$< HTML >$
$< PRE >$

MODULE $TLAPlus2Grammar$

EXTENDS $Naturals$, $Sequences$, $BNFGrammars$

$CommaList(L) \triangleq L \,\&\, (tok(\text{","}) \,\&\, L)^*$
$AtLeast4(s) \triangleq Tok(\{s \circ s \circ s\} \,\&\, \{s\}^+)$

$ReservedWord \triangleq$
{"ASSUME", "ELSE", "LOCAL", "UNION",
"ASSUMPTION", "ENABLED", "MODULE", "VARIABLE",
"AXIOM", "EXCEPT", "OTHER", "VARIABLES",
"CASE", "EXTENDS", "SF_", "WF_",
"CHOOSE", "IF", "SUBSET", "WITH",
"CONSTANT", "IN", "THEN",
"CONSTANTS", "INSTANCE", "THEOREM", "COROLLARY",
"DOMAIN", "LET", "UNCHANGED",
"BY", "HAVE", "QED", "TAKE",
"DEF", "HIDE", "RECURSIVE", "USE",
"DEFINE", "PROOF", "WITNESS", "PICK",
"DEFS", "PROVE", "SUFFICES", "NEW",
"LAMBDA", "STATE", "ACTION", "TEMPORAL",
"OBVIOUS", "OMITTED", "LEMMA", "PROPOSITION",
"ONLY"}

$Letter \triangleq OneOf(\text{"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"})$
$Numeral \triangleq OneOf(\text{"0123456789"})$
$NameChar \triangleq Letter \cup Numeral \cup \{\text{"_"}\}$

$Name \triangleq Tok((NameChar^* \,\&\, Letter \,\&\, NameChar^*) \setminus (\{\text{"WF_"}, \text{"SF_"}\} \,\&\, NameChar^+))$

$Identifier \triangleq Name \setminus Tok(ReservedWord)$

$IdentifierOrTuple \triangleq$
$Identifier \mid tok(\text{"<<"}) \,\&\, CommaList(Identifier) \,\&\, tok(\text{">>"})$

$NumberLexeme \triangleq$
$Numeral^+$
$\mid (Numeral^* \,\&\, \{\text{"."}\} \,\&\, Numeral^+)$
$\mid \{\text{"\\b"}, \text{"\\B"}\} \,\&\, OneOf(\text{"01"})^+$
$\mid \{\text{"\\o"}, \text{"\\O"}\} \,\&\, OneOf(\text{"01234567"})^+$
$\mid \{\text{"\\h"}, \text{"\\H"}\} \,\&\, OneOf(\text{"0123456789abcdefABCDEF"})^+$

$Number \triangleq Tok(NumberLexeme)$

$ProofStepId \triangleq Tok(\{\text{"<"}\} \,\&\, (Numeral^+ \mid \{\text{"*"}\}) \,\&\, \{\text{">"}\} \,\&\, (Letter \mid Numeral \mid \{\text{"_"}\})^+)$

$$BeginStepToken \triangleq Tok(\{\text{“<”}\} \,\&\, (Numeral^{+} \mid \{\text{“*”},\ \text{“+”}\}) \,\&\, \{\text{“>”}\} \,\&\, (Letter \mid Numeral)^{*} \,\&\, \{\text{“.”}\}^{*})$$

$$String \triangleq Tok(\{\text{“\”””}\} \,\&\, \text{STRING} \,\&\, \{\text{“\”””}\})$$

$$PrefixOp \triangleq Tok(\{\text{“-”},\ \text{“}\sim\text{”},\ \text{“\\lnot”},\ \text{“\\neg”},\ \text{“[]”},\ \text{“<>”},\ \text{“DOMAIN”},\ \text{“ENABLED”},\ \text{“SUBSET”},\ \text{“UNCHANGED”},\ \text{“UNION”}\})$$

