

# Progettazione e realizzazione di un simulatore di sistemi biologici

G. Caravagna<sup>1</sup>

<sup>1</sup>Dipartimento di Informatica  
Università di Pisa

Tirocinio interno, a.a.2004/05

Tutori Accademici: Prof.Barbuti, Prof.Maggiolo, Dott.Milazzo

# Indice

- 1 **Introduzione**
  - Lo stato dell'arte
  - Qualche definizione
  - Obiettivi preposti
- 2 **Implementazione**
  - Il modello logico
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 **Esempi**
  - La simulazione biochimica
  - La simulazione evuzionistica
- 4 **Conclusioni**
  - Futuri lavori

# Indice

- 1 **Introduzione**
  - **Lo stato dell'arte**
  - Qualche definizione
  - Obiettivi preposti
- 2 **Implementazione**
  - Il modello logico
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 **Esempi**
  - La simulazione biochimica
  - La simulazione evuzionistica
- 4 **Conclusioni**
  - Futuri lavori

# I Modelli

- sistemi di equazioni differenziali
  - le reazioni tradotte in equazioni risolte rispetto al tempo
- stocastico
  - la durata delle reazioni è una variabile aleatoria esponenziale
- probabilistico
  - *GBS (Generic Biological Simulator)*
  - KE (Kinetic Engine)

# I Modelli

- sistemi di equazioni differenziali
  - le reazioni tradotte in equazioni risolte rispetto al tempo
- stocastico
  - la durata delle reazioni è una variabile aleatoria esponenziale
- probabilistico
  - *GBS (Generic Biological Simulator)*
  - KE (Kinetic Engine)

# Indice

- 1 **Introduzione**
  - Lo stato dell'arte
  - **Qualche definizione**
  - Obiettivi preposti
- 2 **Implementazione**
  - Il modello logico
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 **Esempi**
  - La simulazione biochimica
  - La simulazione evuzionistica
- 4 **Conclusioni**
  - Futuri lavori

# Multinsiemi

## Definition

Un *multinsieme* è

- un insieme in cui le ripetizioni contano

*"Una aggregazione di 3 A ed 1 B può essere descritta da un multinsieme"*

$$\{A, A, A, B\} = \{3A, B\}$$

- possono essere usati semplicemente per descrivere sistemi biologici (*soluzioni chimiche, ...*)

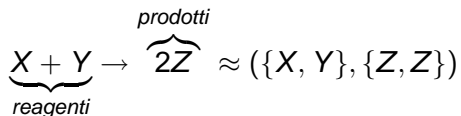
# Regole come funzioni

## Definition

Una *regola*  $R$  è

- una coppia  $R = (M, M')$  con  $M$  ed  $M'$  multinsiemi

" $X$  ed  $Y$  reagendo producono  $2Z$ "



- Cosa genera l'applicazione di una regola ad un multinsieme?

# Applicazione di regole

- applichiamo la regola



al multinsieme  $\{A, A, A, B\}$

- 1 sottraiamo al multinsieme i reagenti

$$\{A, A, A, B\} \setminus \{A, B\} = \{A, A\}$$

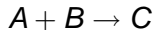
- 2 aggiungiamo adesso i prodotti al multinsieme ottenuto

