

# Intelligenza Artificiale

## Sistemi a regole

Marco Piastra

# 1

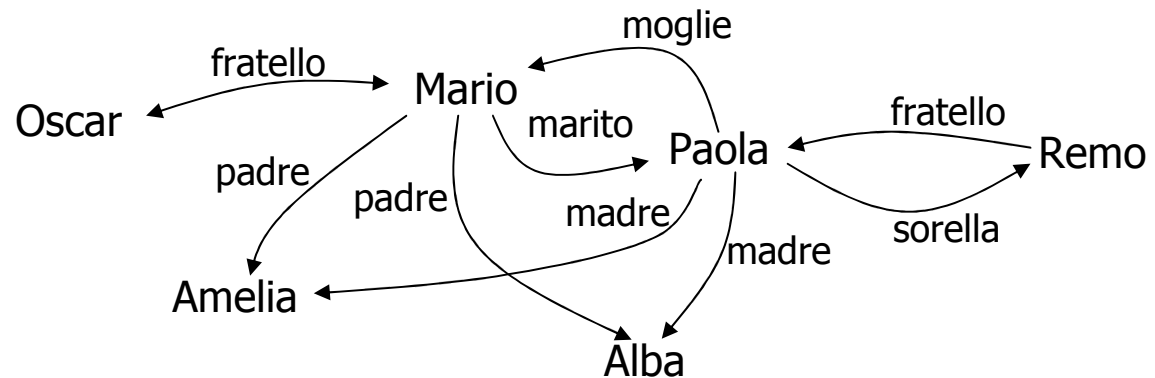
## Sistemi a regole (*Production Systems*)

# Logica e sistemi a regole

- La logica simbolica:
  - è un sistema per la rappresentazione formale del ragionamento
  - si basa su un formalismo di rappresentazione e su regole di derivazione sintattica (regole di inferenza)
- I sistemi a regole
  - adottano una forma semplificata di programmazione logica
  - sono stati concepiti per una classe di applicazioni particolari (*sistemi esperti* o *expert systems*)

# Logica – Fatti e regole

- Si distinguono:
  - *Principi generali*
  - *Fatti specifici*



- Esempio:

*Regole :*

$$\forall x \forall y ((\text{madre}(z,x) \wedge \text{madre}(z,y) \wedge \text{padre}(z,x) \wedge \text{padre}(z,y) \wedge \text{maschio}(x)) \rightarrow \text{fratello}(x, y))$$

$$\forall x \forall y ((\text{madre}(z,x) \wedge \text{madre}(z,y) \wedge \text{padre}(z,x) \wedge \text{padre}(z,y) \wedge \text{femmina}(x)) \rightarrow \text{sorella}(x, y))$$

*Fatti specifici :*

femmina(Amelia), femmina(Alba), femmina(Paola), maschio(Mario)

madre(Paola,Amelia), madre(Paola,Alba)

padre(Mario,Amelia), padre(Mario,Alba)

# Regole di produzione

- Un sistema a regole contiene un insieme di regole di produzione
- Ciascuna regola ha la forma
  - <condizioni>  $\Rightarrow$  <azioni>
  - talvolta anche descritte come  
<LHS - Left Hand Side>  $\Rightarrow$  <RHS - Right Hand Side>
  - condizioni e azioni sono in forma normale congiuntiva (FNC)

- Esempio:

Regola "Fratello"

madre(m,x)

madre(m,y)

padre(p,x)

padre(p,y)

maschio(x)

$\Rightarrow$

fratello(x, y)

Regola "Sorella"

madre(m,x)

madre(m,y)

padre(p,x)

padre(p,y)

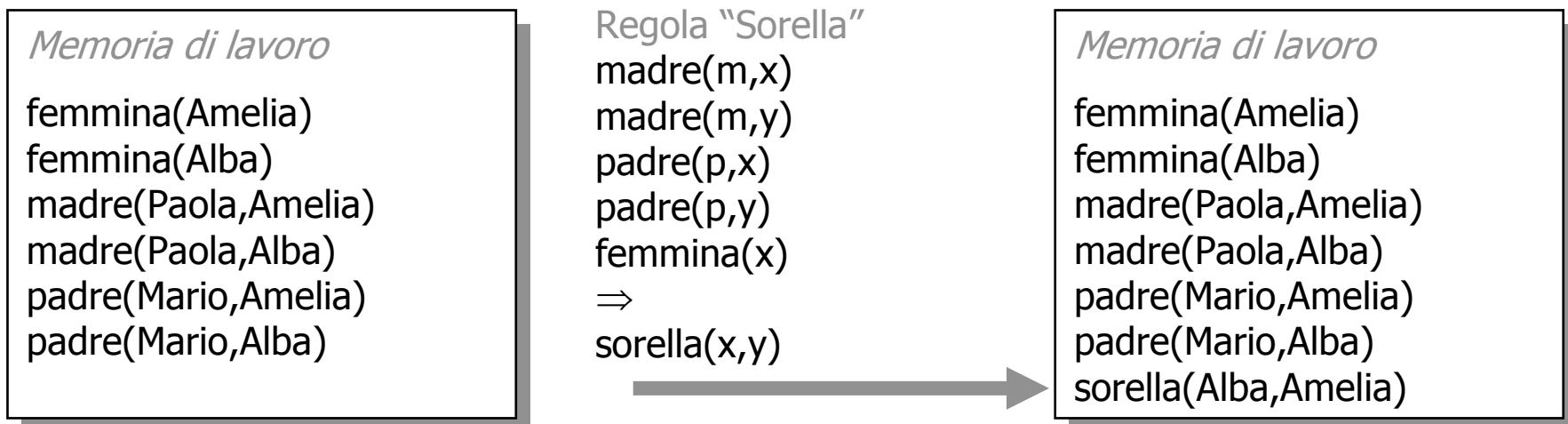
femmina(x)