$$InfixOp \triangleq Tok(\{\ \text{“!!”},\ \text{“\#”},\ \text{“\#\#”},\ \text{“\$”},\ \text{“\$\$”},\ \text{“\%”},\ \text{“\%\%”},\ \text{“\&”},\ \text{“\&\&”},\ \text{“(+)”},\ \text{“(-)”},\ \text{“(.)”},\ \text{“(/)”},\ \text{“(\\X)”},\ \text{“*”},\ \text{“**”},\ \text{“+”},\ \text{“++”},\ \text{“-”},\ \text{“-+->”},\ \text{“--”},\ \text{“-|”},\ \text{“..”},\ \text{“...”},\ \text{“/”},\ \text{“//”},\ \text{“/=”},\ \text{“/\\”},\ \text{“::=”},\ \text{“:=”},\ \text{“:>”},\ \text{“<”},\ \text{“<:”},\ \text{“<=>”},\ \text{“=”},\ \text{“=<”},\ \text{“=>”},\ \text{“=|”},\ \text{“>”},\ \text{“>=”},\ \text{“??”},\ \text{“@@”},\ \text{“\\”},\ \text{“\\/”},\ \text{“\^”},\ \text{“\^\^”},\ \text{“|”},\ \text{“|-”},\ \text{“|=”},\ \text{“||”},\ \text{“}\sim\text{>”},\ \text{“.”},\ \text{“<=”},\ \text{“\\approx”},\ \text{“\\geq”},\ \text{“\\oslash”},\ \text{“\\sqsupseteq”},\ \text{“\\asymp”},\ \text{“\\gg”},\ \text{“\\otimes”},\ \text{“\\star”},\ \text{“\\bigcirc”},\ \text{“\\in”},\ \text{“\\prec”},\ \text{“\\subset”},\ \text{“\\bullet”},\ \text{“\\intersect”},\ \text{“\\preceq”},\ \text{“\\subseteq”},\ \text{“\\cap”},\ \text{“\\land”},\ \text{“\\propto”},\ \text{“\\succ”},\ \text{“\\cdot”},\ \text{“\\leq”},\ \text{“\\sim”},\ \text{“\\succeq”},\ \text{“\\circ”},\ \text{“\\ll”},\ \text{“\\simeq”},\ \text{“\\supset”},\ \text{“\\cong”},\ \text{“\\lor”},\ \text{“\\sqcap”},\ \text{“\\supseteq”},\ \text{“\\cup”},\ \text{“\\o”},\ \text{“\\sqcup”},\ \text{“\\union”},\ \text{“\\div”},\ \text{“\\odot”},\ \text{“\\sqsubset”},\ \text{“\\uplus”},\ \text{“\\doteq”},\ \text{“\\ominus”},\ \text{“\\sqsubseteq”},\ \text{“\\wr”},\ \text{“\\equiv”},\ \text{“\\oplus”},\ \text{“\\sqsupset”},\ \text{“\\notin”}\ \})$$

$$PostfixOp \triangleq Tok(\{\text{“\^+”},\ \text{“\^*”},\ \text{“\^\#”},\ \text{“”}\})$$

$TLAPlusGrammar \triangleq$

LET $P(G) \triangleq$

$$\begin{aligned}
&\wedge\ G.Module ::= \ AtLeast4(\text{“-”}) \\
&\qquad \&\ tok(\text{“MODULE”}) \,\&\, Name \,\&\, AtLeast4(\text{“-”}) \\
&\qquad \&\ \ (Nil \mid (tok(\text{“EXTENDS”}) \,\&\, CommaList(Name))) \\
&\qquad \&\ \ (G.Unit)^{*} \\
&\qquad \&\ \ AtLeast4(\text{“=”}) \\
&\wedge\ G.Unit ::= \\
&\qquad\quad G.VariableDeclaration \\
&\qquad \mid\ G.ConstantDeclaration \\
&\qquad \mid\ G.Recursive \\
&\qquad \mid\ G.UseOrHide \\
&\qquad \mid\ (Nil \mid tok(\text{“LOCAL”})) \,\&\, G.OperatorDefinition
\end{aligned}$$

