale 1, e così via; il numero binario 111 corrisponde, infine, al numero decimale 7. Adottando questa convenzione, una qualsiasi regola di transizione per AC elementari può essere definita semplicemente elencando i nuovi stati della cella centrale. La precedente regola può essere definita, pertanto, nel seguente modo:

$\sigma_{00110110} \equiv$	000	001	010	011	100	101	110	111
	0	0	1	1	0	1	1	0

Anche in questo caso la regola di transizione  $\sigma$  definisce un numero binario. Pertanto si è soliti riferirsi alle regole di transizione degli AC elementari attraverso il numero decimale corrispondente al numero binario definito dalla regola. Per esempio:

$$\begin{aligned} \sigma_{00000000} &\equiv \sigma_0 \\ \dots & \\ \sigma_{00110110} &\equiv \sigma_{54} \\ \dots & \\ \sigma_{11111111} &\equiv \sigma_{255} \end{aligned}$$

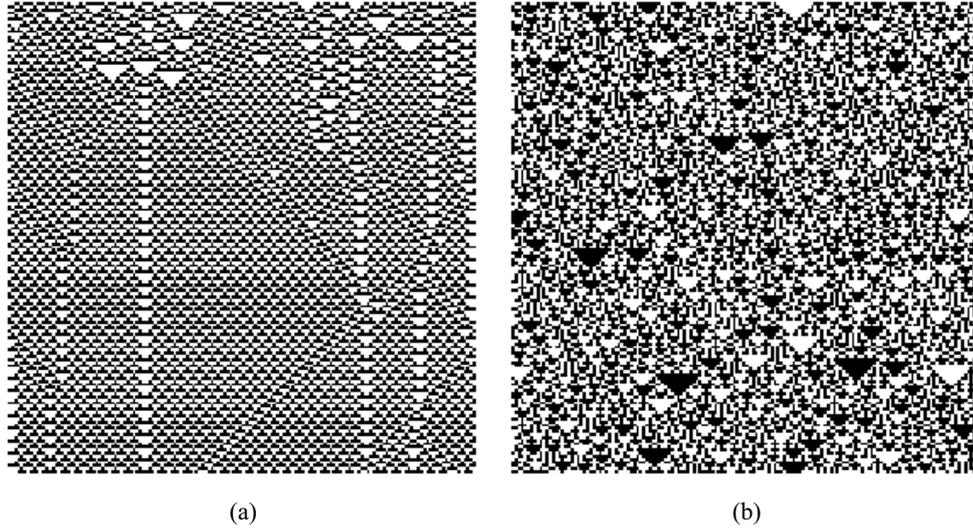


Figura 3.6: I primi 100 passi di calcolo degli AC elementari (a)  $\sigma_{54}$  e (b)  $\sigma_{150}$ . La configurazione iniziale (prima riga dei due grafici) è generata in maniera casuale in modo che ogni cella possa assumere stato 0 (colore bianco) o 1 (colore nero) con eguale probabilità. L'evoluzione temporale è visualizzata dall'alto verso il basso come sequenza di righe.

La figura 3.6 illustra due esempi di AC elementari.

Quando  $r > 1$  il numero di configurazioni del vicinato cresce rapidamente. Per esempio, per  $(k, r) = (2, 2)$  allora  $k^{2r+1} = k^n = 2^5 = 32$ . Di conseguenza, il numero complessivo di regole di transizione diviene  $2^{32} = 4294967296$ , rendendone di fatto impossibile un'analisi esaustiva. Lo stesso vale se si considerano AC con  $(k, r) = (3, 1)$ : in tal caso il numero di possibili regole di transizione è  $3^{27}$ . In questi casi si suole restringere il campo d'indagine a particolari sottoclassi, come gli AC legali totalistici [189].

Un AC è considerato legale se preserva lo stato quiescente (solitamente lo stato nullo), cioè se lo stato della cella centrale di un vicinato in cui tutte le celle siano nello stato quiescente al passo  $t$  rimane nello stato quiescente al passo  $t + 1$ . Gli AC totalistici sono, invece, quegli AC caratterizzati dal fatto che il nuovo stato della cella dipende esclusivamente dalla somma dei valori degli stati delle celle del vicinato al passo precedente. In tal modo, per esempio, le regole legali totalistiche per AC con  $k$  stati e  $r = 1$  passano da  $k^{k^3}$  a  $k^{[k^2(1+k)-1]/2}$ . Quindi, considerando  $k = 2$  si passa da 256 a 32, mentre per  $k=3$  si passa da  $3^{27}$  a  $3^{17}$  regole legali totalistiche. Tuttavia, anche il numero di regole legali totalistiche cresce molto rapidamente con

$k$  ed  $r$  e per questo lo studio delle proprietà degli AC si è concentrata per lo più su AC con pochi stati e vicinati di poche celle.

Come per gli AC elementari, anche gli AC totalistici possono essere specificati tramite un numero,  $c$ , detto codice della regola. Il codice  $c$  è definito in modo tale che il coefficiente di  $2^i$  nella sua rappresentazione binaria sia il valore atteso per lo stato della cella centrale quando la somma degli stati delle celle del vicinato sia pari a  $i$ . Per esempio, per  $k = 2$  ed  $r = 2$  il codice

$$c = 20 = 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

definisce la seguente regola totalistica  $\sigma_{c=20}$  sotto forma di look-up table:

$$\sigma_{c=20} \equiv \begin{array}{c} \Sigma \\ s \end{array} \begin{array}{cccccc} 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{array}$$

essendo  $\Sigma$  la somma degli stati delle celle del vicinato ed  $s$  il nuovo stato della cella centrale.

### 3.5.2 Classi di complessità e computazione universale

Wolfram [189, 191] ha proposto una classificazione degli AC unidimensionali con pochi stati e vicinati di poche celle in base al loro comportamento qualitativo, individuando quattro classi di complessità:

**classe 1** gli AC della classe 1, indipendentemente dallo stato iniziale, convergono ad uno stato finale uniforme;

**classe 2** gli AC della classe 2 convergono verso stati finali in cui semplici strutture rimangono sempre uguali o si ripetono nell'arco di pochi passi di calcolo;

**classe 3** gli AC della classe 3 sono caratterizzati da un comportamento estremamente caotico;

**classe 4** gli AC della classe 4 presentano sia comportamenti ordinati sia caotici. In questa classe si osservano semplici strutture localizzate che, tuttavia, possono interagire tra loro in maniera estremamente complessa.

Benchè altri schemi di classificazione siano stati successivamente proposti, per esempio da Chatè e Manneville [27], da Gutowitz [77] e da McIntosh [125], quello di Wolfram è sicuramente il più noto. Esempi di AC appartenenti alle quattro classi di complessità di Wolfram sono illustrati in figura 3.7.

