Program1 este programul care face verificarea de sintaxa, construirea arborelui formulei si interpretarea. (Tema 1 si 2)

* formula propozitionala introdusa de la tastatura va fi in sintaxa stricta
* doar litere mari (ex. A, B, C - fara x, y, P1, P2 etc.)
* raspunsurile la intrebari vor fi cu da sau nu (litere mici)
* nu e obligatorie construirea unui arbore in cazul in care se vrea returnarea valorii de adevar sub interpretarea respectiva formulei
* daca programul spune ca sintaxa nu este buna, acesta nu se va opri, deci este indicata oprirea acestuia (nu am reusit sa gasesc o rezolvare)

Mai jos am scris cum functioneaza programul:

1) Verificare

Un program recursiv care parcurge, pe rand, fiecare element din formula si verifica cazul in care se afla si cazul in care se va afla.

caz -1: nu e formula prop

caz 0: e formula prop

caz 1: (

urmatorul poate fi 1, 2, 3

caz 2: A

urm 4, 5, 0

caz 3: ¬

urm 1, 2

caz 4: ∧, ∨, ⇒, ⇔

urm 1, 2

caz 5: )

urm 4, 5, 0

Eroare = nu este formula propozitionala

eroare 1: din cazul 1

eroare 2: din cazul 2

eroare 3: din cazul 3

eroare 4: din cazul 4

eroare 5: din cazul 5

eroare 6: par\_d != par\_i (numarul de paranteze deschise diferit de numarul de paranteze inchise)

eroare 7: par != con (numarul de perechi de paranteze diferit de numarul de conectori)

eroare 8: (A) nu e voie

eroare 9: (not A) urm elem ')'

eroare 10: ((...)) nu e voie

Program testat pe tema 1:

((P ∧ Q) ⇒ (¬R)) true

(((P ⇒ Q) ∨ S) ⇔ T) true

((P ⇒ (Q ∧ (S ⇒ T)))) false

(¬(B(¬Q)) ∧ R) false

(P ∧ ((¬Q) ∧ (¬(¬(Q ⇔ (¬R)))))) true

((P ∨ Q) ⇒ ¬(P ∨ Q)) ∧ (P ∨ (¬(¬Q))) false

2) Arbore

Un program care primeste o formula propozitionala si creeaza arborele formulei respective.

Ex.

input: ((P ∧ Q) ⇒ (¬R))

output: [[[P],∧,[Q]],⇒,[¬,[R]]]

3) Interpretare

Un program care primeste o formula propozitionala sau arborele ei si returneaza valoarea de adevar sub interpretarea respectiva a formulei sau a arborelui.

Impartit in 3 subprograme:

* inlocuire\_in\_propozitie - primeste un formula sau arborele formulei sub forma de lista si inlocuieste literele cu valorile de adevar introduse de utilizator de la tastatura
* adev\_sau\_fals - primeste 3 elemente (printe care un conector si minim o valoare de adevar) aflate succesiv in formula si returneaza valoarea de adevar rezultata (ex. A ∧ F = F, ¬F = A)
* interpretare - subprogramul principal in care are loc interpretarea propriu-zisa

input: ((P ∧ Q) ⇒ (¬R))

Ex. cu arbore:

Cate interpretari? 2

Interpretarea 1 a lui P :

A

Interpretarea 1 a lui Q :

A

Interpretarea 1 a lui R :

F

[[A∧A]⇒[¬F]]

[A⇒[¬F]]

[A⇒A]

A

Interpretarea 2 a lui P :

F

Interpretarea 2 a lui Q :

A

Interpretarea 2 a lui R :

F

[[F∧A]⇒[¬F]]

[F⇒[¬F]]

[F⇒A]

A

Ex. fara arbore:

Cate interpretari? 2

Interpretarea 1 a lui P :

A

Interpretarea 1 a lui Q :

A

Interpretarea 1 a lui R :

F

((A∧A)⇒(¬F))

(A⇒(¬F))

(A⇒A)

A

Interpretarea 2 a lui P :

F

Interpretarea 2 a lui Q :

A

Interpretarea 2 a lui R :

F

((F∧A)⇒(¬F))

(F⇒(¬F))

(F⇒A)

A