$\Rightarrow$

sorella(x, y)

# Memoria di lavoro

- Un sistema a regole include anche una *memoria di lavoro* (anche *working memory* o WM)
  - la *memoria di lavoro* contiene la rappresentazione dei *fatti specifici*
  - le *regole* operano sulla *memoria di lavoro*
  - le *condizioni* sono istanziate sulla base dei *fatti specifici*
  - le *azioni* tipicamente comportano l'asserzione o la ritrattazione di *fatti specifici* (ma non solo)



# Agenda, attivazione

- In ogni istante, un sistema a regole mantiene un *agenda* che contiene le *regole istanziate*
- Il sistema sceglie le *regole istanziate* da attivare
- L'*attivazione* (*firing*) delle regole avviene in modo sequenziale

## Memoria di lavoro

femmina(Amelia)  
 femmina(Alba)  
 madre(Paola,Amelia)  
 madre(Paola,Alba)  
 padre(Mario,Amelia)  
 padre(Mario,Alba)

## Agenda

Regola "Sorella"  
 (istanza 1)

madre(Paola,Amelia)  
 madre(Paola,Alba)  
 padre(Mario,Amelia)  
 padre(Mario,Alba)  
 femmina(Amelia)

⇒

sorella(Amelia, Alba)

Regola "Sorella"  
 (istanza 2)

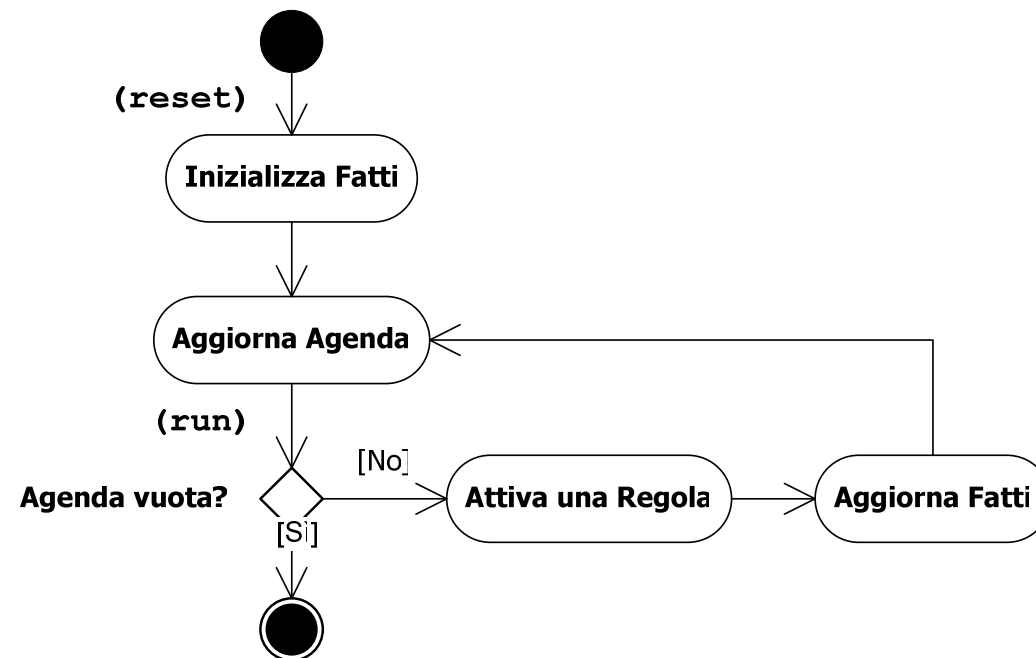
madre(Paola,Alba)  
 madre(Paola,Amelia)  
 padre(Mario,Alba)  
 padre(Mario,Alba)  
 femmina(Alba)

⇒

sorella(Alba,Amelia)

# Ciclo di esecuzione

- Il sistema a regole procede ciclicamente:
  - aggiorna l'agenda (regole istanziate)
  - sceglie ed attiva una regola  
(la scelta è casuale, rispettando le classi di precedenza)
  - aggiorna la memoria di lavoro (fatti)