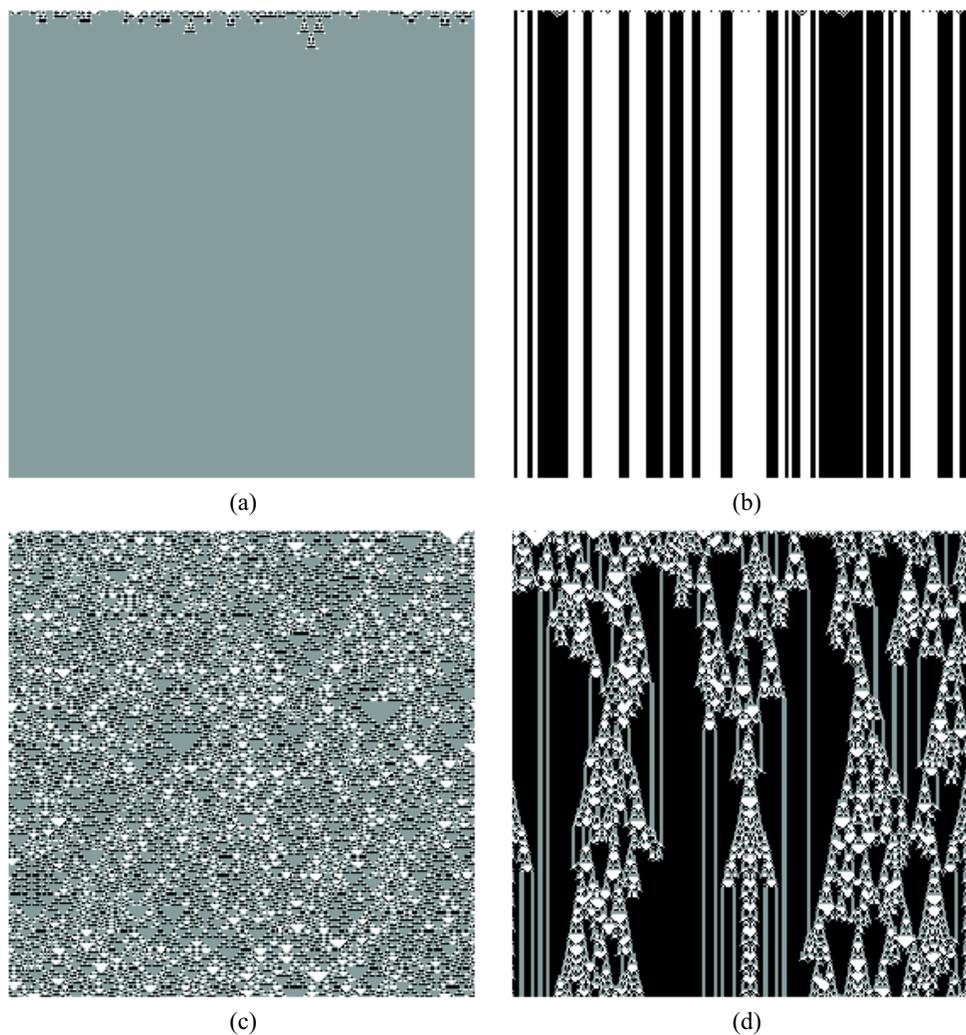


Figura 3.7: I primi 250 passi di calcolo degli AC legali totalistici  $k = 3$ ,  $r = 1$  (a)  $\sigma_{c=1014}$ , (b)  $\sigma_{c=1008}$ , (c)  $\sigma_{c=1020}$  e (d)  $\sigma_{c=2043}$ . Gli AC illustrati appartengono rispettivamente alle classi di complessità 1, 2, 3 e 4. La configurazione iniziale, composta da 250 celle, è generata in maniera casuale in modo che ogni cella possa assumere stato 0 (colore bianco), 1 (colore grigio) o 2 (colore nero) con eguale probabilità. L'evoluzione temporale è visualizzata dall'alto verso il basso come sequenza di righe.

Tra le classi di complessità, la classe 4 è risultata particolarmente interessante per la presenza di strutture (glider) in grado di propagarsi nello spazio e nel tempo, tanto da aver spinto Wolfram ad avanzare l'ipotesi che gli AC di tale classe possano essere capaci di computazione universale.

Un calcolatore può essere visto come un sistema in grado di trasformare una sequenza iniziale di bit in una sequenza finale codificante il risultato della computazione. La sequenza iniziale può essere considerata come la codifica dell'informazione che deve essere elaborata e del programma che deve essere eseguito, mentre la sequenza finale, o parte di essa, può essere considerata come risultato della computazione. Così, gli AC studiati da Wolfram possono essere visti come calcolatori: le configurazioni iniziali codificano dati e programma; le configurazioni finali (ottenute tramite un numero sufficiente di passi di calcolo) codificano il risultato della computazione. Se gli AC analizzati da Wolfram fossero capaci di computazione universale, dunque, con una prefissata regola di transizione dovrebbe essere possibile rappresentare nella configurazione iniziale ogni possibile programma.

In effetti, l'universalità computazionale degli AC è nota sin dalla loro nascita. La costruzione di von Neumann di una macchina in grado di autoriprodursi si basa, infatti, sulla dimostrazione dell'esistenza di un calcolatore/costruttore universale in un AC con 29 stati [182]. In seguito, Codd [35], Thatcher [174], Berlekamp, Conway e Guy [20] e Fredkin e Toffoli [67], solo per citarne alcuni, hanno dimostrato l'universalità computazionale per più semplici AC.

L'ipotesi di universalità computazionale per gli ancora più semplici AC studiati da Wolfram nasce dall'osservazione che i glider possano operare come "elaboratori" dell'informazione codificata nella configurazione iniziale. Per mezzo dei glider lo stato di una cella in una particolare posizione dello spazio cellulare può, infatti, influenzare nel tempo gli stati di celle in posizioni arbitrariamente lontane (figura 3.7d). Inoltre i glider possono interagire tra loro in maniera estremamente complessa e tramite essi è teoricamente possibile riprodurre, così come dimostrato per il Gioco della Vita di John Horton Conway [20, 71], le porte logiche fondamentali di un computer digitale universale. E' noto, dunque, che alcuni AC siano capaci di computazione universale, ma rimane questione aperta quale sia quel particolare AC capace di computazione universale con il più piccolo numero di stati e di vicini possibile.

La possibilità di computazione universale implica che ogni computazione può essere eseguita tramite AC. Inoltre, i meccanismi di processazione dell'informazione osservati in molti processi naturali appaiono più vicini agli AC che alle Macchine di Turing. Per questo motivo tali processi possono essere simulati più efficientemente

tramite AC che non tramite altri modelli di calcolo [191]. Se si pensa, infine, che con le conoscenze tecnologiche attuali un semplice AC unidimensionale universale con svariati milioni di celle può essere facilmente realizzato su un singolo wafer di silicio con tempo di clock bassissimo, dell'ordine dei miliardesimi di secondo, si capisce che la questione relativa alla progettazione di hardware dedicato potrebbe aprire nuove prospettive in termini di *performance* per la simulazione di quei fenomeni (fisici, biologici, economici, ecc.) che ben si prestano a essere descritti in termini di Automi Cellulari [189].

### 3.5.3 Il margine del caos

L'ipotesi di Wolfram secondo cui semplici AC unidimensionali siano capaci di computazione universale è stata ripresa successivamente da Chris Langton. In un lavoro del 1990 dal titolo "Computation at the edge of chaos: phase transition and emergent computation" [112], Langton ha dimostrato che un'opportuna parametrizzazione dello spazio delle regole permette di individuare con una certa approssimazione sia le relazioni tra le classi di complessità, sia le zone di tale spazio in cui si collocano gli AC delle varie classi.

Per parametrizzare lo spazio delle regole, Langton ha utilizzato il parametro  $\lambda$  [111] che misura la percentuale di transizioni non quiescenti nella funzione di transizione dell'AC. Il parametro  $\lambda$  è definito nel seguente modo:

$$\lambda = \frac{k^n - n_q}{k^n} = 1 - \frac{n_q}{k^n}$$

essendo  $k$  il numero di stati della cella,  $n = 2r + 1$  il numero di celle del vicinato ed  $n_q$  il numero di transizione che terminano nello stato quiescente. Se  $n_q = k^n$ , tutte le transizioni della look-up table sono verso lo stato quiescente e  $\lambda = 0$ ; se  $n_q = 0$ , invece, non esistono transizioni verso lo stato quiescente e  $\lambda = 1$ ; infine, quando nella look-up table tutti gli stati sono rappresentati nella stessa misura  $\lambda = 1 - 1/k$ .

*Esempio 3.5.1.* Si consideri un AC con  $k = 2$  stati in cui la regola di transizione mappa la metà dei  $2^n$  vicinati nello stato 0 e l'altra metà nello stato 1. Di conseguenza  $n_q = 2^n/2$ , quindi

$$\lambda = 1 - \frac{n_q}{k^n} = 1 - \frac{\frac{1}{2}2^n}{2^n} = 1 - \frac{1}{2} = 0.5$$

Se, invece,  $k = 4$  e la regola di transizione mappa esattamente un quarto dei  $4^n$

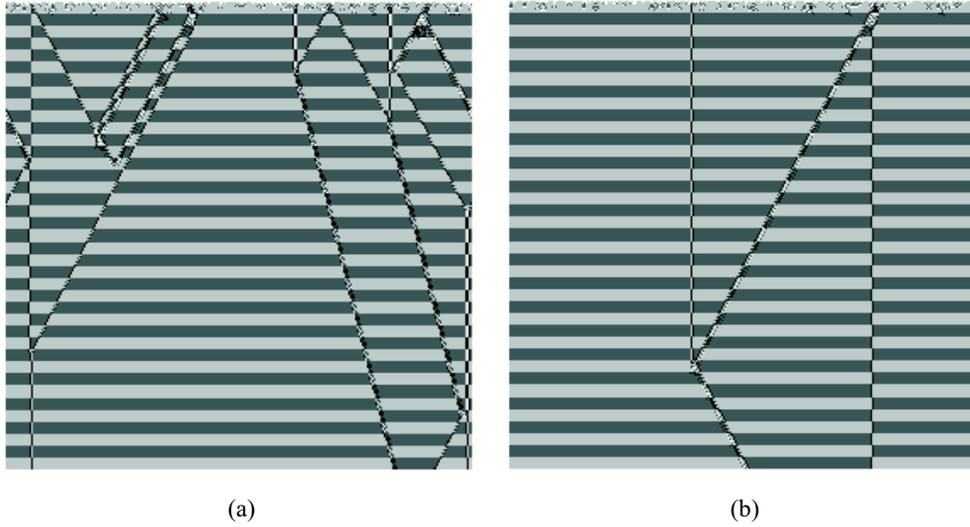


Figura 3.8: Esempio di AC al “margine del caos”. Le figure (a) e (b) mostrano l’evoluzione dello stesso AC con  $k = 4$  stati ed  $r = 1$  a partire da due differenti configurazioni iniziali. Le tonalità di grigio rappresentano i quattro possibili stati della cella, dal bianco per lo stato 0, al nero per lo stato 3.

vicinati in ognuno degli stati 0, 1, 2 e 3, allora  $n = \frac{1}{4}4^n$  e

$$\lambda = 1 - \frac{n_q}{k^n} = 1 - \frac{\frac{1}{4}4^n}{4^n} = 1 - \frac{1}{4} = 0.75$$

I valori  $\lambda = 0$  e  $\lambda = 1 - 1/k$  rappresentano rispettivamente le regole di transizione più omogenee e più eterogenee che possono essere definite utilizzando  $k$  stati.

