CS 365 — Programming Language Concepts

Functional programming

Apr 9, 2008

LEm Concrete Syntax

$$
\begin{array}{rcl}
\mathit{Program} & \longrightarrow & \mathit{Expression}\ \mathsf{;} \\
\mathit{Expression} & \longrightarrow & \mathit{Application} \\
\mathit{Application} & \longrightarrow & \mathit{PrimaryExpression}\ \{\ \text{“(”}\ \mathit{Expression}\ \text{“)”}\ \}\ \mathsf{*} \\
\mathit{PrimaryExpression} & \longrightarrow & \mathit{Identifier}\ |\ \mathit{Abstraction}\ |\ \text{“(”}\ \mathit{Expression}\ \text{“)”} \\
\mathit{Abstraction} & \longrightarrow & \mathsf{fn}\ \text{“(”}\ \mathit{Identifier}\ \text{“)”}\ {=}{>}\ \mathit{Expression}
\end{array}
$$

BoolEm Concrete Syntax

$$
\begin{array}{rcl}
\mathit{Program} & \longrightarrow & \mathit{Expression}\ ; \\
\mathit{Expression} & \longrightarrow & \underline{\mathit{Condition} \mid \mathit{Disjunction}} \\
\underline{\mathit{Condition}} & \longrightarrow & \underline{\textsf{if}\ \mathit{Expression}\ \textsf{then}\ \mathit{Expression}\ \textsf{else}\ \mathit{Expression}} \\
\underline{\mathit{Disjunction}} & \longrightarrow & \underline{\mathit{Conjunction}\ \{\ \textsf{orelse}\ \mathit{Conjunction}\ \}*} \\
\underline{\mathit{Conjunction}} & \longrightarrow & \underline{\mathit{Negation}\ \{\ \textsf{andalso}\ \mathit{Negation}\ \}*} \\
\underline{\mathit{Negation}} & \longrightarrow & \underline{\{\ \textsf{not}\ \}_{\textsf{opt}}\ \mathit{Application}} \\
\mathit{Application} & \longrightarrow & \mathit{PrimaryExpression}\ \{\ \text{“(”}\ \mathit{Expression}\ \text{“)”}\ \}\ * \mid \text{“(”}\ \mathit{Expression}\ \text{“)”} \\
\mathit{PrimaryExpression} & \longrightarrow & \mathit{Identifier} \mid \mathit{Literal} \mid \mathit{Abstraction} \\
\mathit{Abstraction} & \longrightarrow & \textsf{fn}\ \text{“(”}\ \mathit{Identifier}\ \text{“)”}\ => \mathit{Expression}
\end{array}
$$

DeclEm Concrete Syntax

$$
\begin{array}{rcl}
\textit{Program} & \longrightarrow & \textit{Expression}\ ; \\
\textit{Expression} & \longrightarrow & \textit{Condition} \mid \textit{Disjunction} \mid \underline{\textit{Let}} \\
\underline{\textit{Let}} & \longrightarrow & \underline{\textsf{let}\ \{\ \textit{Declaration}\ \}\ *\ \textsf{in}\ \textit{Expression}\ \textsf{end}} \\
\underline{\textit{Declaration}} & \longrightarrow & \underline{\textsf{val}\ \textit{Identifier} = \textit{Expression}\ ;} \\
\textit{Condition} & \longrightarrow & \textsf{if}\ \textit{Expression}\ \textsf{then}\ \textit{Expression}\ \textsf{else}\ \textit{Expression} \\
\textit{Disjunction} & \longrightarrow & \textit{Conjunction}\ \{\ \textsf{orelse}\ \textit{Conjunction}\ \}* \\
\textit{Conjunction} & \longrightarrow & \textit{Negation}\ \{\ \textsf{andalso}\ \textit{Negation}\ \}* \\
\textit{Negation} & \longrightarrow & \{\ \textsf{not}\ \}_{\textsf{opt}}\ \textit{Application} \\
\textit{Application} & \longrightarrow & \textit{PrimaryExpression}\ \{\ \text{“(”}\ \underline{\textit{Expressions}}\ \text{“)”}\ \}\ * \\
\underline{\textit{Expressions}} & \longrightarrow & \underline{\textit{Expression}\ \{\ ,\ \textit{Expression}\ \}\ *} \\
\textit{PrimaryExpression} & \longrightarrow & \textit{Identifier} \mid \textit{Literal} \mid \textit{Abstraction} \mid \text{“(”}\ \textit{Expression}\ \text{“)”} \\
\textit{Abstraction} & \longrightarrow & \textsf{fn}\ \text{“(”}\ \underline{\textit{Identifiers}}\ \text{“)”} => \textit{Expression} \\
\underline{\textit{Identifiers}} & \longrightarrow & \underline{\textit{Identifier}\ \{\ ,\ \textit{Identifier}\ \}}
\end{array}
$$

