Interpreter Programming Project
CS 5361 Theory of Programming Languages

Below is the syntax and semantics for a language that has a (possible empty) simple variable declaration secton followed by a single Boolean expression which is ended 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 input is of valid syntax and (if valid) compute the value of the expression. If syntactically valid, but the expression contains an undefined variable, it should give a runtime "undefinied variable" error rather than a syntax error. 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
$\langle P\rangle$	$::=$	$\langle D\rangle\langle P\rangle$	$\{\#, \sim, T, F, var, (\}$
$\langle D\rangle$	$::=$	$\# var := \langle V\rangle ; \langle D\rangle$	$\{\#\}$
	$::=$	ε	$\{\sim, T, F, var, (\}$
$\langle V\rangle$	$::=$	T	$\{T\}$
	$::=$	F	$\{F\}$
$\langle B\rangle$	$::=$	$\langle IT\rangle$.	$\{\sim, T, F, (\}$
$\langle IT\rangle$	$::=$	$\langle OT\rangle\ \langle IT\_Tail\rangle$	$\{\sim, T, F, var, (\}$
$\langle IT\_Tail\rangle$	$::=$	$->\ \langle OT\rangle\ \langle IT\_Tail\rangle$	$\{->\}$
	$::=$	ε	$\{.,)\}$
$\langle OT\rangle$	$::=$	$\langle AT\rangle\ \langle OT\_Tail\rangle$	$\{\sim, T, F, var, (\}$
$\langle OT\_Tail\rangle$	$::=$	$\vee\ \langle AT\rangle\ \langle OT\_Tail\rangle$	$\{\vee\}$
	$::=$	ε	$\{->, .,)\}$
$\langle AT\rangle$	$::=$	$\langle L\rangle\ \langle AT\_Tail\rangle$	$\{\sim, T, F, var, (\}$
$\langle AT\_Tail\rangle$	$::=$	$\wedge\ \langle L\rangle\ \langle AT\_Tail\rangle$	$\{\wedge\}$
	$::=$	ε	$\{\vee, ->, .,)\}$
$\langle L\rangle$	$::=$	$\langle A\rangle$	$\{T, F, var, (\}$
	$::=$	$\sim\langle L\rangle$	$\{\sim\}$
$\langle A\rangle$	$::=$	T	$\{T\}$
	$::=$	F	$\{F\}$
	$::=$	var	$\{var\}$
	$::=$	$(\ \langle IT\rangle\ )$	$\{(\}$

note: a *var* is a token which consists of a single lower case letter.

Syntactic Domains:

$\langle P\rangle$: Prog
$\langle D\rangle$: Dec_stmt
$\langle V\rangle$: Bool_val

$\langle B \rangle$: Bool_stmt
$\langle IT \rangle$: Imply_term
$\langle OT \rangle$: Or_term
$\langle AT \rangle$: And_term
$\langle IT\_Tail \rangle$: Imply_tail
$\langle OT\_Tail \rangle$: Or_tail
$\langle AT\_Tail \rangle$: And_tail
$\langle L \rangle$: Literal
$\langle A \rangle$: Atom

Semantic Domain:
$\tau : b = \{T.F\}$ (Boolean values True and False)
$\sigma : st = var \leftarrow b$ (State containing variable assignements)

Semantic Function Domains:
Ω : Prog $\rightarrow\ b$
Δ : Dec_stmt $\times\ st \rightarrow\ st$
ω : Bool_val $\rightarrow\ b$
α : Bool_stmt $\times\ st \rightarrow\ b$
β : Imply_term $\times\ st \rightarrow\ b$
δ : Or_term $\times\ st \rightarrow\ b$
γ : And_term $\times\ st \rightarrow\ b$
$\lambda : b \times$ Imply_tail $\times\ st \rightarrow\ b$
$\mu : b \times$ Or_tail $\times\ st \rightarrow\ b$
$\eta : b \times$ And_tail $\times\ st \rightarrow\ b$
ϕ : Literal $\times\ st \rightarrow\ b$
ψ : Atom $\times\ st \rightarrow\ b$

