Intelligenza Artificiale I

Esercitazione 3

Marco Piastra

Notazione prefissa

(file dilemma.jess)

- In Jess, come in Lisp, le espressioni sono in forma prefissa

Esempio:

```
(defrule move-alone
   ?node <- (status (search-depth ?num)
                    (farmer-location ?fs))
  (opposite-of ?fs ?ns)
  =>
  (duplicate ?node (search-depth (+ 1 ?num))
                   (parent ?node)
                   (farmer-location ?ns)
                   (last-move alone))
)
```

Vincoli sulle variabili (uguaglianza, disuguaglianza)

(file dilemma.jess)

- Vengono espressi in forma abbreviata

Esempio:

```
(defrule fox-eats-goat
  (declare (salience 100))
    ?node <- (status (farmer-location ?s1)
                     (fox-location ?s2&~?s1)
                     (goat-location ?s2))
     =>
    (retract ?node)
)
```

Variabile con un vincolo

Vincoli sulle variabili (uguaglianza, disuguaglianza)

(file dilemma.jess)

- Vengono espressi in forma abbreviata

Esempio:

```
(defrule fox-eats-goat
  (declare (salience 100))
    ?node <- (status (farmer-location ?s1)
                     (fox-location ?s2&~?s1)
                     (goat-location ?s2))
     =>
    (retract ?node)
)
```

Variabile

Vincoli sulle variabili (uguaglianza, disuguaglianza)

(file dilemma.jess)

- Vengono espressi in forma abbreviata

Esempio:

```
(defrule fox-eats-goat
  (declare (salience 100))
    ?node <- (status (farmer-location ?s1)
                     (fox-location ?s2&~?s1)
                     (goat-location ?s2))
     =>
    (retract ?node)
)
```

Vincolo

Vincoli sulle variabili (uguaglianza, disuguaglianza)

(file dilemma.jess)

- Vengono espressi in forma abbreviata

Esempio:

```
(defrule fox-eats-goat
  (declare (salience 100))
    ?node <- (status (farmer-location ?s1)
                     (fox-location ?s2&~?s1)
                     (goat-location ?s2))
     =>
    (retract ?node)
)
```

Sintassi generale: `&`<*variabile*>
(in forma positiva)

Vincoli sulle variabili (uguaglianza, disuguaglianza)

(file dilemma.jess)

- Vengono espressi in forma abbreviata

Esempio:

```
(defrule fox-eats-goat
  (declare (salience 100))
    ?node <- (status (farmer-location ?s1)
                     (fox-location ?s2&~?s1)
                     (goat-location ?s2))
     =>
    (retract ?node)
)
```

Sintassi generale: **&~**<*variabile*>
(in forma negativa)

Vincoli sulle variabili (uguaglianza, disuguaglianza)

(file dilemma.jess)

- Vengono espressi in forma abbreviata

Esempio:

```
(defrule fox-eats-goat
  (declare (salience 100))
    ?node <- (status (farmer-location ?s1)
                     (fox-location ?s2&~?s1)
                     (goat-location ?s2))
     =>
    (retract ?node)
)
```

In questo caso il vincolo è negativo

Significato:
la variabile `?s2` si
lega ad un valore
che *non* sia uguale a `?s2`

Vincoli sulle variabili (uguaglianza, disuguaglianza)

(file dilemma.jess)

- Vengono espressi in forma abbreviata

Esempio:

```
(defrule fox-eats-goat
  (declare (salience 100))
    ?node <- (status (farmer-location ?s1)
                     (fox-location ?s2&~?s1)
                     (goat-location ?s2))
     =>
    (retract ?node)
)
```

In questo caso il vincolo è negativo

Significato:
la variabile `?s2` si
lega ad un valore
che *non* sia uguale a `?s1`

(per un riferimento completo
alla sintassi delle LHS di Jess
vedere la pagina del corso)

duplicate

(file dilemma.jess)

- Crea una copia e modifica (la copia, non l'originale)

Esempio:

```
(defrule move-alone
    ?node <- (status (search-depth ?num)
                     (farmer-location ?fs))
    (opposite-of ?fs ?ns)
    =>
    (duplicate ?node (search-depth (+ 1 ?num))
                     (parent ?node)
                     (farmer-location ?ns)
                     (last-move alone))
)
```

Fox, Goat & Cabbage

(file dilemma.jess)

- Come salvare capra e cavoli
  a) Come funziona il programma?
  b) Quale struttura viene costruita in memoria e cosa rappresenta?
  c) Cosa succederebbe se eliminassimo le priorità? (provare per credere)
  d) Come si impediscono i cicli infiniti?

Advanced Fox, Goat & Cabbage

(file dilemma.jess)

- Come andare oltre

e) Quale strategia viene adottata per costruire la struttura di memoria?

Modificare il file dilemma.jess ed inserire all'inizio: **`(set-strategy breadth)`**

Si può inserire anche come comando, al prompt

Cosa cambia nell'ordine di esecuzione delle regole?

Suggerimento:

1) Aprire due finestre, con due Jess
2) Inserire **`(set-strategy breadth)`** in una sola delle due
3) Eseguire un passo alla volta, usando **`(run 1)`** e **`(agenda)`**, in parallelo

(Per tornare alla strategia originale: **`(set-strategy depth)`**)

E' la strategia di default

Agenda (Come funziona Jess - terza approssimazione)

- Completare il box della CONFLICT RESOLUTION nei due casi