Langton ha analizzato il comportamento di numerosi AC legali totalistici con  $k = 4$  stati ed  $r = 1$  al variare di  $\lambda$  nell’intervallo  $[0, 0.75]$ . I risultati hanno mostrato che per valori piccoli di  $\lambda$  il comportamento degli AC analizzati è essenzialmente ordinato, tipico delle classi di complessità 1 e 2, mentre per valori grandi di  $\lambda$  il comportamento osservato è essenzialmente caotico, tipico della classe 3. Tra queste due zone (ordine e caos), tuttavia, Langton ne ha osservata una terza molto piccola in prossimità del valore  $\lambda = 0.45$ . In questa zona, definita dallo stesso Langton “margine del caos”, la dinamica dell’AC è in grado di generare sia strutture statiche, sia strutture in grado di propagarsi nello spazio e nel tempo, tipiche della classe 4 di Wolfram. La figura 3.8 illustra un esempio di un AC al “margine del caos”. A differenza della prime due zone, solo al margine del caos l’informazione codificata nella configurazione iniziale dell’AC può propagarsi su lunghe distanze,

condizione necessaria perchè si possa parlare di computazione. Inoltre, come si può anche notare in figura 3.8, nei pressi del margine del caos la dinamica degli AC dipende sensibilmente dalla configurazione iniziale. In tal modo, in accordo con l'ipotesi di Wolfram, l'AC è in grado di "discriminare" tra differenti informazioni (programma e dati) codificate nella stringa iniziale, rendendo così potenzialmente possibile un'effettiva elaborazione.

### 3.5.4 Meccanica Computazionale

Lo studio sulla classificazione degli AC di Wolfram si colloca nell'ambito di un'analisi di tipo puramente macroscopico poichè si fonda esclusivamente sull'osservazione delle configurazioni spazio-temporali generate dagli AC, disinteressandosi completamente delle cause per cui tali configurazioni emergono. Diametralmente all'opposto, il tentativo di Langton di caratterizzare lo spazio delle regole attraverso il parametro  $\lambda$  si colloca all'interno di un approccio prettamente microscopico poichè si fonda esclusivamente sull'analisi delle interazioni locali definite dalla regola di transizione.

Entrambi gli approcci possono essere soggetti a critiche. Per esempio, lo schema di classificazione di Wolfram è soggettivo e, di conseguenza, differenti osservatori potrebbero non concordare sulla classe di appartenenza di particolari AC. Il lavoro di Langton, invece, si pone in qualche modo in antitesi rispetto al noto principio secondo cui in sistemi non lineari l'equazione che definisce il sistema (la regola di transizione nel caso degli AC) non determina direttamente il comportamento del sistema a lungo termine [46]. E' nota, infatti, la non assoluta precisione di  $\lambda$ : non tutte le regole al margine del caos risultano effettivamente complesse e, viceversa, regole complesse possono collocarsi anche al di fuori di tale zona. In considerazione di questo, Wuensche [194, 192, 193] ha proposto il parametro  $Z$  che meglio sembra descrivere il comportamento degli AC a partire dall'analisi della regola di transizione.

La Meccanica Computazionale, proposta da Crutchfield [43] e applicata agli AC dallo stesso Crutchfield e da Hanson [80, 44, 78, 79], è un'approccio che si colloca a un livello intermedio tra il microscopico e il macroscopico nello studio della computazione emergente nei sistemi dinamici discreti. Applicata agli AC, l'obiettivo della Meccanica Computazionale è scoprire un'appropriata configurazione base a partire dalla quale descrivere le componenti strutturali emergenti fondamentali e le loro interazioni.

Una configurazione base consiste in un insieme di linguaggi formali  $\Lambda = \{\Lambda^i, i =$

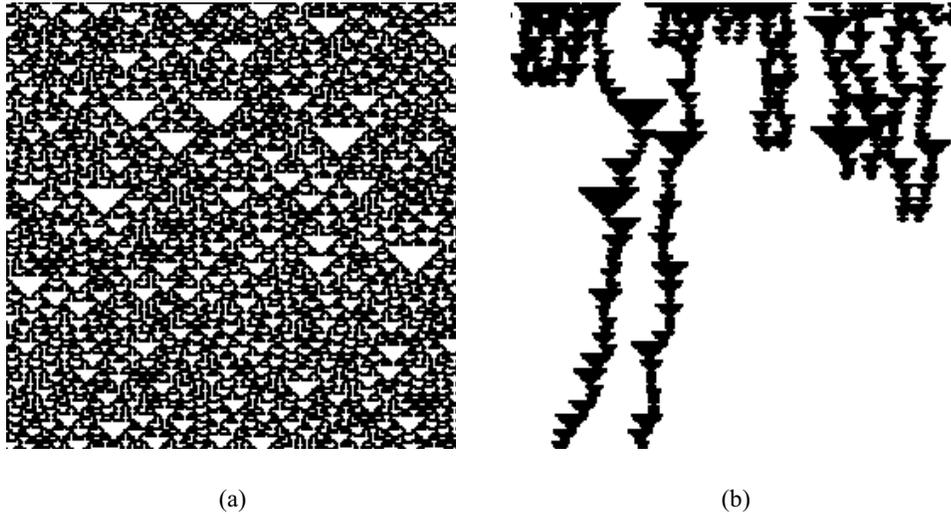


Figura 3.9: Diagramma spazio-temporale (a) dell’AC elementare  $\sigma_{18}$  e (b) dello stesso AC in cui sono state “filtrate” le istanze del dominio regolare  $\Lambda^0$  ed evidenziate in nero le particelle emergenti.

$0, 1, \dots\}$ . Per esempio, la configurazione base dell’AC elementare  $\sigma_{18}$ , illustrato in figura 3.9a, è  $\Lambda = \{\Lambda^0 = (0*)^+\}$ , dove, nel contesto specifico della Meccanica Computazionale, il simbolo  $*$  è un simbolo jolly che può assumere sia il valore 0 che il valore 1, mentre l’apice  $+$  indica una sequenza pari di ciò che è contenuto tra parentesi. In altri termini, in molte regioni nelle configurazioni spazio-temporali dell’AC  $\sigma_{18}$  gli stati delle celle sono istanze del linguaggio  $\Lambda^0$ , per esempio  $0101\dots 01$ , oppure  $0000\dots 00$ .

Crutchfield e Hanson definiscono dominio regolare  $\Lambda^j$  una regione della configurazione spazio-temporale dell’AC invariante sia rispetto allo spazio che al tempo tale che sia un linguaggio regolare. Individuare un dominio regolare<sup>3</sup> significa, almeno in parte, comprendere la dinamica dell’AC. I domini regolari definiscono, infatti, motivi di “regolarità” nelle corrispondenti zone dei diagrammi spazio-temporali.

I domini possono essere filtrati mettendo in evidenza le zone di demarcazione. Tali zone sono, di fatto, glider che, nella terminologia tipica della Meccanica Computazionale, prendono il nome di particelle. Le particelle che emergono fil-

<sup>3</sup>I domini regolari sono spesso individuabili tramite l’osservazione diretta delle configurazioni spazio-temporali degli AC. Per i casi in cui questo risulti difficile, come per l’AC  $\sigma_{18}$ , Crutchfield e Hanson [43, 78] hanno sviluppato un metodo automatico per il riconoscimento dei domini regolari, chiamato  $\epsilon$  - *machine*.

