3M−1 C言語の問題向き拡張システムOPTECの高速化

小島泰三 杉本明 阿部茂
三菱電機株式会社 中央研究所

1 はじめに

C言語を問題向きに拡張するためのプログラム言語拡張システムOPTECを開発した．以前より試作システムを用いた実現方式の検討を行なっており[1, 2, 3]，その経験を元に実用システムを新たに設計し，実装した．

本システムは，変換システム生成系と実行支援ライブラリから構成される．本システムの特長は，データ型をパタン変換に導入し、このパタン変換の枠組においてプログラム変換を行なっていることである．そして，パタン変換によるシンタクスの拡張，及びオーバローディングによるセマンティクスの拡張を単一の枠組で実現することを可能とした．本システムでは，このプログラム言語拡張機能を用い，C言語を問題向きに柔軟に拡張することができる．

本稿では，実用版 OPTEC の概要，そして本システムで採用した高速なプログラム変換処理方式について説明する．

2 システム概要

OPTEC システムは問題向きに拡張されたC言語表現から通常のC言語表現へのパタン変換を行なう変換システムの生成系と，変換実行をサポートするランタイムライブラリから構成されるC言語拡張システムである．図 1に OPTEC システムの構成を示す．

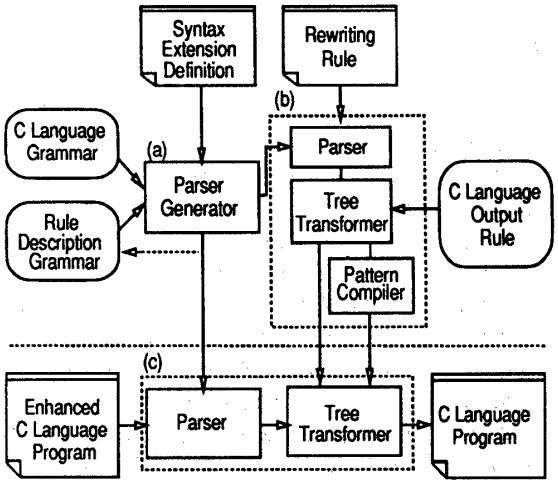

図 1: システム構成

本システムにおける処理は、以下の3つのステップから構成される．

1. パーサジェネレータ (図 1(a)) を用いて構文解析部の生成を行なう．このときルールコンパイラ (図 1(b)) 用及び変換システム (図 1(c)) 用の2種類を生成する．本ステップでは，ベースであるC言語のシンタクス定義と追加拡張するシンタクス定義を用い，YACC ソースファイルと初期化データを生成する．
2. 変換システムの生成を行なう．このステップでは，まず先のステップで生成した構文解析部を組み込んだルールコンパイラを用いて，変換ルール定義をC言語プログラムに変換する．次に，変換システム用の構文解析部と本ステップで生成したC言語プログラムをコンパイルリンクし，変換システムを生成する．
3. 先のステップで生成した変換システムを用いて，問題向きに拡張されたC言語ソースから通常のC言語表現への変換を行なう．この処理では，問題向きに拡張された記述の構文ツリーと，変換ルールの構文ツリーとの間でパタンマッチングを行い適用可能ルールを求める．ルールが検出されるとマッチングしたツリーに対してルールを適用し，ツリー書き換えを行なう．なおマッチングするルールがない場合は，無変換で出力する．

試作システムは通常のCプリプロセッサのように，1つのプログラムとして実現していた．これに対し実用版では，問題向き拡張を統制でき，また変換速度の高速化が要求された．このため，(1) エンドプログラマには変換システムのみを提供し，問題向き拡張を集中的に管理できる，(2) 変換ルールの読み込みやパタンマッチングの準備を前処理として分離することにより，変換処理時の高速化が可能であることから前述のシステム構成とした．

3 変換方式

本システムでは，型推定を伴うパタンマッチングを用いることにより，データ型を導入したパタン変換を高速に処理する。以下では，ルール適用と型推定について説明し，次に型推定を伴うボトムアップなパタンマッチングによるツリー書き換え処理について説明する．

3.1 ツリー書き換えとデータ型

従来からパタンマッチングを用いたツリー書き換えは，コンパイラのコード生成，数式システムのインタプリタの実現などで用いられている．パタンマッチングによる書き換え処理では，ある書き換えの後の次の書き換え対象の選択が処理速度に影響する．このため書き換えの性格を把握することが重要である．

本システムの場合，データ型をパタン変換に導入したことが書き換え処理の特徴となっている．例えば，書き換えにより式の型が決定し，そのため上位レベルでの書き換え処理が発生する場合がある．さらにまた，C言語には型検査の機能が備わっているため，もし書き換えによりツリーがC言語基本型間の演算式に変換される場合にはボトムアップな型検査が必要である．一方，同じパタンであっても位置により異なる書き換えが要求されることがある．例えば代入演算子の左右では式の持つ意味は異なる．従って，ルール適用はトップダウンで行なう必要がある．つまり，本システムにおける書き換え処理では，トップダウンな処理とボトムアップな処理が同様に要求される．また試作システムを用いた実験では，パタンマッチングと型検査の組み合わせ方が変換速度に大きく影響を及ぼ

Overloading in Pattern Transformation for Extending C
Taizo KOJIMA, Akira SUGIMOTO, Shigeru ABE
Mitsubishi Electric Corp.

