

Intelligenza Artificiale I

Esercitazione 3

Marco Piastra

Semantic Tableau, regole alfa e beta

- Un tableau è un'insieme di fbf

Le regole (alfa e beta) trasformano un tableau in uno o due nuovi tableau

Si parte da un tableau iniziale

- Regole alfa (o di espansione)

(a1)	(a2)	(a3)	(a4)
$\neg(\neg\varphi)$	$\varphi \wedge \psi$	$\neg(\varphi \vee \psi)$	$\neg(\varphi \rightarrow \psi)$
φ	φ, ψ	$\neg\varphi, \neg\psi$	$\varphi, \neg\psi$

- Regole beta (o di biforcazione)

(b1)	(b2)	(b3)	(b4)	(b5)
$\varphi \vee \psi$	$\neg(\varphi \wedge \psi)$	$\varphi \rightarrow \psi$	$\varphi \leftrightarrow \psi$	$\neg(\varphi \leftrightarrow \psi)$
/ \	/ \	/ \	/ \	/ \
$\varphi \quad \psi$	$\neg\varphi \quad \neg\psi$	$\neg\varphi \quad \psi$	$\neg\varphi, \neg\psi \quad \varphi, \psi$	$\neg\varphi, \psi \quad \varphi, \neg\psi$

Metodo dei *Semantic Tableau*

- Procedura:

Dato il problema $\Gamma \models \varphi$

Si assume $\Gamma \cup \{\neg\varphi\}$ come nodo (*tableau*) iniziale

Su ciascun ramo, in modalità *depth-first*

Se il nodo contiene solo letterali,

se il nodo contiene una contraddizione, chiudere il ramo
altrimenti terminare la procedura [fallimento]

Se il nodo contiene formule composite:

- a) applicare le regole alfa
- b) applicare le regole beta

Notare: le regole alfa vengono applicate prima delle regole beta

deffunction

(file semanticTableau.jess)

- Definisce una funzione

Esempio:

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
      then
      (return ?n)
      else
      (return
        (duplicate ?n
          (id (gensym*)) ;; Always make sure it has a unique id
          (op1 (duplicate-deep (fact-slot-value ?n op1)))
          (op2 (duplicate-deep (fact-slot-value ?n op2)))
        )
      )
  )
)
```

deffunction

(file semanticTableau.jess)

- Definisce una funzione

nome

Esempio:

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
    then
      (return ?n)
    else
      (return
        (duplicate ?n
          (id (gensym*)) ;; Always make sure it has a unique id
          (op1 (duplicate-deep (fact-slot-value ?n op1)))
          (op2 (duplicate-deep (fact-slot-value ?n op2)))
        )
      )
    )
  )
)
```

deffunction

(file semanticTableau.jess)

- Definisce una funzione

Esempio:

parametri

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
      then
      (return ?n)
      else
      (return
        (duplicate ?n
          (id (gensym*)) ;; Always make sure it has a unique id
          (op1 (duplicate-deep (fact-slot-value ?n op1)))
          (op2 (duplicate-deep (fact-slot-value ?n op2)))
        )
      )
  )
)
```

deffunction

(file semanticTableau.jess)

- Definisce una funzione

Esempio:

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
      then
      (return ?n)
      else
      (return
        (duplicate ?n
          (id (gensym*)) ;; Always make sure it has a unique id
          (op1 (duplicate-deep (fact-slot-value ?n op1)))
          (op2 (duplicate-deep (fact-slot-value ?n op2)))
        )
      )
  )
)
```

controllo di flusso

deffunction

(file semanticTableau.jess)

- Definisce una funzione

Esempio:

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
      then
      (return ?n)      ritorno di valori
      else
      (return
        (duplicate ?n
          (id (gensym*)) ;; Always make sure it has a unique id
          (op1 (duplicate-deep (fact-slot-value ?n op1)))
          (op2 (duplicate-deep (fact-slot-value ?n op2)))
        )
      )
  )
)
```


deffunction

(file semanticTableau.jess)

- Definisce una funzione

Esempio:

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
    then
      (return ?n)
    else
      (return
        (duplicate ?n
          (id (gensym*)) ;; Always make sure it has a unique id
          (op1 (duplicate-deep (fact-slot-value ?n op1)))
          (op2 (duplicate-deep (fact-slot-value ?n op2)))
        )
      )
    )
  )
)
```

predicato speciale di Jess:
vale TRUE se l'argomento è un *simbolo*

deffunction

(file semanticTableau.jess)

- Definisce una funzione

Esempio:

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
    then
      (return ?n)
    else
      (return
        (duplicate ?n
          (id (gensym*)) ;; Always make sure it has a unique id
          (op1 (duplicate-deep (fact-slot-value ?n op1)))
          (op2 (duplicate-deep (fact-slot-value ?n op2)))
        )
      )
    )
  )
)
```

funzione speciale di Jess:
ritorna un (nuovo) *simbolo univoco*

Semantic Tableau

(file semanticTableau.jess)

- Domande:
 - a) Cosa vuol dire metodo per *refutazione*?
(non procedere oltre prima di aver capito)
 - b) Come vengono rappresentate le formule?
 - c) Come vengono rappresentati i tableau?
 - d) Come funziona l'implementazione Jess del metodo?
 - e) Perché si usa **duplicate-deep** e **retract-deep**?
 - f) A cosa servono le priorità?

(usare l'esempio st-TheProblem.jess)