<b>Domini Regolari</b>
$\Lambda^0 = \{(0*)^+\}$ ( $*$ = 0 oppure $*$ = 1)
<b>Particelle</b>
$\alpha \sim \mathbf{P} = \{1(00)^n1, n = 0, 1, 2, \dots\}$
<b>Interazioni</b>
$\alpha + \alpha \rightarrow \emptyset$ ( $\Lambda^0$ )

Tabella 3.1: Catalogo dell'AC elementare  $\sigma_{18}$ . Sono riportati i domini regolari, le particelle emergenti e le loro interazioni.

trando l'unico dominio regolare  $\Lambda^0$  dell'AC  $\sigma_{18}$  sono istanze del linguaggio  $\mathbf{P} = \{1(00)^n1, n = 0, 1, 2, \dots\}$ . La figura 3.9b illustra la configurazione spazio-temporale dell'AC  $\sigma_{18}$  in cui le celle di  $\Lambda^0$  sono colorate in bianco, mentre le celle di  $\mathbf{P}$  sono colorate in nero.

Per l'analisi del comportamento dell'AC, domini e particelle sono riportate in un catalogo. Nel caso dell'AC  $\sigma_{18}$  il catalogo è particolarmente semplice (tabella 3.1).

Il risultato è che l'AC  $\sigma_{18}$  può essere analizzato in maniera strutturale, piuttosto che semplicemente definirlo caotico secondo la classificazione di Wolfram. Le particelle emergenti portano con sè vari tipi d'informazione relativi alle regioni della configurazione iniziale dell'AC da cui provengono. Le loro interazioni, invece, fungono da elaboratori d'informazione.

### 3.5.5 Algoritmi Genetici e computazione emergente negli Automi Cellulari

Crutchfield e Mitchell [45, 46] hanno applicato gli AG alla ricerca di AC unidimensionali in grado di eseguire compiti complessi che coinvolgano la coordinazione globale tra celle arbitrariamente lontane.

Un tipico esempio è costituito dal problema  $\rho_0 > \rho_c$  che consiste nel valutare se nella configurazione iniziale dell'AC la frazione  $\rho_0$  di celle nello stato 1 è maggiore di un valore prefissato  $\rho_c$ . Solitamente  $\rho_c = 1/2$  e il problema si riduce a determinare se nella configurazione iniziale ci siano più celle nello stato 1 o nello stato 0. Nel primo caso l'obbiettivo per l'AC è convergere, in un prefissato numero di passi  $T_{max}$ , nella configurazione finale in cui le celle siano tutte nello stato 1, nel secondo caso nella configurazione finale in cui le celle siano tutte nello stato 0. Ovviamente il compito è indefinito nel caso in cui sia  $\rho_0 = \rho_c$ . La *performance*  $P_N^I(\sigma)$  di un

AC definito dalla regola di transizione  $\sigma$  è calcolata considerando  $I$  configurazioni iniziali di  $N$  celle generate casualmente, iterando l'AC per al più  $T_{max}$  passi e determinando la frazione di volte in cui la configurazione iniziale è classificata correttamente. Il compito  $\rho_c = 1/2$ , banale per modelli computazionali come la Macchina di Turing, è tutt'altro che semplice per un AC poichè non esiste un controllo globale che possa contare il numero di celle in un dato stato, anzi ogni cella può interagire soltanto con poche altre celle vicine.

Per i 256 AC elementari, la massima *performance* è all'incirca  $P_N^{10^4} = 0.5$  per  $N \in \{149, 599, 999\}$ . Per AC  $(k, r) = (2, 2)$  la massima *performance* è all'incirca  $P_N^{10^4} = 0.58$  per  $N = 149$  e  $P_N^{10^4} = 0.5$  per  $N \in \{599, 999\}$ . Per AC  $(k, r) = (2, 3)$  le *performance* osservate sono decisamente superiori e alcune strategie di risoluzione del problema particolarmente interessanti.

L'AG utilizzato per la ricerca di AC  $(k, r) = (2, 3)$  in grado di risolvere il compito  $\rho_c = 1/2$  è di tipo binario. Il genotipo codifica la regola di transizione dell'AC (soluzione candidata del problema) considerando i  $k^{2r+1} = 2^{2 \cdot 3 + 1} = 128$  bit di output della look-up table. Lo spazio di ricerca per l'AG contiene, in tal modo,  $2^{128}$  soluzioni candidate e risulta troppo ampio per una ricerca esaustiva. Una popolazione iniziale di  $M$  genotipi è selezionata in maniera casuale. La fitness  $F_N^I(\sigma)$  di un genotipo che rappresenta l'AC definito dalla regola di transizione  $\sigma$  è calcolata considerando  $I$  configurazioni iniziali di  $N$  celle generate casualmente, iterando l'AC per al più  $T_{max}$  passi e determinando la frazione di volte in cui la configurazione iniziale viene classificata correttamente. In ognuna delle  $G$  generazioni dell'AG, (1) è generato un nuovo insieme  $I$  di configurazioni iniziali dell'AC, (2) è calcolato il valore della funzione di fitness  $F_N^I(\sigma)$  per ogni genotipo della popolazione, (3) i genotipi vengono ordinati in base al valore di fitness, (4) un numero  $E$  dei migliori è copiato nella nuova popolazione senza modifiche (schema steady-state elitistico), (5) i rimanenti  $M - E$  genotipi sono ottenuti tramite crossover a singolo punto (con probabilità  $p_c$ ) e mutazione (con probabilità  $p_m$ ) sugli  $E$  genotipi migliori; lo schema adottato è con rimpiazzamento, cioè un cromosoma elitario può essere scelto più volte per la ricombinazione. I valori dei parametri adottati negli esperimenti brevemente descritti sotto sono:  $M = 100$ ,  $I = 100$ ,  $E = 20$ ,  $N = 149$ ,  $T_{max} = 2N$ ,  $p_c = 1.0$ ,  $p_m = 0.016$  e  $G = 100$ .

In ognuno dei 300 esperimenti eseguiti, l'AG ha evidenziato tre differenti strategie per la risoluzione del compito  $\rho_c = 1/2$ : default, espansione di blocchi, particellare. Tra queste la più interessante, e al contempo più performante, è risultata la strategia particellare. La figura 3.10 mostra un esempio di un AC,  $\sigma_{par}$ , che adotta la strategia particellare per risolvere il compito di classificazione  $\rho_c = 1/2$ . Le sue

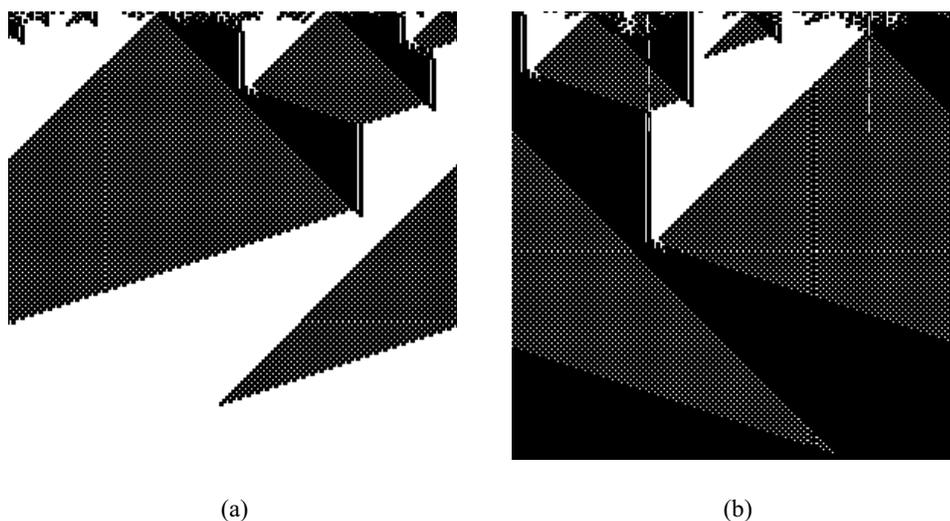


Figura 3.10: Diagrammi spazio-temporali di un AC,  $\sigma_{par}$ , evoluto tramite AG che risolve il problema di classificazione  $\rho_c = 1/2$  adottando la strategia particellare: la figura (a) mostra l'evoluzione di una configurazione iniziale con  $\rho_0 = 0.48$ ; la figura (b) mostra l'evoluzione di una configurazione iniziale con  $\rho_0 = 0.51$ .

