Program	$\rightarrow$	*Expression* `;`
Expression	$\rightarrow$	*TailFormExpression* | *SimpleExpression*
SimpleExpression	$\rightarrow$	*Disjunction* | *SimpleLet* | *SimpleConditional*
Disjunction	$\rightarrow$	*Conjunction* { `orelse` *Conjunction* } `*`
Conjunction	$\rightarrow$	*Negation* { `andalso` *Negation* } `*`
Negation	$\rightarrow$	{ `not` }? *PrimaryExpression*
SimpleLet	$\rightarrow$	`let` { *Declaration* }+ `in` *SimpleExpression* `end`
Declaration	$\rightarrow$	*VariableDeclaration* | *FunctionDeclaration*
VariableDeclaration	$\rightarrow$	`val` ID `=` *SimpleExpression* `;`
FunctionDeclaration	$\rightarrow$	`fun` ID `(` *FormalParams* `)` `=` *TailFormExpression* `;`
SimpleConditional	$\rightarrow$	`if` *SimpleExpression* `then` *SimpleExpression* `else` *SimpleExpression*
TailFormExpression	$\rightarrow$	*Application* | *Let* | *Condition*
Let	$\rightarrow$	`let` { *Declaration* }+ `in` *TailFormExpression* `end`
Abstraction	$\rightarrow$	`fn` `(` *FormalParams* `)` `=>` *TailFormExpression*
FormalParams	$\rightarrow$	ID { `,` ID } `*`
Conditional	$\rightarrow$	`if` *SimpleExpression* `then` *TailFormExpression* `else` *TailFormExpression*
Application	$\rightarrow$	*SimpleExpression* `(` *SimpleExpressions* `)`
SimpleExpressions	$\rightarrow$	*SimpleExpression* { `,` *SimpleExpression* } `*`
PrimaryExpression	$\rightarrow$	*Variable* | *ParenthesizedExpression* | LITERAL
Variable	$\rightarrow$	ID
ParenthesizedExpression	$\rightarrow$	`(` *SimpleExpression* `)`