

Interpreter Programming Project
 CS 5361 Theory of Programming Languages
 Summer I 2016
 Due Date : to be determined

Below is the syntax and semantics for a single Boolean expression followed by a period. Write a program which prompts the user to input a file name which contains the Boolean expression or simply to input the string to be checked (indicate which input method you will use in your comments). You may assume that no input will be longer than 100 characters in length. Expressions may contain white spaces and white spaces should be considered to be delimiters (i.e. a white space between the $-$ and $>$ of the implication symbol would be a syntax error). The program should check if the expression in the file is of valid syntax and (if valid) compute the value of the expression. The output should either be an error message or a message that gives the value of the expression. **You must use the techniques taught in the class this is a recursive descent interpreter.**

Syntax: (note: for \vee use the lowercase letter "v" and for \wedge use the caret symbol)

		Selection Sets
$\langle B \rangle$	$::= \langle IT \rangle .$	$\{\sim, T, F, ()\}$
$\langle IT \rangle$	$::= \langle OT \rangle \langle IT_Tail \rangle$	$\{\sim, T, F, ()\}$
$\langle IT_Tail \rangle$	$::= - > \langle OT \rangle \langle IT_Tail \rangle$	$\{- >\}$
	$::= \varepsilon$	$\{.,)\}$
$\langle OT \rangle$	$::= \langle AT \rangle \langle OT_Tail \rangle$	$\{\sim, T, F, ()\}$
$\langle OT_Tail \rangle$	$::= \vee \langle AT \rangle \langle OT_Tail \rangle$	$\{\vee\}$
	$::= \varepsilon$	$\{- >, .,)\}$
$\langle AT \rangle$	$::= \langle L \rangle \langle AT_Tail \rangle$	$\{\sim, T, F, ()\}$
$\langle AT_Tail \rangle$	$::= \wedge \langle L \rangle \langle AT_Tail \rangle$	$\{\wedge\}$
	$::= \varepsilon$	$\{\vee, - >, .,)\}$
$\langle L \rangle$	$::= \langle A \rangle$	$\{T, F, ()\}$
	$::= \sim \langle L \rangle$	$\{\sim\}$
$\langle A \rangle$	$::= T$	$\{T\}$
	$::= F$	$\{F\}$
	$::= (\langle IT \rangle)$	$\{()\}$

Syntactic Domains:

$\langle B \rangle$: Bool_stmt
 $\langle IT \rangle$: Impl_y_term
 $\langle OT \rangle$: Or_term
 $\langle AT \rangle$: And_term
 $\langle IT_Tail \rangle$: Impl_y_tail
 $\langle OT_Tail \rangle$: Or_tail
 $\langle AT_Tail \rangle$: And_tail
 $\langle L \rangle$: Literal
 $\langle A \rangle$: Atom

Semantic Domain:

$b = \{T, F\}$ (Boolean values True and False)

Semantic Function Domains:

$\alpha : \text{Bool_stmt} \rightarrow b$

$\beta : \text{ImPLY_term} \rightarrow b$

$\delta : \text{Or_term} \rightarrow b$

$\gamma : \text{And_term} \rightarrow b$

$\lambda : b \times \text{ImPLY_tail} \rightarrow b$

$\mu : b \times \text{Or_tail} \rightarrow b$

$\eta : b \times \text{And_tail} \rightarrow b$

$\phi : \text{Literal} \rightarrow b$

$\psi : \text{Atom} \rightarrow b$

Semantic Equations:

$\alpha(\langle IT \rangle) = \beta(\langle IT \rangle)$

$\beta(\langle OT \rangle \langle IT_Tail \rangle) = \lambda(\delta(\langle OT \rangle), \langle IT_Tail \rangle)$

$\delta(\langle AT \rangle \langle OT_Tail \rangle) = \mu(\gamma(\langle AT \rangle), \langle OT_Tail \rangle)$

$\gamma(\langle L \rangle \langle AT_Tail \rangle) = \eta(\phi(\langle L \rangle), \langle AT_Tail \rangle)$

$\lambda(b, \varepsilon) = b$ (where $b \in \{T, F\}$)

$\lambda(F, \text{"- >" } \langle OT \rangle \langle IT_Tail \rangle) = \lambda(T, \langle IT_Tail \rangle)$

$\lambda(T, \text{"- >" } \langle OT \rangle \langle IT_Tail \rangle) = \lambda(\delta(\langle OT \rangle), \langle IT_Tail \rangle)$

$\mu(b, \varepsilon) = b$ (where $b \in \{T, F\}$)

$\mu(T, \text{"\vee" } \langle AT \rangle \langle OT_Tail \rangle) = T$

$\mu(F, \text{"\vee" } \langle AT \rangle \langle OT_Tail \rangle) = \mu(\gamma(\langle AT \rangle), \langle OT_Tail \rangle)$

$\eta(b, \varepsilon) = b$ (where $b \in \{T, F\}$)

$\eta(F, \text{"\wedge" } \langle L \rangle \langle AT_Tail \rangle) = F$

$\eta(T, \text{"\wedge" } \langle L \rangle \langle AT_Tail \rangle) = \eta(\phi(\langle L \rangle), \langle AT_Tail \rangle)$

$\phi(\text{"\sim" } \langle L \rangle) = \text{if } \phi(\langle L \rangle) = T \text{ then } F \text{ else if } \phi(\langle L \rangle) = F \text{ then } T$

$\phi(\langle A \rangle) = \psi(\langle A \rangle)$

$\psi(\text{"T"}) = T$

$\psi(\text{"F"}) = F$

$\psi(\text{"(" } \langle IT \rangle \text{"})"} = (\beta(\langle IT \rangle))$