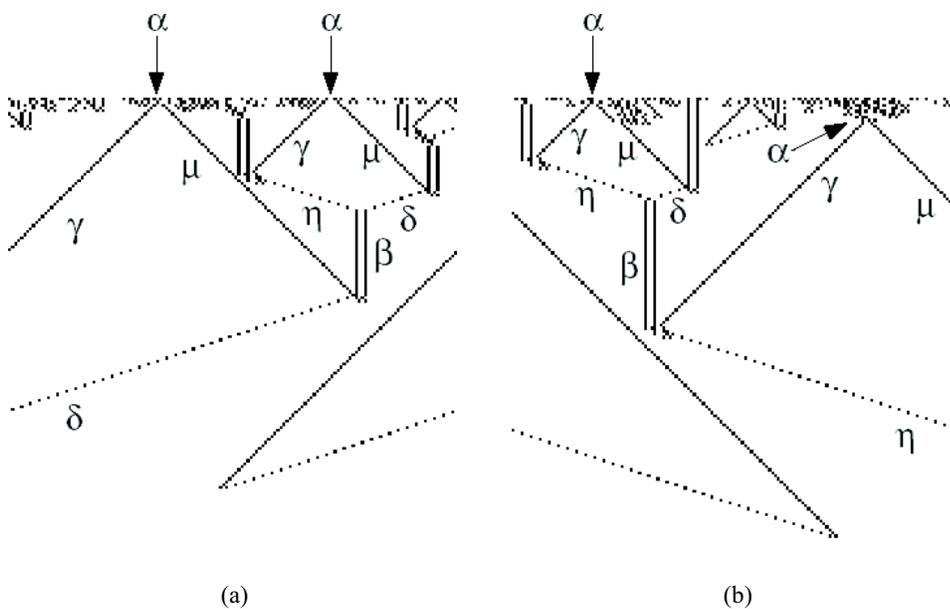


Figura 3.11: Diagrammi spazio-temporali filtrati dei domini regolari dell'AC  $\sigma_{par}$ , che risolve il problema di classificazione  $\rho_c = 1/2$  adottando la strategia particellare: la figura (a) mostra l'evoluzione di una configurazione iniziale con  $\rho_0 = 0.48$ ; la figura (b) mostra l'evoluzione di una configurazione iniziale con  $\rho_0 = 0.51$ .

<b>Domini Regolari</b>
$\Lambda^0 = \{0^+\}, \Lambda^1 = \{1^+\}, \Lambda^2 = \{(011)^+\}$
<b>Particelle</b>
$\alpha \sim \Lambda^1\Lambda^0, \beta \sim \Lambda^0\Lambda^1, \gamma \sim \Lambda^1\Lambda^2,$ $\delta \sim \Lambda^2\Lambda^1, \eta \sim \Lambda^0\Lambda^2, \gamma \sim \Lambda^2\Lambda^0$
<b>Interazioni</b>
$\alpha \rightarrow \gamma + \mu, \beta + \gamma \rightarrow \eta, \mu + \beta \rightarrow \delta,$ $\eta + \delta \rightarrow \beta, \eta + \mu \rightarrow \emptyset, \gamma + \delta \rightarrow \emptyset$

Tabella 3.2: Catalogo dell'AC elementare  $\sigma_{par}$ . Sono riportati i domini regolari, le particelle emergenti e le loro interazioni.

*performace* sono  $P_{149}^{104}(\sigma_{par}) = 0.775$ ,  $P_{599}^{104}(\sigma_{par}) = 0.740$  e  $P_{999}^{104}(\sigma_{par}) = 0.728$ .

Nella fase iniziale, la strategia dell'AC  $\sigma_{par}$  consiste nel mappare regioni ad alta densità di celle nello stato 1 in sottoconfigurazioni di celle tutte nello stato 1, e viceversa. Il bordo tra tali regioni e le regioni a scacchiera possono essere interpretate come segnali indicanti una situazione d'ambiguità. Le regioni con densità locale pari proprio a  $\rho_c = 1/2$ , dunque, non sono classificate direttamente, ma tale compito è demandato al comportamento e all'interazione di tali segnali.

Per comprendere con precisione la strategia dell'AC  $\sigma_{par}$ , può essere utile ricorrere agli strumenti propri della Meccanica Computazionale, descritta nel paragrafo precedente. La configurazione base di  $\sigma_{par}$ , visibile in figura 3.10, è formalmente definita come  $\Lambda = \{\Lambda^0 = 00^+, \Lambda^1 = 11^+, \Lambda^2 = (01)^+\}$ , corrispondente alle regioni tutto bianco, tutto nero e a scacchiera. Filtrando il dominio regolare definito da  $\Lambda$  (figura 3.11) si notano differenti particelle, etichettate con lettere greche. La tabella 3.2 illustra il catalogo per l'AC  $\sigma_{par}$ . La dinamica dell'AC può essere ora interpretata in termini di particelle, che trasportano l'informazione sulle densità locali delle regioni da cui hanno avuto origine. Le loro interazioni combinano e processano l'informazione: ad esempio si osserva che quando due particelle si scontrano sopravvive quella proveniente dalla zona della configurazione iniziale a densità più alta. La sequenza di urti particellari è dunque responsabile della classificazione della configurazione iniziale. La computazione, in definitiva, è effettuata a livello intermedio tra il microscopico e il macroscopico, cioè al livello particellare.

### 3.5.6 Altri lavori teorici sugli Automi Cellulari

Gli studi illustrati in questo capitolo non esauriscono l'insieme delle ricerche teoriche sugli AC. Dopo la loro introduzione, infatti, numerosi contributi sono venuti da ricercatori da tutte le parti del mondo. Il problema della reversibilità negli AC è stato studiato, solo per citarne alcuni autori, da Moore [130], Myhill [135], Di Gregorio e Trautteur [61], Kari [101], e da Toffoli e Margolus [177]. Sulla scia dei lavori di Crutchfield, Hanson e Mitchell sulla Meccanica Computazionale e sull'evoluzione di AC in grado di eseguire compiti che richiedono coordinamento globale negli AC, Jiménez-Morales ha adottato un approccio evolutivo basato su AG per lo studio di comportamenti collettivi non banali negli AC [98, 97, 99], mentre Tomassini e Vemzi [179] hanno studiato il problema di classificazione  $\rho_c = 1/2$  per AC asincroni in cui l'aggiornamento dello stato delle celle non avviene simultaneamente. Altri interessanti lavori riprendono, sulla scia dell'originario studio di von Neumann, il problema dell'autoriproduzione negli AC. Tra essi si ricordano i lavori di Azpeitia e Ibáñez [14] e di Bilotta et al. [22]. Roli e Zambonelli [159] hanno studiato, infine, l'emergenza di strutture nella dinamica spazio-temporale di AC dissipativi, cioè di AC visti come sistemi aperti in cui l'ambiente può influenzarne la dinamica.

## 3.6 Alcune applicazioni degli Automi Cellulari

In campo applicativo, gli AC si prestano particolarmente bene alla modellizzazione e simulazione di alcune classi di sistemi complessi, caratterizzati dall'interazione di numerosi componenti "elementari". L'ipotesi, largamente diffusa nella scienza tradizionale, secondo cui se il comportamento di un sistema è complesso il modello che lo descrive deve essere necessariamente di pari complessità, lascia il posto all'idea secondo cui il suo comportamento possa essere descritto, almeno in certi casi, in termini estremamente semplici, attraverso la specificazione delle leggi che definiscono le interazioni locali tra le componenti base del sistema [191].

In alcuni settori l'applicazione degli AC ha dato risultati paragonabili, se non superiori, a quelli ottenuti tramite approcci più tradizionali. Un esempio particolarmente significativo è l'applicazione alla modellizzazione del comportamento dei fluidi turbolenti attraverso modelli noti con il nome di Gas Reticolari e di modelli di Boltzmann su reticolo. Un altro importante campo d'applicazione è la "Vita Artificiale", disciplina che si occupa dello studio della vita e del comportamento nei sistemi naturali e artificiali. Negli ultimi anni, inoltre, gli AC sono stati applicati con successo nella modellizzazione di fenomeni naturali macroscopici. Di

seguito sono brevemente descritti alcuni esempi applicativi in Vita Artificiale e nel settore dei Gas Reticolari e dei modelli di Boltzmann su reticolo. L'applicazione degli AC alla modellizzazione di fenomeni naturali macroscopici è, invece, discussa più approfonditamente nel capitolo successivo.