$$\{A, A\} \cup \{C\} = \{A, A, C\}$$

- non più *applicabile* perchè  $\{A, B\} \not\subseteq \{A, A, C\}$

# Applicazione di regole

- applichiamo la regola



al multinsieme  $\{A, A, A, B\}$

- 1 sottraiamo al multinsieme i reagenti

$$\{A, A, A, B\} \setminus \{A, B\} = \{A, A\}$$

- 2 aggiungiamo adesso i prodotti al multinsieme ottenuto

$$\{A, A\} \cup \{C\} = \{A, A, C\}$$

- non più *applicabile* perchè  $\{A, B\} \not\subseteq \{A, A, C\}$

# Applicazione di regole

- applichiamo la regola



al multinsieme  $\{A, A, A, B\}$

- 1 sottraiamo al multinsieme i reagenti

$$\{A, A, A, B\} \setminus \{A, B\} = \{A, A\}$$

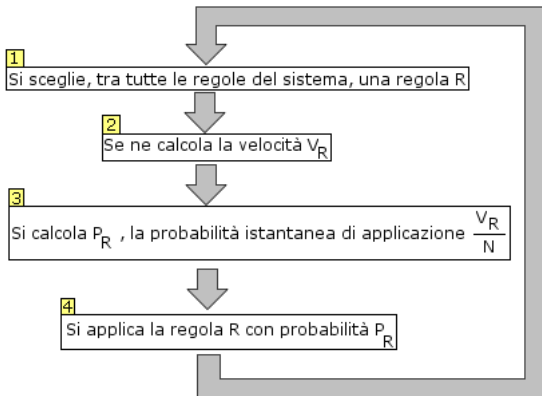
- 2 aggiungiamo adesso i prodotti al multinsieme ottenuto

$$\{A, A\} \cup \{C\} = \{A, A, C\}$$

- non più *applicabile* perchè  $\{A, B\} \not\subseteq \{A, A, C\}$

# L'algoritmo di simulazione

Dato un multinsieme ed un insieme di regole



# Indice

- 1 **Introduzione**
  - Lo stato dell'arte
  - Qualche definizione
  - **Obiettivi preposti**
- 2 **Implementazione**
  - Il modello logico
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 **Esempi**
  - La simulazione biochimica
  - La simulazione evuzionistica
- 4 **Conclusioni**
  - Futuri lavori

# Caratteristiche Principali

- facilmente applicabile a diversi sistemi biologici
  - "sistemi biologici con caratteristiche comuni sono simulabili dal GBS, qualsiasi sia la loro *semantica*", **OBIETTIVO PRIMARIO**
- distribuito
- motore *parallelo*
  - threads e sincronizzazione
- protocollo di comunicazione ben definito
  - BSP (*Biological Simulation Protocol*)

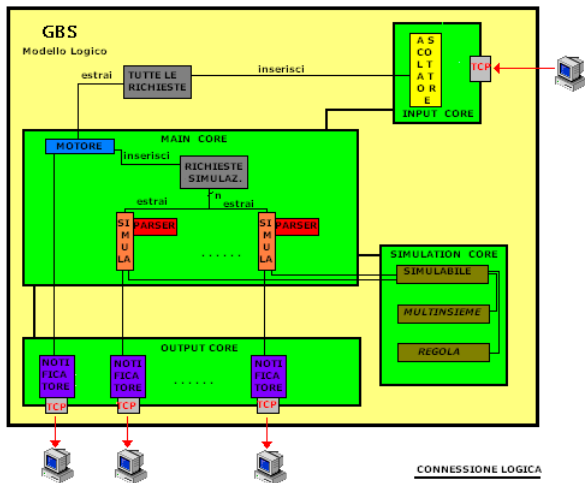
# Caratteristiche Principali

- facilmente applicabile a diversi sistemi biologici
  - "sistemi biologici con caratteristiche comuni sono simulabili dal GBS, qualsiasi sia la loro *semantica*", **OBIETTIVO PRIMARIO**
- distribuito
- motore *parallelo*
  - threads e sincronizzazione
- protocollo di comunicazione ben definito
  - BSP (*Biological Simulation Protocol*)

# Indice

- 1 Introduzione
  - Lo stato dell'arte
  - Qualche definizione
  - Obiettivi preposti
- 2 **Implementazione**
  - **Il modello logico**
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 Esempi
  - La simulazione biochimica
  - La simulazione evuzionistica
- 4 Conclusioni
  - Futuri lavori

# Il modello logico



# Indice

- 1 Introduzione
  - Lo stato dell'arte
  - Qualche definizione
  - Obiettivi preposti
- 2 **Implementazione**
  - Il modello logico
  - **L'utilizzo di Java**
  - Ottenere l'estendibilità
- 3 Esempi
  - La simulazione biochimica
  - La simulazione evuzionistica
- 4 Conclusioni
  - Futuri lavori

