Surprise Test

Ans 1: I have taken the following assumptions for this question.

→ S is made unambiguous by use of paranthesis

→ Only include symbols ∧, ∨, ~ for simplicity

→ Denote truth value as m[a]

model

fun PL-TRUE?(s,m):

① LHS="" RHS="" logic_op=""

② Scan s from left to right, for a ~, (, or symbol.

③ If symbol found, LHS = symbol, goto line 4

④ else if ~ found, logic_op = ~, goto line 2

else if left paranthesis, (is found:

set n=1

while(n>0): { LHS += next_char
if next_char =): n--;
else if (: n++;

④ If logic_op = "~", logic_op = next_ch.

⑤ Repeat ③, ④ & replace LHS with RHS

⑥ If LHS & RHS is not symbol,

LHS = PL_TRUE (LHS, m)

RHS = PL_TRUE(RHS, m)

⑦ If logic_op = ~ :

return true if LHS is false,

else return false

⑧ Else if logic_op = ∧

return true if LHS & RHS are true

else false

⑨ else if logic_op = ∨

return true if LHS or RHS are true

else false

Ans 2:

Examples of sentences:

(1) true

(2) A ∨ true

(3) false

Ans 3: In the partial model m, assume that k is the no. of symbols is 5 do not appear.

Despite the truth-table of the symbols in the sentence, some sentences will be always true or always false.

So, the only way to be sure is to evaluate all the rows of the truth table one-by-one, and check

which one is true & false, to determine the sentence's truth value.

But due to the existence of always true & always false sentences, in the worst case $O(2^k)$ operation to determine sentence's truth value

---

M. Mahesh Reddy
180ICS31

Last Page