### 3.6.1 Automi Cellulari nella Vita Artificiale: il problema dell'auto-riproduzione

La Vita Artificiale può essere definita come la disciplina che si occupa della vita e del comportamento di sistemi artificiali che “vivono” in ambienti artificiali. Chris Langton [113], uno dei suoi padri fondatori, definisce il nuovo settore di ricerca come:

“[...] lo studio di sistemi *manufatti* che esibiscono comportamenti caratteristici dei sistemi viventi. La Vita Artificiale amplia la Biologia classica, che si occupa dell'analisi degli organismi viventi, tentando di *sintetizzare* comportamenti simili alla vita (like-life behaviors) nei computer o in altri media artificiali. [...] la Vita Artificiale può contribuire in Biologia Teorica nell'inquadrare la *vita così come la conosciamo* nel più ampio contesto della *vita come potrebbe essere*”.

Di seguito è brevemente illustrato il filone di ricerca della Vita Artificiale, in cui gli AC hanno giocato un ruolo di primo piano, che si occupa del problema dell'auto-riproduzione<sup>4</sup>.

Fu lo stesso Langton a suggerire che gli AC potessero rappresentare un modello estremamente efficace per gli studi in Vita Artificiale [110]. In effetti, uno dei suoi precursori fu proprio John von Neumann che, già a partire dalla fine degli anni '40, si era dedicato allo studio dell'auto-riproduzione degli organismi viventi adottando un approccio “artificiale” basato sugli AC.

Successivamente, prima Codd, alla fine degli anni '60, e poi Langton, intorno alla metà degli anni '80, hanno proposto modelli semplificati rispetto all'originario modello di auto-riproduzione di von Neumann con strutture autoreplicanti. von Neumann era convinto che l'auto-riproduzione dovesse essere definita in modo tale da inglobare la proprietà di computabilità universale; questo spiega la grande complessità del suo modello, che può essere visto essenzialmente a due differenti livelli:

---

<sup>4</sup>Per un approfondimento più generale sulla Vita Artificiale e sulle sue applicazioni si vedano i proceedings delle conferenze *Artificial Life* che, a partire dalla prima tenutasi a Los Alamos nel 1987, si ripetono con cadenza biennale.

1) l'AC vero e proprio, con 29 stati per la cella e una complicatissima funzione di transizione; 2) un “costruttore universale” (il robot assemblatore, ovvero una Macchina di Turing Universale) inglobato nello spazio cellulare.

Codd, pur condividendo l'ipotesi di von Neumann, riteneva che la complessità del suo modello fosse troppo elevata e ne propose uno alternativo con soli 8 stati [35]. Solo nel 1984, però, Langton propose un modello con strutture autoreplicanti (i Langton's Loop) non equivalenti a una Macchina di Turing dal punto di vista computazionale, dimostrando così che la proprietà di computabilità universale è sufficiente ma non necessaria all'autoriproduzione [109].

Bisogna, tuttavia, sottolineare una fondamentale differenza tra il punto di vista di Langton e quello di von Neumann sull'autoriproduzione. Come accennato in precedenza, von Neumann riteneva la proprietà di computabilità universale necessaria per ipotesi. Langton fece, tuttavia, notare come semplici organismi viventi, per i quali non sia stato possibile dimostrare la proprietà di computabilità universale, fossero capaci di autoriprodursi. Rinunciare all'universalità computazionale non implicava dunque, secondo Langton, perdere le caratteristiche fondamentali del processo dell'autoriproduzione. Questo, tuttavia, significava accettare i casi di autoriproduzione banali in cui il fenomeno si origina non per le proprietà intrinseche dell'autoriproduttore ma per effetto della regola di transizione dell'AC (la regola di transizione di un AC elementare che assegna a ogni configurazione del vicinato la somma modulo 2 degli stati delle celle fornisce esempi di autoriproduttori banali). Per tale motivo Langton definì come fenomeni “autentici” di autoriproduzione solo quei fenomeni che dipendono principalmente dalle proprietà intrinseche dell'autoriproduttore e solo in minima parte, o per nulla, dalla “fisica” del sistema [109].

I modelli precedentemente descritti replicano particolari strutture (per esempio il Langton's Loop) definite nella configurazione iniziale dell'AC grazie a una funzione di transizione specificamente progettata in modo da operare efficacemente sulla particolare configurazione e non su altre. Chou e Reggia [33] hanno dimostrato, per la prima volta, che è possibile realizzare AC con funzioni di transizione più “generaliste” in grado di far emergere strutture autoreplicanti da configurazioni iniziali del tutto casuali. Tali strutture possono avere caratteristiche e forme differenti e interagire, in modalità anche molto complesse, con altre strutture che contestualmente emergono nello spazio cellulare.

Come già accennato, altri interessanti lavori che riprendono l'originario studio di von Neumann sul problema dell'autoriproduzione negli AC sono dovuti ad Azpeitia e Ibáñez [14] e a Bilotta et al. [22]. In particolare, quest'ultimo lavoro

si colloca, in una certa misura, sulla linea tracciata da Chou e Reggia: strutture autoreplicanti emergono da configurazioni casuali e possono mutuamente interagire nello spazio cellulare. La novità metodologica sta nel fatto che la ricerca di tali strutture non è mirata, ma è un effetto diretto della ricerca di regole di transizione complesse dell'AC. La ricerca di tali regole è eseguita tramite Algoritmi Genetici, la cui funzione di fitness premia quelle regole che risultano maggiormente complesse secondo la definizione di complessità proposta da Wuensche [193]. Se da una parte è vero che la computabilità universale è solo condizione sufficiente ma non necessaria all'autoriproduzione [109], i risultati di questo studio sembrano indicare che la complessità, così come proposta da Wuensche, invece lo sia. Successivi studi hanno, tuttavia, mostrato che anche regole non complesse possono far emergere strutture autoreplicanti nella dinamica dell'AC con la conseguenza che anche la condizione di complessità risulta non necessaria al fenomeno dell'autoriproduzione [149].

Come si evince dal filone di ricerca sull'autoriproduzione, dunque, la Vita Artificiale ha prodotto ipotesi di lavoro e risultati originali e di estremo interesse, sia dal punto di vista teorico che da quello delle possibili applicazioni. In questo contesto, gli AC hanno giocato, e continuano a giocare, un ruolo di primo piano.

### 3.6.2 Gas Reticolari e modelli di Boltzmann su reticolo

La dinamica dei fluidi è quella branca della fisica che si occupa del comportamento di gas e liquidi [108]. La fluidodinamica classica si fonda sulle equazioni di Navier-Stokes che formalizzano le leggi di conservazione della massa e dell'impulso:

$$\begin{aligned}\frac{\partial \rho u}{\partial t} &= -\frac{\partial \rho u u}{\partial x} - \frac{\partial \rho u v}{\partial y} - \frac{\partial p}{\partial x} + \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ \frac{\partial \rho v}{\partial t} &= -\frac{\partial \rho u v}{\partial x} - \frac{\partial \rho v v}{\partial y} - \frac{\partial p}{\partial y} + \mu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)\end{aligned}$$

essendo  $u$  e  $v$  le velocità del fluido nelle direzioni  $x$  e  $y$  di un sistema di riferimento cartesiano ortogonale,  $\rho$  la densità del fluido,  $p$  la pressione esercitata sul fluido e  $t$  il tempo. Il primo membro rappresenta la variazione rispetto al tempo della quantità di moto per unità di volume. Tale variazione è dovuta a tre effetti: il flusso dovuto al campo di moto del fluido (i primi due termini al secondo membro), detto termine di convezione, le forze di pressione (terzo termine) e la dissipazione (quarto e quinto termine). La non linearità (dovuta al termine di convezione), causa principale della

difficoltà di risoluzione per casi non idealizzati [171], ha spinto alcuni ricercatori a tentare un approccio alternativo allo studio della fluidodinamica basato sugli AC che ha preso il nome di Gas Reticolari.

### Gas Reticolari

L'idea base dei Gas Reticolari è modellare un fluido attraverso un sistema di particelle ognuna delle quali è vincolata a muoversi, con velocità costante, solo lungo le direzioni di un reticolo discreto. Le leggi locali sono definite in modo da garantire l'invarianza del numero di particelle (conservazione della massa) e la conservazione della quantità di moto.

I primi a proporre un approccio di questo tipo sono stati, nel 1976, Hardy, Pomeau e de Pazzis [81]. Il modello HPP (dalle iniziali dei tre ricercatori) è basato su una griglia quadrata in cui ciascun nodo è collegato a quattro vicini (vicinato di von Neuman, figura 3.3a). Ogni nodo può contenere da un minimo di zero a un massimo di quattro particelle di massa unitaria, ognuna delle quali può spostarsi soltanto lungo una delle direzioni del reticolo con velocità costante, pari a 1 (cioè, ogni particella può spostarsi in un passo dell'AC dalla cella in cui si trova in una cella adiacente). Poichè si è supposta sia la massa che la velocità unitarie, la quantità di moto di una particella può essere definita semplicemente dal verso (i versi consentiti sono Sud, Ovest, Nord ed Est). Pertanto, le possibili configurazioni per ogni nodo dell'HPP sono  $2^4 = 16$ . Per esempio la configurazione 0000 indica l'assenza di particelle, la configurazione 0001 indica la presenza di una particella che si sta spostando verso Est, la configurazione 0010 indica la presenza di una particella che si sta spostando verso Nord, e così via.

