

抽象度の高い CASE ツール記述を可能にする 代数データ型による C 言語ソースのモデルの提案

情報科学科 小嶋 一成

指導教員：山本 晋一郎

1 はじめに

CASE ツールプラットフォーム Sapid[1] は、CASE ツール作成の際に必要な字句・構文・意味解析の結果を C 言語ソースのモデルとして提供する。また、Sapid よりも利便性の高い C 言語ソースのモデルとして、Language.C[2] の代数データ型 AST モデルがある。しかし、代数データ型 AST モデルは Sapid と比べて保持できる情報が少ない。そこで Sapid と同等の情報を保持できるように代数データ型 AST モデルを拡張した。C 言語ソースからモデルを生成する部分は、本来自前で構文解析を行うべきだが、今回は Sapid のモデルを提案モデルに変換することで実現した。

2 CASE ツールと CASE ツールプラットフォーム

CASE ツールとは、ソフトウェア開発に使用されるツールのことである。一般的には、開発の上流工程を支援する上流 CASE ツールと、プログラミングやテストなどの下流工程を支援する下流 CASE ツールに大別される。

特に下流 CASE ツールの内部には、対象プログラム言語の字句解析器や構文解析器が自然に必要な。しかし、そのような解析器は、CASE ツールの本質ではない。様々な CASE ツールに共通して用いられる解析器を提供する CASE ツールプラットフォームは、CASE ツール作成の負担を軽減できる。このような CASE ツールプラットフォームとして Sapid がある。

Sapid の中核は、C 言語のソースプログラムを、13 種類の实体と 29 種類の関連としてモデル化した関係モデルを保持したソフトウェアデータベース (SDB) と、そのアクセスルーチン (AR) である。字句や構文の情報だけでなく、同じ変数を意味するものには同じ ID を付ける、といった意味解析も行われる。CASE ツール作成に必要な字句解析や構文解析の結果はすべて SDB に保存されているので、CASE ツール作成者は本質部分の開発に注力できる。

しかし、Sapid が提供する関係モデルでは、C 言語の構成要素へのアクセス処理が煩雑になるという問題がある。ここではこのアクセス処理のことをノード指定処理と呼ぶ。モデルは、ノード指定処理のアルゴリズムに深く関わる。ノード指定処理を簡潔に記述できるモデルを提供することで、CASE ツールプラットフォームの利便性が向上する。

3 C 言語ソースのモデル

ノード指定処理を簡潔に記述できるモデルとして、Haskell で C 言語ソースを扱うライブラリである Language.C の代数データ型 AST モデルに着目した。

代数データ型による AST モデルは、宣言や列挙型など 28 種類の代数データ型ノードの木構造で構成される。しかし、マルチプラットフォームな開発等に欠かせない前処理の情報を表現することができないという制約がある。また、同時に解析可能なソースファイルは 1 つまでという制約がある。

本研究では、Language.C のモデルを改良して、より多くの情報を扱うことが出来る AST モデルを提案する。提案モデルは、Language.C のモデルに、マクロとプリプロセッサを表すノードと、ソースファイルを表すノードを加えた、計 34 種類のノードの木構造で構成される。これにより、Language.C の前処理後のソースファイル 1 つのみという制約が撤廃される。また、C 言語の前処理系は C 言語の文法とは独立しているため、一般的な解析器では前処理情報を AST に含めることはできない。本研究では開発者視点により近いモデルを生成するため、展開した

コードが単体で C 言語の文法を満たすという条件を付けることで、前処理情報を AST に含められるようにした。

4 C プログラムからのモデルの構築

Language.C では意味解析の機能を持たない。一方 Sapid では意味解析が実装されている。本研究では意味解析機能を利用するため、Sapid の関係モデルを提案モデルに変換することで実現した。しかし、前処理前の情報は未実装となっている。また、Sapid では GCC 拡張と C99 の C 言語規格に対応していないため、関係モデルからの変換という実装ではこれらの文法を利用したモデルを生成できない。これらの文法は Language.C では実装済みであり、提案モデルでもモデル上では表現可能である。

5 評価

表 1 は、Sapid, Language.C, 提案モデルと、提案モデルの実装における機能を比較したものである。対応しているものについている。

ノード指定のステップ数は、定義された関数の名前を出力する CASE ツールを作成し、その本質部分の関数のステップ数を計測した。Sapid では CASE ツール記述の言語が他と違い C 言語であるが、公平に比較を行うため、同様のアプローチを Haskell で記述した擬似コードのステップ数を計測した。なお、本来の C 言語で記述した場合のステップ数は 12 である。

Sapid と提案モデルを比較すると、ノード指定のステップ数が約半分になっており、簡潔な処理でノード指定処理を記述可能となっている。

表 1 Sapid, Language.C, 提案モデル, 実装の比較

	Sapid	Language.C	提案モデル	実装
CASE ツール記述言語	C	Haskell	Haskell	Haskell
意味解析				
複数ソースファイル				
前処理前の情報				
GCC 拡張・C99 規格				
ノード指定のステップ数	11	4	4	4

6 おわりに

本研究は、抽象度の高い CASE ツール記述を可能にするため、代数データ型による C 言語ソースのモデルを提案した。具体的には、既存の Language.C が採用している C 言語ソースモデルを拡張し、複数ソースファイルへの対応と一部の前処理前のコードに対応できるようになった。しかし、一部の機能についてはモデルのみの対応となっており、未実装となっている。今後の課題として、Sapid で解析可能だが未実装となっている情報を全て取得できるように改良する。さらに、Sapid からの変換という実装から自前での字句・構文解析を行う実装へ改良することが挙げられる。

参考文献

- [1] 福安直樹; 山本晋一郎; 阿草清滋. 細粒度リポジトリに基づいた CASE ツール・プラットフォーム Sapid. 情報処理学会論文誌, 1998, 39.6: 1990-1998.
- [2] Language.C, <http://trac.sivity.net/language.c/>