

第2章 コンパイラーのフェーズ

2.1 コンパイラーの構成

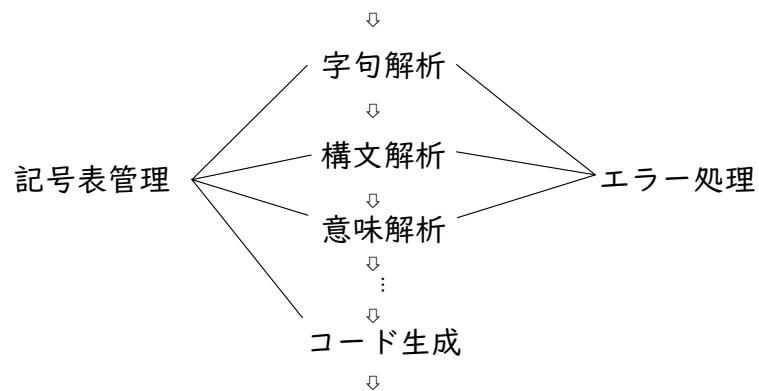
コンパイラーはいくつかの _____ にわけることができる

1. 字句解析 (lexical analysis)
2. 構文解析 (syntax analysis, parsing)
3. _____ (_____) (semantic analysis, static analysis)
 _____ など、字句解析・構文解析では見つけられない間違いを発見する。
 - 誤り検知（引数の数や型など）
 - オーバーローディング（多重定義）の解決
 例えば + は double の演算, int の演算?
 - 自動型変換の挿入
 $\underline{2} * 3.14 \cdots \text{int} \rightarrow \text{double}$ の変換を挿入する必要がある。
4. _____ 中間語は理想的なコンピューターの機械語と考えることができる。「理想的」とは例えば「レジスターが無限にある」などである。
5. _____ (code optimization)
 生成したコードを、効率のよい形へ変換する。

注意: 「最適」という表現を使うが、本当に「最適」にするのはそもそも無理である。
6. _____ レジスター割り付け (register allocation, レジスターをどのように使うかを決める) を行い、機械語を生成する。

意味解析までを _____ 、中間語生成以降を _____ と呼ぶことがある。

2.2 全体図



2.3 記号表

記号表 (symbol table) とは、識別子 (identifier, ソースプログラム中に出現する名前)に関する情報の表である。型、ソースプログラム中の場所、アドレス、スコープなどさまざまな情報を含む。