Quando due particelle con velocità opposte si trovano nella stessa cella (configurazioni 0101 e 1010), si verifica un urto e le particelle vengono deviate di un angolo retto. In tutti gli altri casi, compresi i casi in cui i cammini si incrociano (per esempio nella configurazione 1111), le particelle proseguono il loro cammino senza deviazioni.

Quando il numero di particelle è sufficientemente grande, su scala macroscopica l'impressione è quella di un fluido continuo. L'AC HPP si è dimostrato capace di modellare correttamente fenomeni fluidodinamici come la propagazione di un'onda, ma il suo comportamento risulta poco realistico nella simulazione di fluidi turbolenti. Il problema principale è dovuto al vincolo reticolare. Quest'ultimo è, infatti, responsabile dei così detti invarianti spuri [171], cioè invarianti che non hanno corrispondenti nel continuo. In pratica l'HPP, oltre a conservare la massa e la quantità

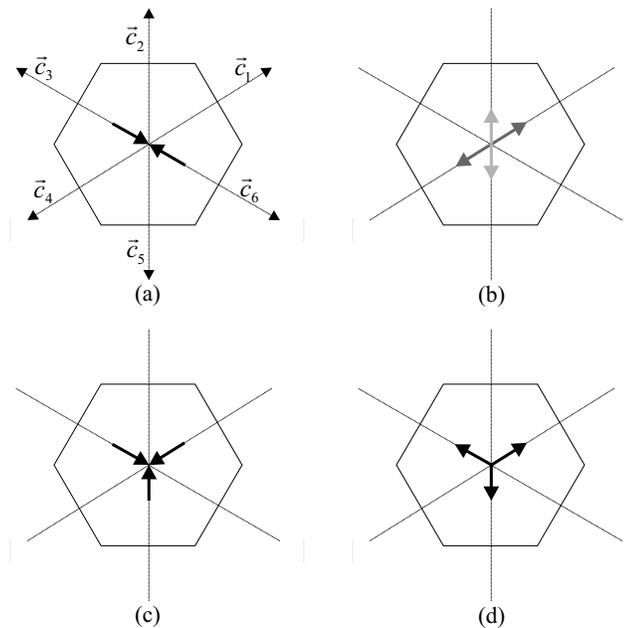


Figura 3.12: Regole di collisione del modello FHP. (a) e (b) descrivono, rispettivamente, la configurazione della cella prima e dopo l’urto nel caso di due particelle; si noti che in questo caso la regola di transizione è probabilistica: le particelle possono, cioè, deviare verso destra (configurazione in grigio chiaro) o sinistra (configurazione in grigio scuro) con la stessa probabilità). (c) e (d) descrivono, rispettivamente, la configurazione della cella prima e dopo l’urto nel caso di tre particelle. Le sei possibili direzioni delle particelle,  $c_1, c_2, \dots, c_6$ , sono evidenziate in (a).

di moto, conserva anche la quantità di moto lungo le righe e le colonne del reticolo, come si può dedurre facilmente dall’analisi delle regole di transizione. A causa di questo eccesso d’invarianti, il moto delle particelle risulta fortemente anisotropo e il fluido non è sufficientemente libero di muoversi e diffondere la sua quantità di moto in tutte le direzioni.

Nel 1986, però, Frish, Hasslacher e Pomeau [70], intuirono che l’impiego di un reticolo a maglia esagonale regolare (figura 3.1c), in sostituzione di quello a maglia quadrata, potesse bastare a ristabilire l’isotropia perduta. Nel modello FHP (ancora dalle iniziali dei tre ricercatori), ogni cella è collegata alle sei celle vicine secondo angoli di 60 gradi (figura 3.3c). Potendo le particelle muoversi lungo tre direzioni invece che due, il numero di possibili stati della cella passa da  $2^4$  a  $2^6$ , pur rimanendo le leggi di transizione sostanzialmente equivalenti a quelle del modello HPP, con la differenza però che nel modello FHP sono possibili anche urti

che coinvolgono contemporaneamente tre particelle.

Sia  $n_i(\vec{r}, t)$  il numero di particelle (0 o 1) entranti nella cella, individuata dal vettore  $\vec{r}$  al passo  $t$  lungo la direzione  $\vec{c}_i$  ( $i = 1, 2, \dots, 6$ ) (figura 3.12). Sia, inoltre,  $\tau$  il tempo corrispondente a un passo di calcolo del modello FHP e  $\lambda$  la distanza tra due celle vicine del reticolo esagonale. Le velocità  $\vec{v}_i$  delle particelle lungo le direzioni  $\vec{c}_i$  sono definite nel seguente modo:

$$\vec{v}_i = \frac{\lambda}{\tau} \vec{c}_i$$

La funzione di transizione della cella può essere dedotta dalle regole di collisione, illustrate in figura 3.12, considerando il caso in cui non si verificano urti e i due casi in cui, invece, hanno luogo collisioni tra due o tre particelle:

**Nessuna collisione.** Se non si verifica alcuna collisione le particelle proseguono il loro cammino lungo l'originaria direzione reticolare, senza alcuna deviazione. Pertanto vale la seguente regola d'aggiornamento:

$$n_i(\vec{r} + \lambda \vec{c}_i, t + \tau) = n_i(\vec{r}, t)$$

cioè, ogni particella che al tempo  $t$  si trovi nella cella individuata dal vettore  $\vec{r}$ , al passo successivo,  $t + \tau$ , sarà nella cella adiacente, lungo la stessa direzione, individuata dal vettore  $\vec{r} + \lambda \vec{c}_i$ .

**Collisione tra due particelle.** Se solo  $n_i$  ed  $n_{i+3}$  valgono 1 nella cella  $\vec{r}$ , come illustrato in figura 3.12a-b, si verifica una collisione. La condizione

$$D_i = n_i n_{i+3} (1 - n_{i+1}) (1 - n_{i+2}) (1 - n_{i+4}) (1 - n_{i+5}) = 1$$

identifica tale urto. Pertanto il numero di particelle che permangono lungo la direzione  $\vec{c}_i$  dopo l'urto è dato da:

$$n_i - D_i$$

Tuttavia, una nuova particella può comparire nella direzione  $\vec{c}_i$  per effetto delle collisioni tra  $n_{i-1}$  ed  $n_{i+2}$  oppure tra  $n_{i+1}$  ed  $n_{i+4}$ . Il numero di particelle create nella direzione  $\vec{c}_i$  può essere, pertanto, dedotto dalla seguente formula:

$$q D_{i-1} + (1 - q) D_{i+1}$$

essendo  $q = q(\vec{r}, t)$  una variabile booleana random che vale 0 se dopo l'urto le particelle deviano verso sinistra, 1 se deviano verso destra. Pertanto, il numero complessivo di particelle lungo  $\vec{c}_i$  è dato da:

$$n_i - D_i + q D_{i-1} + (1 - q) D_{i+1}$$

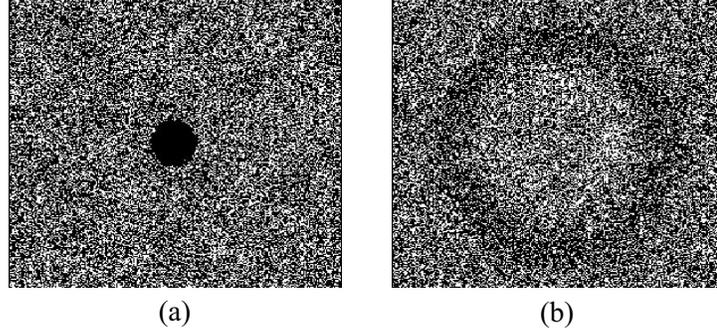


Figura 3.13: Dinamica di un'onda (b) nel modello FHP dovuta alla diffusione di una concentrazione di particelle nella zona centrale del reticolo (a).