# Il linguaggio Java

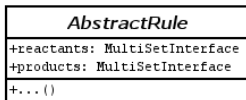
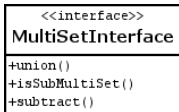
- Favorisce il raggiungimento degli obiettivi preposti
  - facile specializzazione del GBS
    - interfacce
    - classi astratte
    - programmazione *"più facile"*
  - portabilità

# Indice

- 1 Introduzione
  - Lo stato dell'arte
  - Qualche definizione
  - Obiettivi preposti
- 2 **Implementazione**
  - Il modello logico
  - L'utilizzo di Java
  - **Ottenere l'estendibilità**
- 3 Esempi
  - La simulazione biochimica
  - La simulazione evolucionistica
- 4 Conclusioni
  - Futuri lavori

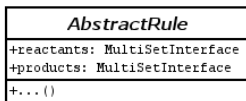
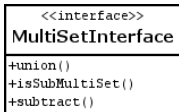
# specializzazione come caratteristica principale

- interfacce
  - *MultiSetInterface*: multinsiemi con le operazioni definite
- classi astratte
  - *AbstractRule*: regole come coppie di *MultiSetInterface*
- *class loading dinamico*



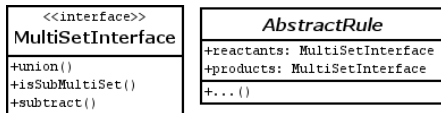
# specializzazione come caratteristica principale

- interfacce
  - *MultiSetInterface*: multinsiemi con le operazioni definite
- classi astratte
  - *AbstractRule*: regole come coppie di *MultiSetInterface*
- *class loading dinamico*



## specializzazione come caratteristica principale

- interfacce
  - *MultiSetInterface*: multinsiemi con le operazioni definite
- classi astratte
  - *AbstractRule*: regole come coppie di *MultiSetInterface*
- *class loading dinamico*



# Indice

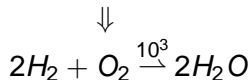
- 1 Introduzione
  - Lo stato dell'arte
  - Qualche definizione
  - Obiettivi preposti
- 2 Implementazione
  - Il modello logico
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 **Esempi**
  - **La simulazione biochimica**
  - La simulazione evuzionistica
- 4 Conclusioni
  - Futuri lavori

# La simulazione biochimica: descrizione

L'evoluzione di una soluzione di molecole nel tempo

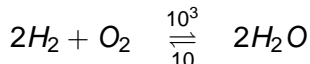
- velocità delle reazioni
  - costante cinetica  $k$
- reazioni descritte da regole

"2 molecole di  $H_2$  ed 1 molecola di  $O_2$  producono 2 molecole di  $H_2O$  con velocità  $10^3$ "



## La simulazione biochimica: l'acqua

- una soluzione contenente  $\{10H_2, 10O_2, 100H_2O\}$  e le regole

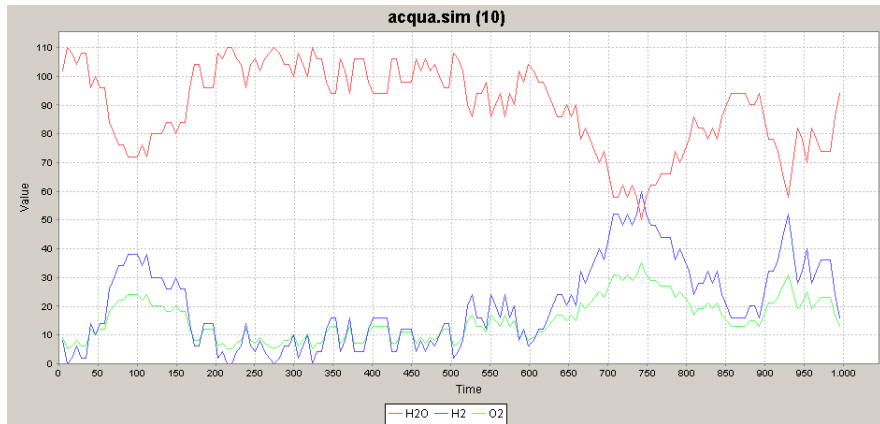


- le probabilità istantanee si calcolano come

$$\vec{P} = \frac{[H_2]^2 [O_2]}{N} \vec{k} = \frac{10^2 10}{N} 10^3 = \frac{10^6}{N}$$

$$\overleftarrow{P} = \frac{[H_2O]^2}{N} \overleftarrow{k} = \frac{100^2}{N} 10 = \frac{100^2}{N} 10 = \frac{10^5}{N}$$