$$
\begin{array}{l}
\quad | \quad (Nil \mid tok(\text{“LOCAL”})) \,\&\, G.FunctionDefinition \\
\quad | \quad (Nil \mid tok(\text{“LOCAL”})) \,\&\, G.Instance \\
\quad | \quad (Nil \mid tok(\text{“LOCAL”})) \,\&\, G.ModuleDefinition \\
\quad | \quad G.Assumption \\
\quad | \quad G.Theorem \,\&\, (Nil \mid G.Proof) \\
\quad | \quad G.Module \\
\quad | \quad AtLeast4(\text{“-”})
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.VariableDeclaration ::= \\
\qquad Tok(\{\text{“VARIABLE”},\ \text{“VARIABLES”}\}) \,\&\, CommaList(Identifier)
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.ConstantDeclaration ::= \\
\qquad Tok(\{\text{“CONSTANT”},\ \text{“CONSTANTS”}\}) \,\&\, CommaList(G.OpDecl)
\end{array}
$$

$$
\wedge\ G.Recursive ::= tok(\text{“RECURSIVE”}) \,\&\, CommaList(G.OpDecl)
$$

$$
\begin{array}{lll}
\wedge\ G.OpDecl ::= & & Identifier \\
& | & Identifier \,\&\, tok(\text{“(”}) \,\& \\
& & \qquad CommaList(tok(\text{“\_”})) \,\&\, tok(\text{“)”}) \\
& | & PrefixOp \,\&\, tok(\text{“\_”}) \\
& | & tok(\text{“\_”}) \ \&\ InfixOp \,\&\, tok(\text{“\_”}) \\
& | & tok(\text{“\_”}) \ \&\ PostfixOp
\end{array}
$$

$$
\begin{array}{lll}
\wedge & G.OperatorDefinition ::= & \\
& \quad (& G.NonFixLHS \\
& \quad\ | & PrefixOp \quad \&\ Identifier \\
& \quad\ | & Identifier \quad \&\ InfixOp \,\&\ Identifier \\
& \quad\ | & Identifier \quad \&\ PostfixOp) \\
& \&\ & tok(\text{“==”}) \\
& \&\ & G.Expression
\end{array}
$$

$$
\begin{array}{lll}
\wedge\ G.NonFixLHS ::= & & \\
& Identifier & \\
\&\ & (\quad Nil & \\
& \quad\ | \quad tok(\text{“(”}) & \\
& \qquad \&\ CommaList(Identifier \mid\ G.OpDecl) & \\
& \qquad \&\ tok(\text{“)”}))
\end{array}
$$

$$
\begin{array}{lll}
\wedge\ G.FunctionDefinition ::= & & \\
& Identifier & \\
\&\ & tok(\text{“[”}) \,\&\, CommaList(G.QuantifierBound) \,\&\, tok(\text{“]”}) & \\
\&\ & tok(\text{“==”}) & \\
\&\ & G.Expression &
\end{array}
$$

$$
\begin{array}{lll}
\wedge\ G.QuantifierBound ::= & & \\
& (IdentifierOrTuple \mid CommaList(Identifier)) & \\
\&\ & tok(\text{“\textbackslash\textbackslash in”}) & \\
\&\ & G.Expression &
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.Instance ::= \\
\qquad tok(\text{“INSTANCE”}) \\
\quad \&\ \ Name \\
\quad \&\ \ (Nil \mid tok(\text{“WITH”}) \,\&\, CommaList(G.Substitution))
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.Substitution ::= \\
\qquad (Identifier \mid PrefixOp \mid InfixOp \mid PostfixOp) \\
\quad \&\ \ tok(\text{“<-”}) \\
\quad \&\ \ G.Argument
\end{array}
$$

$$
\wedge\ G.Argument ::= G.Expression \ \mid G.Opname \mid G.Lambda
$$

$$
\begin{array}{l}
\wedge\ G.Lambda \ ::= tok(\text{“LAMBDA”}) \,\&\, CommaList(Identifier) \\
\qquad\qquad\qquad\qquad \&\, tok(\text{“:”}) \,\&\, G.Expression
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.OpName ::= \\
\quad (Identifier \mid PrefixOp \mid InfixOp \mid PostfixOp \mid ProofStepId) \\
\ \ \&\,(\ tok(\text{“!”}) \\
\qquad \&\,(Identifier \mid PrefixOp \mid InfixOp \mid PostfixOp \\
\qquad\quad \mid Tok(\{\text{“<<”},\ \text{“>>”},\ \text{“@”}\} \cup Numeral^{+})) \\
\quad)^{*}
\end{array}
$$

