

Artificial Intelligence

Lab 3

Marco Piastra

Semantic Tableau, alpha and beta rules

- A tableau a set of wff

There is an initial tableau

Alpha and beta rules transform a tableau in one or two new tableau

- Alpha rules (*expansion*)

(a1)

$$\begin{array}{c} \neg(\neg\varphi) \\ | \\ \varphi \end{array}$$

(a2)

$$\begin{array}{c} \varphi \wedge \psi \\ | \\ \varphi, \psi \end{array}$$

(a3)

$$\begin{array}{c} \neg(\varphi \vee \psi) \\ | \\ \neg\varphi, \neg\psi \end{array}$$

(a4)

$$\begin{array}{c} \neg(\varphi \rightarrow \psi) \\ | \\ \varphi, \neg\psi \end{array}$$

- Beta rules (*bifurcation*)

(b1)

$$\begin{array}{c} \varphi \vee \psi \\ / \quad \backslash \\ \varphi \quad \psi \end{array}$$

(b2)

$$\begin{array}{c} \neg(\varphi \wedge \psi) \\ / \quad \backslash \\ \neg\varphi \quad \neg\psi \end{array}$$

(b3)

$$\begin{array}{c} \varphi \rightarrow \psi \\ / \quad \backslash \\ \neg\varphi \quad \psi \end{array}$$

(b4)

$$\begin{array}{c} \varphi \leftrightarrow \psi \\ / \quad \backslash \\ \neg\varphi, \neg\psi \quad \varphi, \psi \end{array}$$

(b5)

$$\begin{array}{c} \neg(\varphi \leftrightarrow \psi) \\ / \quad \backslash \\ \neg\varphi, \psi \quad \varphi, \neg\psi \end{array}$$

deffunction

(file semanticTableau.jess)

▪ Functions

Example:

```
(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)

▪ Functions

name

parameters

Example:

```
(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)

▪ Functions

Example:

```
(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)))
        )
      )
  )
)
```

flow control

deffunction

(file semanticTableau.jess)

▪ Functions

Example:

```
(deffunction duplicate-deep (?n)
  (if (symbolp ?n)
      then
      (return ?n)           returning values
      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)

▪ Functions

Example:

```
(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)))
          )
        )
    )
)
```

a Jess predicate:

returns TRUE if the argument is an atomic *symbol*

deffunction

(file semanticTableau.jess)

▪ Functions

Example:

```
(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))))
         )
        )
  )
)
```

a Jess function:
returns a new and unique *symbol*

Semantic Tableau

(file semanticTableau.jess)

- Questions:
 - a) What does it mean that this is a *refutation method*?
(make sure you understand this before going further)
 - b) How are wffs represented?
 - c) How are the tableau represented?
 - d) How would you describe the algorithm implemented in Jess?
 - e) What is the meaning of **duplicate-deep** and **retract-deep**?
 - f) What is the role of salience?

(see the example st-TheProblem.jess)