Semantic Equations:
$\Omega(\ll \langle D \rangle \langle B \rangle \gg) = \alpha(\ll \langle B \rangle \gg, \Delta(\ll \langle D \rangle \gg, \sigma[\ ]))$
$\Delta(\ll \#var := \langle V \rangle; \langle D \rangle \gg, \sigma) = \Delta(\ll \langle D \rangle, update(\sigma, \omega(\ll \langle V \rangle \gg)/var))$
$\Delta(\varepsilon, \sigma) = \sigma$
$\omega(\ll T \gg) = T$
$\omega(\ll F \gg) = F$
$\alpha(\ll \langle IT \rangle . \gg, \sigma) = \beta(\ll \langle IT \rangle \gg, \sigma)$
$\beta(\ll \langle OT \rangle \langle IT\_Tail \rangle \gg, \sigma) = \lambda(\delta(\ll \langle OT \rangle \gg, \sigma), \ll \langle IT\_Tail \rangle \gg, \sigma)$
$\delta(\ll \langle AT \rangle \langle OT\_Tail \rangle \gg, \sigma) = \mu(\gamma(\ll \langle AT \rangle \gg, \sigma), \ll \langle OT\_Tail \rangle \gg, \sigma)$
$\gamma(\ll \langle L \rangle \langle AT\_Tail \rangle \gg, \sigma) = \eta(\phi(\ll \langle L \rangle \gg, \sigma), \ll \langle AT\_Tail \rangle \gg, \sigma)$
$\lambda(b, \varepsilon, \sigma) = b$ (where $b \in \{T, F\}$)
$\lambda(F, \ll - > \langle OT \rangle \langle IT\_Tail \rangle \gg, \sigma) = \lambda(T, \ll \langle IT\_Tail \rangle \gg, \sigma)$
$\lambda(T, \ll - > \langle OT \rangle \langle IT\_Tail \rangle \gg, \sigma) = \lambda(\delta(\ll \langle OT \rangle \gg, \sigma), \ll \langle IT\_Tail \rangle \gg, \sigma)$
$\mu(b, \varepsilon, \sigma) = b$ (where $b \in \{T, F\}$)
$\mu(T, \ll \vee \langle AT \rangle \langle OT\_Tail \rangle \gg, \sigma) = T$
$\mu(F, \ll \vee" \langle AT \rangle \langle OT\_Tail \rangle \gg, \sigma) = \mu(\gamma(\ll \langle AT \rangle \gg, \sigma), \ll \langle OT\_Tail \rangle \gg, \sigma)$
$\eta(b, \varepsilon, \sigma) = b$ (where $b \in \{T, F\}$)
$\eta(F, \ll \wedge \langle L \rangle \langle AT\_Tail \rangle \gg, \sigma) = F$
$\eta(T, \ll \wedge \langle L \rangle \langle AT\_Tail \rangle \gg, \sigma) = \eta(\phi(\ll \langle L \rangle \gg, \sigma), \ll \langle AT\_Tail \rangle \gg, \sigma)$

$\phi(\ll\sim \langle L\rangle \gg, \sigma) = \text{if } \phi(\ll \langle L\rangle \gg, \sigma) = T \text{ then } F$
$\qquad\qquad \text{else if } \phi(\ll \langle L\rangle \gg, \sigma) = F \text{ then } T$
$\phi(\ll \langle A\rangle \gg, \sigma) = \psi(\ll \langle A\rangle \gg, \sigma)$
$\psi(\ll T \gg, \sigma) = T$
$\psi(\ll F \gg, \sigma) = F$
$\psi(\ll var \gg, \sigma) = \sigma[var]$ (the value of var in the state)
$\psi(\ll (\langle IT\rangle) \gg, \sigma) = (\beta(\ll \langle IT\rangle \gg, \sigma))$