Em Concrete Syntax

$$
\begin{array}{rcl}
\mathit{Program} & \longrightarrow & \mathit{Expression}\ ; \\
\mathit{Expression} & \longrightarrow & \mathit{Condition} \mid \mathit{Disjunction} \mid \mathit{Let} \\
\mathit{Let} & \longrightarrow & \textsf{let}\ \{\ \mathit{Declaration}\ \} * \ \textsf{in}\ \mathit{Expression}\ \textsf{end} \\
\mathit{Declaration} & \longrightarrow & \underline{\ \mathit{VariableDeclaration} \mid \mathit{FunctionDeclaration}\ } \\
\underline{\mathit{VariableDeclaration}} & \longrightarrow & \textsf{val}\ \mathit{Identifier} = \mathit{Expression}\ ; \\
\underline{\mathit{FunctionDeclaration}} & \longrightarrow & \underline{\textsf{fun}\ \mathit{Identifier}\ \text{“(”}\ \mathit{Identifiers}\ \text{“)”} = \mathit{Expression}\ ;} \\
\mathit{Condition} & \longrightarrow & \textsf{if}\ \mathit{Expression}\ \textsf{then}\ \mathit{Expression}\ \textsf{else}\ \mathit{Expression} \\
\mathit{Disjunction} & \longrightarrow & \mathit{Conjunction}\ \{\ \textsf{orelse}\ \mathit{Conjunction}\ \}* \\
\mathit{Conjunction} & \longrightarrow & \mathit{Negation}\ \{\ \textsf{andalso}\ \mathit{Negation}\ \}* \\
\mathit{Negation} & \longrightarrow & \{\ \textsf{not}\ \}_{\textsf{opt}}\ \mathit{Application} \\
\mathit{Application} & \longrightarrow & \mathit{PrimaryExpression}\ \{\ \text{“(”}\ \mathit{Expressions}\ \text{“)”}\ \} * \\
\mathit{Expressions} & \longrightarrow & \mathit{Expression}\ \{\ ,\ \mathit{Expression}\ \} * \\
\mathit{PrimaryExpression} & \longrightarrow & \mathit{Identifier} \mid \mathit{Literal} \mid \mathit{Abstraction} \mid \text{“(”}\ \mathit{Expression}\ \text{“)”} \\
\mathit{Abstraction} & \longrightarrow & \textsf{fn}\ \text{“(”}\ \mathit{Identifiers}\ \text{“)”} => \mathit{Expression} \\
\mathit{Identifiers} & \longrightarrow & \mathit{Identifier}\ \{\ ,\ \mathit{Identifier}\ \}
\end{array}
$$

ListEm Concrete Syntax

$$\begin{array}{rcl}
\mathit{Program} & \longrightarrow & \mathit{Expression}\ ; \\
\mathit{Expression} & \longrightarrow & \mathit{Condition} \mid \mathit{Disjunction} \mid \mathit{Let} \\
\mathit{Let} & \longrightarrow & \textsf{let}\ \{\ \mathit{Declaration}\ \}*\ \textsf{in}\ \mathit{Expression}\ \textsf{end} \\
\mathit{Declaration} & \longrightarrow & \mathit{VariableDeclaration} \mid \mathit{FunctionDeclaration} \\
\mathit{VariableDeclaration} & \longrightarrow & \textsf{val}\ \mathit{Identifier} = \mathit{Expression}\ ; \\
\mathit{FunctionDeclaration} & \longrightarrow & \textsf{fun}\ \mathit{Identifier}\ \text{“(”}\ \mathit{Identifiers}\ \text{“)”} = \mathit{Expression}\ ; \\
\mathit{Condition} & \longrightarrow & \textsf{if}\ \mathit{Expression}\ \textsf{then}\ \mathit{Expression}\ \textsf{else}\ \mathit{Expression} \\
\mathit{Disjunction} & \longrightarrow & \mathit{Conjunction}\ \{\ \textsf{orelse}\ \mathit{Conjunction}\ \}* \\
\mathit{Conjunction} & \longrightarrow & \underline{\mathit{Cons}}\ \{\ \textsf{andalso}\ \underline{\mathit{Cons}}\ \}* \\
\underline{\mathit{Cons}} & \longrightarrow & \underline{\mathit{Negation}\ ::\ \mathit{Expression}} \\
\mathit{Negation} & \longrightarrow & \{\ \textsf{not}\ \}_{\textsf{opt}}\ \underline{\mathit{Test}} \\
\underline{\mathit{Test}} & \longrightarrow & \mathit{Application}\ \underline{\{\ =\textsf{nil}\ \}_{\textsf{opt}}} \\
\mathit{Application} & \longrightarrow & \mathit{PrimaryExpression}\ \{\ \text{“(”}\ \mathit{Expressions}\ \text{“)”}\ \}\ * \\
\mathit{Expressions} & \longrightarrow & \mathit{Expression}\ \{\ ,\ \mathit{Expression}\ \}\ * \\
\mathit{PrimaryExpression} & \longrightarrow & \mathit{Identifier} \mid \mathit{Literal} \mid \mathit{Abstraction}\ \underline{\mid \mathit{List} \mid \mathit{Car} \mid \mathit{Cdr}} \\
 & & \mid\ \text{“(”}\ \mathit{Expression}\ \text{“)”} \\
\underline{\mathit{List}} & \longrightarrow & \underline{[\ \mathit{Expressions}\ ]} \\
\underline{\mathit{Car}} & \longrightarrow & \underline{\textsf{hd}\ \text{“(”}\ \mathit{Expression}\ \text{“)”}} \\
\underline{\mathit{Cdr}} & \longrightarrow & \underline{\textsf{tl}\ \text{“(”}\ \mathit{Expression}\ \text{“)”}} \\
\mathit{Abstraction} & \longrightarrow & \textsf{fn}\ \text{“(”}\ \mathit{Identifiers}\ \text{“)”} => \mathit{Expression} \\
\mathit{Identifiers} & \longrightarrow & \mathit{Identifier}\ \{\ ,\ \mathit{Identifier}\ \}
\end{array}$$