# La simulazione biochimica: un grafico



# Indice

- 1 Introduzione
  - Lo stato dell'arte
  - Qualche definizione
  - Obiettivi preposti
- 2 Implementazione
  - Il modello logico
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 **Esempi**
  - La simulazione biochimica
  - **La simulazione evuzionistica**
- 4 Conclusioni
  - Futuri lavori

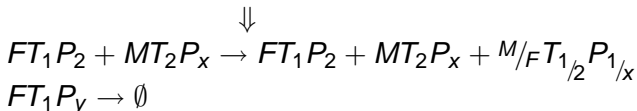
# La simulazione evolutivistica: descrizione

L'evoluzione di una *specie animale* nel tempo

- caratteristiche genetiche ( $T_i$ ) e preferenze per l'accoppiamento ( $P_j$ )
- per esprimere la velocità non basta una costante cinetica
- regole *parametriche* di *nascita e morte*

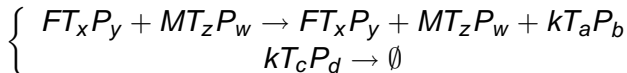
"Una femmina di tipo 1 si accoppia con un maschio di tipo 2 e nascerà un figlio/a di tipo 1/2"

"Una femmina di tipo 1 muore"



## La simulazione evolutivistica

- un multinsieme di individui  $\{kT_iP_j\}$  con  $k \in \{M, F\}$  ed  $i, j \in \{0, 1, 2\}$
- le regole *parametriche*

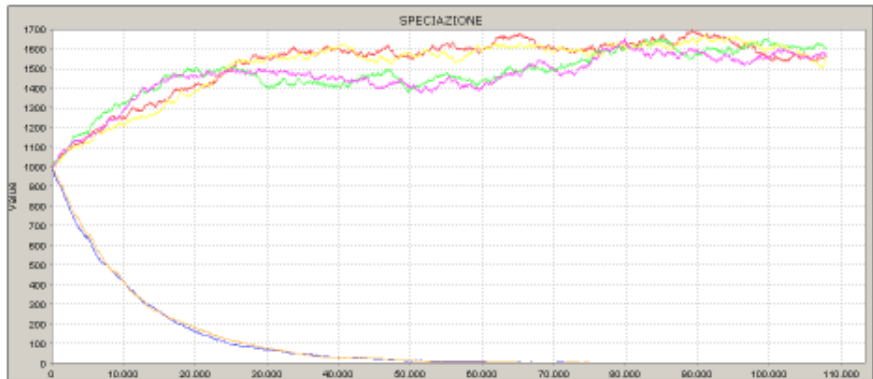


- una volta istanziata ( $MT_1P_2 \rightarrow \emptyset$ ) vale comunque

$$p_{morte} = \frac{V_{morte}}{N}$$

## Grafici

- Utilizzando parametri opportuni alcune specie prevalgono su altre



# Indice

- 1 Introduzione
  - Lo stato dell'arte
  - Qualche definizione
  - Obiettivi preposti
- 2 Implementazione
  - Il modello logico
  - L'utilizzo di Java
  - Ottenere l'estendibilità
- 3 Esempi
  - La simulazione biochimica
  - La simulazione evuzionistica
- 4 **Conclusioni**
  - **Futuri lavori**

# Futuri lavori

- linguaggi standard per la descrizione
  - SBML (*System Biological Markup Language*)
- estensione del modello
  - SBML è molto più espressivo

