```
<program> → <function>[[<function>]]
<function> → int <identifier>(<parameter-list>) <compound-statement>
<function> → int <identifier>(<parameter-list>);
<parameter-list> → [<parameter>[[,<parameter>]]]
<parameter> → int <declarator>
<declarator> → <identifier>[[[<number>]]]
<compound-statement> → {<declaration-list><statement-list>}
<declaration-list> → [[<declaration>]]
<declaration> → int <declarator-list>;
<declarator-list> → <declarator>[[,<declarator>]]
<statement-list> → [[<statement>]]
<statement> → <assignment-statement> | <compound-statement> | <if-statement> |
              <while-statement> | <return-statement> | <read-statement> |
              <write-statement> | <writeln-statement>
<assignment-statement> → <variable> = <logical-expression>;
<logical-expression> → [!]<conditional-expression>
<conditional-expression> → <expression>[<relative-operator><expression>]
<expression> → <term>[[<additive-operator><term>]]
<term> → <factor>[[<multiplicative-operator><factor>]]
<factor> → <variable> | (<logical-expression>) | <identifier>(<argument-list>) |
           <number>
<variable> → <identifier>[[[<logical-expression>]]]
<argument-list> → [<logical-expression>[[,<logical-expression>]]]
<if-statement> → if (<logical-expression>) <statement> else <statement>
<while-statement> → while (<logical-expression>) <statement>
<return-statement> → return <logical-expression>;
<read-statement> → read <variable>;
<write-statement> → write <logical-expression>;
<writeln-statement> → writeln;
<relative-operator> → == | != | < | > | <= | >=
<additive-operator> → + | -
<multiplicative-operator> → * | / | %
<identifier> → <alphabetic-character>[[<alphabetic-numeric-character>]]
<number> → <numeric-character>[[<numeric-character>]]
```

図 A.1　tiny-C 言語の文法規則