Appendix A: SimpliPy BNF Grammar

```
<pgm> ::= <block>

<block> ::= <stmt>+

<stmt> ::= <exp-assign-stmt>
         | <call-assign-stmt>
         | <if-stmt>
         | <while-stmt>
         | <def-stmt>
         | <ret-stmt>
         | <pass-stmt>
         | <break-stmt>
         | <continue-stmt>
         | <global-stmt>
         | <nonlocal-stmt>

<exp-assign-stmt> ::= <id> = <exp>
<call-assign-stmt> ::= <id> = <id> ( <exp> ... )
<if-stmt> ::= if <exp>: <block> else: <block>
<while-stmt> ::= while <exp>: <block>
                 (* Constraint: Block must end with continue *)
<def-stmt> ::= def <id> ( <id> ... ): <block>
               (* Constraint: Block must end with return *)
<ret-stmt> ::= return <exp>
<pass-stmt> ::= pass
<break-stmt> ::= break
<continue-stmt> ::= continue
<global-stmt> ::= global <id> ...
<nonlocal-stmt> ::= nonlocal <id> ...

<exp> ::= <const>
        | <id>
        | <exp> <binop> <exp>
        | <unop> <exp>
        | ( <exp> )

<const> ::= int | float | str | bool | None
<unop> ::= - | + | not
<binop> ::= + | - | * | / | // | % | **
          | > | < | == | <= | >= | !=
          | and | or
          | is | is not | in | not in
```