$$
\wedge\ G.OpArgs ::= tok(\text{“(”}) \,\&\, CommaList(G.Argument) \,\&\, tok(\text{“)”})
$$

$$
\begin{array}{l}
\wedge\ G.InstOrSubexprPrefix ::= \\
\ \ (\quad (Nil \mid ProofStepId \,\&\, tok(\text{“!”})) \\
\quad \&\,((\quad Identifier \,\&\, (Nil \mid G.OpArgs) \\
\qquad\quad \mid Tok(\{\text{“<<”},\ \text{“>>”},\ \text{“:”}\} \cup Numeral^{+}) \\
\qquad\quad \mid G.OpArgs \\
\qquad\quad \mid (PrefixOp \mid PostfixOp) \,\&\, tok(\text{“(”}) \quad \&\, G.Expression \,\&\, tok(\text{“)”}) \\
\qquad\quad \mid InfixOp \,\&\, tok(\text{“(”}) \,\&\, G.Expression \,\&\, tok(\text{“,”}) \\
\qquad\qquad \&\, G.Expression \,\&\, tok(\text{“)”}) \\
\qquad)\ \ \&\ \ tok(\text{“!”}) \\
\quad)^{*} \\
\ \ )\setminus Nil
\end{array}
$$

$$
\wedge\ G.InstancePrefix ::= \ \ldots
$$

$$
\begin{array}{l}
\wedge\ G.GeneralIdentifier ::= \\
\qquad (G.InstOrSubexprPrefix \mid Nil) \,\&\, Identifier \\
\qquad \mid\ ProofStepId
\end{array}
$$

$$
\wedge\ G.GeneralIdentifier ::= \ \ldots
$$

$$
\wedge\ G.GeneralPrefixOp ::= \ \ldots
$$

$$
\wedge\ G.GeneralInfixOp ::= \ \ldots
$$

$$
\wedge\ G.GeneralPostfixOp ::= \ \ldots
$$

$$
\begin{array}{l}
\wedge\ G.ModuleDefinition ::= \quad G.NonFixLHS \\
\qquad\qquad\qquad\qquad\qquad\quad \& \ tok(\text{“==”}) \\
\qquad\qquad\qquad\qquad\qquad\quad \& \ G.Instance
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.Assumption ::= \\
\qquad Tok(\{\text{“ASSUME”},\ \text{“ASSUMPTION”},\ \text{“AXIOM”}\}) \\
\qquad\quad \& \ (Nil \mid Name \ \& \ tok(\text{“==”})) \ \& \ G.Expression
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.Theorem ::= \\
\qquad Tok(\{\text{“THEOREM”},\ \text{“PROPOSITION”},\ \text{“LEMMA”},\ \text{“COROLLARY”}\}) \\
\qquad\quad \& \ (Nil \mid Name \ \& \ tok(\text{“==”})) \ \& \ (G.Expression \mid G.AssumeProve)
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.AssumeProve ::= \quad tok(\text{“ASSUME”}) \\
\qquad\qquad\qquad\qquad\quad \& \ CommaList(G.Expression \mid G.New \mid G.InnerAssumeProve) \\
\qquad\qquad\qquad\qquad\quad \& \ tok(\text{“PROVE”}) \\
\qquad\qquad\qquad\qquad\quad \& \ G.Expression
\end{array}
$$

$$
\wedge\ G.InnerAssumeProve ::= (Nil \mid Name \ \& \ tok(\text{“::”})) \ \& \ G.AssumeProve
$$

