実践的プログラミング

C0115253 新田　剛志

1. 字句解析

字句解析とは素のプログラムを読み込んで、書かれている単語のまとまりを作り、１つ１つにどういう意味があるのか判明させる処理です。

1.字句の始まりとなり得る文字(英数字, 記号, 改行文字) を得る.

2.その文字が, 変数or 予約語, 数字, リテラル, 演算子のどれに該当するのか判断する.

3.対応する種別に応じて字句に許容されない文字が現れるまでソースを読み, 初めの字から許容されない 文字が現れるまでの文字をつなげた物を１字句の内容とする.

4.得られた文字列に対して実際にどのような字句なのか, 予約語なのか, どの演算子な　　のか等をより細かく 見て字句を確定し出力する.

上記の手順を繰り返すことで字句をひとつずつ取り出す。

LexicalUnitでは定義的に正規表現を埋め込んだこのEnumを使って処理を行う。

Tokenparserのクラスはソースコードを一文字ずつよんでTokenに分割する.

下のクラスは ソースコード（テキスト）を１文字ずつよんで、なんか意味ある始めの一字（EOFやスペースではなく次のTokenを構成する１字） まで読み、上のEnumのマッチを使ってどれか確定させ、tokenにします。

1. 構文解析

　 構文解析とは、字句の並びから、プログラム言語の文法(Syntax)と照らし合わせAST（抽象構文木)を作ること.

　非終端要素についてクラスを１つ１つ作ります。 それぞれのクラスはNodeクラスを継承するようにします。

Nodeクラスは抽象クラスとしparse()の関数実装を強制する。

Nodeクラスを継承した非終端要素のクラスでisMatch()とparse()を実装する。

isMatchは、与えられた字句が、その非終端要素のファースト集合に含まれるかどうかを判定し、含まれるならば自身をインスタンス化して返し。 また、今回の文法はLL2である（２個先まで読まないとどの文法規則を適用すればいいか確定しない cf 代入文(NAME => EQ)と関数呼び出し(NAME => LP))ので そういう重複がありえる非終端要素の場合は確定させるまで読んで返えす。

parse()は子のisMatch(stmtlistのisMatch)を呼び、自身のフィールドに子のインスタンスを保持する。 こうして、ProgramがフィールドにStmtListのインスタンスを持ち、StmtListがフィールドにAssignStmtやLoopBlockのインスタンスを持ち、順々につなげていくことでリンクするように、構文木が表現できる。

StmtListNodeではStmtListも同じように、ファースト集合はStmtNodeとBlockNodeのファースト集合の和集合になるので、ProgramNodeと同じように定義する。 parseもStmtListの下はStmtNodeかBlockNodeなので、どちらか調べてさらにそれぞれparseにする。 子のparseに成功したら、自身のフィールドで子ノードとして保持しておく。

LoopBlockNodeでは下の非終端要素になると、終端要素をチェックが始まる。DO や　UNTILなどの順番をチェックする必要がある.

ExprNodeでは一式を読み逆ポーランド記法変換する.また二逆ポーランド記法の計算アルゴリズムを利用しツリーを生成させ,三生成したツリーをインスタンスに適用する.

1. 実行

文字通りプログラムを実行することです。 例えば a=1というコードであれば実際に、変数aを用意し、そこに1という値を代入する処理を行う。また`PRINT "Hello"であればPRINT関数を実行する。

LoopBlockクラスはevel()メソッドは下記の処理をする.

•子のCondNodeをevalしループ実行判定

•実行するならStmtListを実行

•繰り返す

ExprNodeクラスのeval()メソッドであれば単純に式を計算して１つの値を返すようにする。

それぞれの構文要素のクラスにあった計算方法をそれぞれのクラスのevalとして実装すること

ProgramNodeのevalを呼ぶと再帰的に実行が行える.

Enviromentクラスでは変数テーブル,関数テーブルを持たせ,変数を問い合わせする処理や新規定義する処理,関数を呼び出す処理を行う.

IF文などの条件判定ノードやループノードの実行は小のCondNodeの評価を行って、ループを行い、どのStmtを実行するか決める.

1. まとめ

字句解析,構文解析,実行系の順にプログラムを作っていくことで読み込んだプログラムの状態遷移図を作成し、最適化を施し、その状態遷移を実現するようにJavaのコードを書けば字句判定が完成する.状態遷移図からJavaのコードをIFやループを使って生み出すのに時間がかかるので、 字句を正規表現で表した後、Javaの標準ライブラリに正規表現処理を行う.