**Collisione tra tre particelle.** La condizione

$$T_i = n_i n_{i+2} n_{i+4} (1 - n_{i+1}) (1 - n_{i+3}) (1 - n_{i+5}) = 1$$

identifica un urto tra tre particelle (figura 3.12c-d). L'effetto è quello di deviare le particelle di un angolo di 180 gradi. Pertanto il numero di particelle che rimangono lungo la direzione  $\vec{c}_i$  dopo l'urto è dato da:

$$n_i - T_i$$

Tuttavia, un urto tra le particelle  $n_{i+1}, n_{i+3}$  ed  $n_{i+5}$  determina la comparsa di una nuova particella lungo la direzione  $\vec{c}_i$ , condizione che si esprime nel seguente modo:

$$T_{i+3}$$

Il numero complessivo di particelle nella direzione  $\vec{c}_i$  per effetto di collisioni tra tre particelle è dato, dunque, da:

$$n_i - T_i + T_{i+3}$$

Pertanto la dinamica del modello FHP lungo ogni direzione del reticolo è definita dalla seguente legge:

$$n_i(\vec{r} + \lambda \vec{c}_i, t + \tau) = n_i(\vec{r}, t) + \Omega_i(\vec{r}, t)$$

in cui il termine  $\Omega_i(\vec{r}, t) = -D_i + qD_{i-1} + (1 - q)D_{i+1} - T_i + T_{i+3}$  rappresenta gli effetti delle collisioni tra le particelle;  $\Omega_i$  è detto, infatti, termine di collisione.

Con queste poche e, tutto sommato, semplici modifiche (l'introduzione del reticolo esagonale e le nuove regole di collisione), l'FHP ha risolto il problema degli invarianti spuri e si è dimostrato capace di riprodurre correttamente fenomeni fluidodinamici complessi. Inoltre, Chopard e Droz [30] hanno dimostrato che l'introduzione di particelle con velocità nulla nelle versioni successive del modello FHP [62] consente di derivare dalle regole di collisione le equazioni di Navier-Stokes per fluidi non comprimibili. La figura 3.13 illustra un'applicazione del modello FHP.

L'isotropia, cioè l'equivalenza di tutte le direzioni del reticolo, gioca dunque un ruolo estremamente importante nella modellizzazione del comportamento delle particelle del fluido. Questo aspetto costituì un problema apparentemente insormontabile nel passaggio alla terza dimensione, che maggiormente interessa sul piano delle applicazioni pratiche. Infatti non esiste alcun solido elementare in grado di riempire in maniera congruente (cioè senza buchi) lo spazio, garantendo al tempo stesso l'isotropia [171]. Nel 1987 Frish, d'Humières e Lallemand, notarono però che già nello spazio a quattro dimensioni un solido con le proprietà richieste esiste. Si tratta di un ipercubo (la generalizzazione di un cubo) a facce centrate, su cui si è basata la costruzione del reticolo FCHC (Face Centered Hyper Cube) [69]. Il passaggio alla terza dimensione, benchè ineccepibile dal punto di vista teorico, pose tuttavia pesanti problemi in termini computazionali. Nell'FCHC si hanno 12 possibili direzioni e, di conseguenza, il numero di possibili stati diventa  $2^{24}$  (oltre 16 milioni). Così, la tabella di transizione che definisce l'esito di tutte le possibili configurazioni della cella veniva a essere di ben 48 Mbyte. Pur non considerando il fatto che nel 1987 una tale quantità era considerata decisamente alta, il problema più pesante era che una tabella di transizione così grande richiedeva tempi d'accesso inaccettabili e, per questo, molti studi si occuparono di ridurre le dimensioni.

### Modelli di Boltzmann su reticolo

Contemporaneamente alla ricerca di tavole di collisione ridotte, si è sviluppata una soluzione alternativa, oggi nota come metodo di Boltzmann su reticolo, proposta per la prima volta da McNamara e Zanetti [126] dell'Università di Chicago e, contemporaneamente, da Higuera e Jimenez del Politecnico di Madrid [84].

Un vantaggio non trascurabile dei modelli di Boltzmann su reticolo, rispetto ai Gas Reticolari, è la più elevata efficienza computazionale, visto che le quantità d'interesse non sono più le singole particelle ma la loro densità. Così, il numero di componenti del sistema si riduce sensibilmente e non è più necessaria la fase del calcolo di medie spazio-temporali per la determinazione delle quantità fisiche

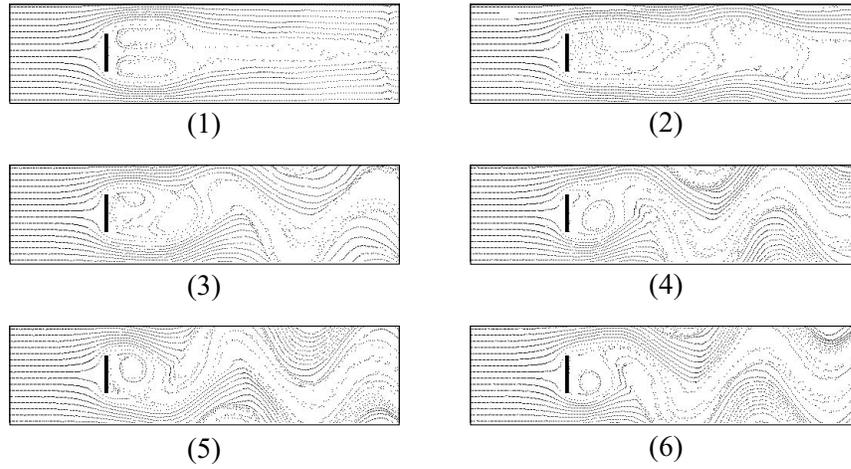


Figura 3.14: Simulazione di un flusso intorno a una lamina in un modello di Boltzmann su reticolo di 512x128 celle e  $\xi = 1$ . Le figure da 1 a 6 illustrano l'evoluzione del sistema.

d'interesse del sistema: la densità del fluido,  $\rho$ , la quantità di moto,  $\rho \vec{u}$ , sono note in ogni cella del reticolo.

Nella sua forma più semplice [155, 29], la dinamica di un modello di Boltzmann su reticolo può essere descritta nel seguente modo:

$$f_i(\vec{r} + \tau \vec{v}_i, t + \tau) - f_i(\vec{r}, t) = \Omega_i(f_i(\vec{r}, t)) = \frac{1}{\xi} \left( f_i^{(eq)}(\vec{r}, t) - f_i(\vec{r}, t) \right)$$

dove  $f_i(\vec{r}, t)$  rappresenta la densità di particelle che al tempo  $t$  si trovano nella cella  $\vec{r}$  con velocità  $\vec{v}_i$ ;  $f_i^{(eq)}(\vec{r}, t)$  rappresenta la, così detta, distribuzione d'equilibrio locale;  $\xi$ , detto termine di rilassamento, rappresenta il numero di passi di calcolo necessari affinché si raggiunga l'equilibrio nel contesto locale del vicinato.

Le leggi che determinano la dinamica nei modelli di Boltzmann su reticolo riducono, dunque, le condizioni di non equilibrio nel contesto locale del vicinato. La funzione  $f_i^{(eq)}(\vec{r}, t)$  specifica le condizioni d'equilibrio locale del sistema in funzione della densità,  $\rho = \sum f_i$ , e della quantità di moto,  $\rho \vec{u} = \sum f_i \vec{v}_i$ , del fluido nella cella. Il parametro  $\xi$  esprime, invece, la dipendenza del sistema dalla viscosità,  $\nu$ , del fluido. Vale, infatti la seguente relazione:  $\nu = K(\xi - 1/2)$ , dove  $K$  è una costante che dipende dalla geometria del reticolo [31]. In tal modo la viscosità diviene un parametro esplicito del modello, a differenza di quanto accade nei Gas Reticolari in cui essa dipende implicitamente dalla specificazione delle leggi che regolano gli urti tra le particelle.

I modelli di Boltzmann su reticolo coinvolgono, dunque, quantità esprimibili non più come numeri interi (come nel caso delle singole particelle nei Gas Reticolari) ma come numeri reali. Se da una parte questo può rappresentare un problema dal punto di vista della stabilità numerica (intesa come propagazione dell'errore, tipica dei modelli basati sull'elaborazione di valori in virgola mobile), dall'altra offre una maggiore flessibilità nella calibratura del modello dato che è possibile esplicitare, come nel caso della viscosità, alcuni importanti parametri del sistema. Inoltre, come per il modello FHP, anche per i modelli di Boltzmann su reticolo è stata dimostrata l'equivalenza con le equazioni di Navier-Stokes per fluidi incomprimibili [30]. La figura 3.14 illustra la dinamica del modello BKG [155] nella simulazione di un flusso intorno a una lamina.

Studi teorici e applicazioni dei Gas Reticolari e del metodo di Boltzmann su reticolo sono descritti in Rothman e Zaleski [160], in Luo [118], in Chopard e Masselot [32], in Bernaschi et al. [21], in Succi [172] e in Wolfram [191].