$$
\begin{array}{l}
\wedge\ G.New ::= \quad ((\ (Nil \quad \mid tok(\text{“NEW”})) \\
\qquad\qquad\quad \& \ (Nil \mid tok(\text{“CONSTANT”}) \mid tok(\text{“VARIABLE”}) \mid tok(\text{“STATE”}) \\
\qquad\qquad\qquad\quad \mid tok(\text{“ACTION”}) \mid tok(\text{“TEMPORAL”})) \\
\qquad\qquad\ )\setminus Nil) \\
\qquad\quad \& \ ((Identifier \ \& \ tok(\text{“\textbackslash\textbackslash in”}) \ \& \ G.Expression) \mid G.OpDecl)
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.Proof ::= \quad G.TerminalProof \\
\qquad\qquad\quad \mid \ G.NonTerminalProof
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.TerminalProof ::= \quad (tok(\text{“PROOF”}) \mid Nil) \\
\qquad\qquad\qquad\qquad\quad \& \ (\ tok(\text{“BY”}) \ \& \ \ (tok(\text{“ONLY”}) \mid Nil) \ \& \ G.UseBody \\
\qquad\qquad\qquad\qquad\qquad\ \mid tok(\text{“OBVIOUS”}) \\
\qquad\qquad\qquad\qquad\qquad\ \mid tok(\text{“OMITTED”}) \\
\qquad\qquad\qquad\qquad\qquad\ )
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.NonTerminalProof ::= \\
\qquad\quad (Nil \mid tok(\text{“PROOF”})) \\
\qquad \& \ G.Step^{*} \\
\qquad \& \ G.QEDStep
\end{array}
$$

$$
\begin{array}{l}
\wedge\ G.Step ::= \\
\qquad\quad BeginStepToken \\
\qquad \& \ (\ (\ G.UseOrHide \\
\qquad\qquad \mid (\ (Nil \mid tok(\text{“DEFINE”})) \\
\qquad\qquad\quad \& \ (\ G.OperatorDefinition \\
\qquad\qquad\qquad\quad \mid G.FunctionDefinition \\
\qquad\qquad\qquad\quad \mid G.ModuleDefinition)^{+} \\
\qquad\qquad\quad)\\
\qquad\qquad \mid G.Instance
\end{array}
$$