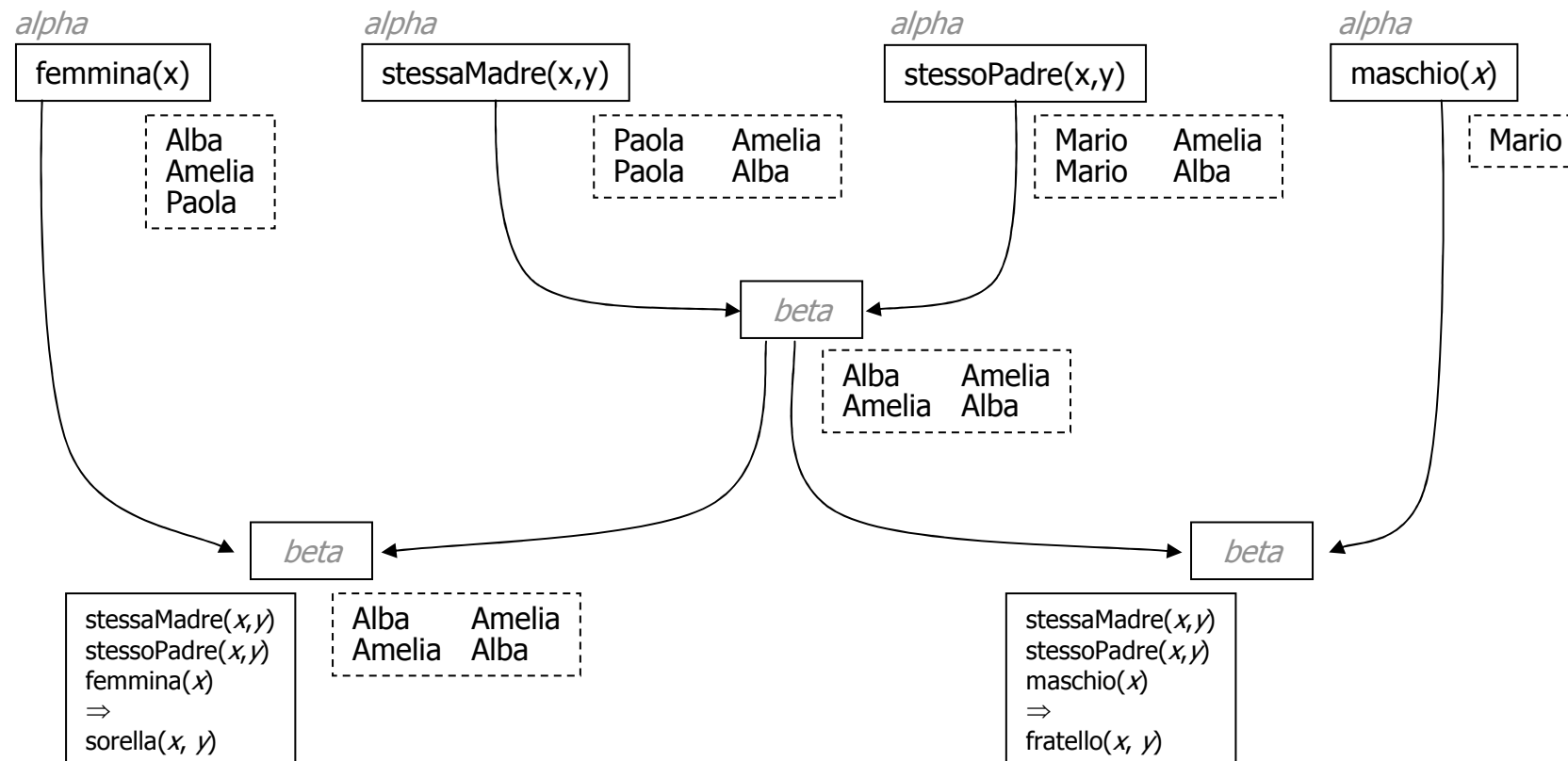


# Sistema a regole - come funzionano?

- Il punto critico è **l'aggiornamento dell'agenda**
  - occorre identificare tutte le *istanziazioni* delle regole
  - evitando i cicli infiniti
    - le regole vanno inserite nell'agenda solo in presenza di *fatti nuovi*
    - diversamente, la loro attivazione è inutile
    - si veda l'esempio precedente
- Il confronto diretto è dispendioso
  - sarebbe necessario confrontare tutte le *regole* con tutti i *fatti*
    - distinguendo i fatti nuovi da quelli già noti

# Algoritmo Rete (C. Forgy, 1980)

- Le *condizioni* di un *insieme di regole* vengono rappresentate in forma di *grafo aciclico*
  - a cui viene 'agganciata' la rappresentazione della *memoria di lavoro*



# Aggiornamento della Rete

- I nuovi fatti vengono 'agganciati' ai nodi di pertinenza
  - "Gino ha gli stessi genitori di Paola"