$$
\begin{array}{l}
\quad | \; tok(\text{“HAVE”}) \,\&\, G.Expression \\
\quad | \; tok(\text{“WITNESS”}) \,\&\, CommaList(G.Expression) \\
\quad | \; tok(\text{“TAKE”}) \,\&\, (\;\; CommaList(G.QuantifierBound) \\
\qquad\qquad\qquad\qquad | \; CommaList(Identifier)) \\
\quad) \\
| \; (\;\; (\;\; (\;\; (Nil \,|\, tok(\text{“SUFFICES”})) \\
\qquad\quad \&\, (G.Expression \,|\, G.AssumeProve) \\
\qquad\quad) \\
\qquad | \, (tok(\text{“CASE”}) \,\&\, G.Expression) \\
\qquad | \, (\;\; tok(\text{“PICK”}) \\
\qquad\quad \&\, (CommaList(G.QuantifierBound) \,|\, CommaList(Identifier)) \\
\qquad\quad \&\, tok(\text{“:”}) \\
\qquad\quad \&\, G.Expression \\
\qquad\quad) \\
\qquad\quad) \\
\quad \&\, (Nil \,|\, G.Proof) \\
\quad) \\
) \\
\\
\wedge \; G.QEDStep ::= \\
\quad BeginStepToken \,\&\;\; tok(\text{“QED”}) \,\&\, (Nil \,|\, G.Proof) \\
\\
\wedge \; G.UseOrHide ::= \quad (\;\; (tok(\text{“USE”}) \,\&\, (Nil \,|\, tok(\text{“ONLY”}))) \\
\qquad\qquad\qquad\quad | \; tok(\text{“HIDE”})) \\
\qquad\qquad\qquad \&\, G.UseBody \\
\\
\wedge \; G.UseBody \;\; ::= \;\; (\;\; (Nil \quad | \, CommaList(G.Expression \,|\, tok(\text{“MODULE”}) \,\&\, Name)) \\
\qquad\qquad\qquad\quad \&\, (Nil \,|\, Tok(\{\text{“DEF”},\; \text{“DEFS”}\}) \\
\qquad\qquad\qquad\qquad\qquad \&\, CommaList(G.OpName \,| \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tok(\text{“MODULE”}) \,\&\, Name)) \\
\qquad\qquad\qquad) \setminus Nil \\
\\
\wedge \; G.Expression ::= \\
\\
\quad G.GeneralIdentifier \\
\\
\qquad\quad Name \,\&\, (Nil \,|\, tok(\text{“(”}) \,\&\, CommaList(Identifier) \,\&\, tok(\text{“)”})) \\
\qquad\qquad \&\;\; tok(\text{“::”}) \,\&\, G.Expression \\
\\
\qquad | \quad G.InstOrSubexprPrefix \\
\qquad\qquad\quad \&\, (Tok(\{\text{“<<”},\; \text{“>>”},\; \text{“:”}\} \cup Numeral^{+}) \,|\, G.OpArgs) \\
\\
\qquad | \quad G.GeneralIdentifier \,\&\, (Nil \,|\, G.OpArgs) \\
\\
\qquad | \quad PrefixOp \,\&\, G.Expression \\
\\
\qquad | \quad G.Expression \,\&\, InfixOp \,\&\, G.Expression
\end{array}
$$

```
  |  G.Expression & PostfixOp

  |  tok("(") & G.Expression & tok(")")

  |  Tok({"\\A", "\\E"})
     & CommaList(G.QuantifierBound)
     & tok(":")
     & G.Expression

  |  Tok({"\\A", "\\E", "\\AA", "\\EE"})
     & CommaList(Identifier)
     & tok(":")
     & G.Expression

  |  tok("CHOOSE")
     & IdentifierOrTuple
     & (Nil | tok("\\in") & G.Expression)
     & tok(":")
     & G.Expression

  |  tok("{")
     & (Nil | CommaList(G.Expression))
     & tok("}")

  |  tok("{")
     & IdentifierOrTuple & tok("\\in") & G.Expression
     & tok(":")
     & G.Expression
     & tok("}")

  |  tok("{")
     & G.Expression
     & tok(":")
     & CommaList(G.QuantifierBound)
     & tok("}")

  |  G.Expression & tok("[") & CommaList(G.Expression)
        & tok("]")

  |  tok("[")
     & CommaList(G.QuantifierBound)
     & tok("|->")
     & G.Expression
     & tok("]")

  |  tok("[") & G.Expression & tok("->")
           & G.Expression & tok("]")
```

```
  |    tok("[")
       & CommaList(Name & tok("|->") & G.Expression)
       & tok("]")

  |    tok("[")
       & CommaList(Name & tok(":") & G.Expression)
       & tok("]")

  |    tok("[")
       & G.Expression
       & tok("EXCEPT")
       & CommaList(    tok("!")
                     & (tok(".") & Name
                          |   tok("[") & CommaList(G.Expression) & tok("]"))+
                     &   tok("=") & G.Expression)
       & tok("]")

  |  G.Expression & tok(".") & Name

  |  tok("<<") & (CommaList(G.Expression) | Nil) & tok(">>")

  |  G.Expression & (Tok({"\\X", "\\times"})
         & G.Expression)+

  |  tok("[") & G.Expression & tok("]_")
       & G.Expression

  | tok("<<") & G.Expression & tok(">>_") & G.Expression

  |    Tok({"WF_", "SF_"})
       & G.Expression
       & tok("(") & G.Expression & tok(")")

  |    tok("IF")      & G.Expression
       & tok("THEN")  & G.Expression
       & tok("ELSE") & G.Expression

  |  tok("CASE")
     & (LET CaseArm ≜
                G.Expression & tok("->") & G.Expression
        IN   CaseArm & (tok("[]") & CaseArm)*)

     & (      Nil
          | (tok("[]") & tok("OTHER") & tok("->") & G.Expression))

|    tok("LET")
       & (    G.OperatorDefinition
            | G.FunctionDefinition
            | G.ModuleDefinition)+
```

& $tok(\text{“IN”})$
& $G.Expression$

| $(tok(\text{“/\\”}) \,\&\, G.Expression)^+$

| $(tok(\text{“\\/”}) \,\&\, G.Expression)^+$

| $Number$

| $String$

| $tok(\text{“@”})$

IN $LeastGrammar(P)$

$</PRE></HTML>$