すという結果が得られた．そして実用システムでは，次に述べる変換方法を採用した．

3.2 型推定を伴うボトムアップなパタンマッチング

変換方法の特徴は，C言語既存の型検査やルール適用により生じるボトムアップな型の決定を型推定として捉え，パタンマッチング時に型推定を同時に行なうことである．そして本システムでは，一パスのボトムアップなトラバースにより，ツリーに対するすべての適用可能ルールの検出を行ない，次にトップダウンなトラバースによりルールを適用し変換結果を出力する方式を採用した．以下に書き換え方式を示す．

- 変換ルールに変換後の型を定義し，型推定に用いる．C言語組み込みの型検査も同一の枠組の中で処理する．
- ボトムアップなトラバースによりパタンマッチングと型推定を同時に行なう．あるノードがあるルールのパタンのルートとマッチングする場合には，そのルールを記録し，定義されている型をそのノードに付加する．複数のルールが適用可能な場合，最も特定的なルールを適用可能ルールとして選択する．なおこの決定は，パタンを構成するノード数，型あるいはノード種別の指定，ルール定義順により行なう．
- トップダウンなトラバースによりルール適用を行なう．ルール適用可能なノードに出会った場合，そのノードをルートとするツリーに対してルールを適用し，そのサブツリーに対するトラバースは行なわない．
- ルール実行は出力操作により実現する．またサブツリーに対するルール適用は，出力操作の中から再帰的に呼び出す．

3.3 例

以下に拡張記述例と変換過程について説明する．次の記述は，2項演算子 '@' の定義と，型 T に対する変換を示すルール例である．まずシンタクス定義を行なう．これは以下のように演算子の種別，優先度を記述する．

```
$operator (left,13) @;
```

一方ルール記述では，以下のように検出すべきパタン ('[..]')，結果の型 ($type)，及びルール適用時のアクション ('{..}') を記述する．以下の例ではアクションとしては 出力操作 emit のみであるが，通常のC言語プログラムを記述できる．

```
$rule r1 [ $(T:p1) @ $(T:p2) ] $type T
   { emit("at2(#a,#a)",p1,p2); }
$rule r2 [ $(T:p1) @ $(T:p2) @ $(T:p3) ] $type T
   { emit("at3(#a,#a,#a)",p1,p2,p3); }
```

図2に上記ルールによる変換処理を示す．第1ステップはボトムアップなパタンマッチングと型推定である．まずノード n3 において r1 がマッチするので，T型と推定する．次に n2 において r1, r2 両方とマッチングする．r1, r2 を比較すると r2 の方が優先する．そこで r2 を記録し，T型と推定する．最後に n1 においても n2 と同様に r1, r2 とマッチングし，r2 を記録し T型と推定する．以上でパタンマッチング処理は終了する．

第2ステップはトップダウンなルール適用過程である．まず n1 に対してルール r2 を適用し，出力操作 emit 命令を実行する．emit 命令の第一引数は書式制御文字であり，後続の引数の処理を示す．書式制御に '#a' が現れると，該当する引数に対して再帰的

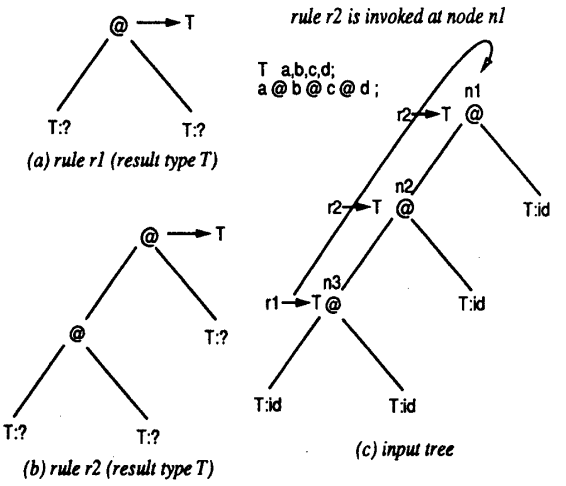

図 2: ボトムアップなパタンマッチング

にルール適用過程が呼び出される．上記例では，パラメタ p1 には n3 をルートとするツリーが束縛されており，このツリーに対してルール r1 が適用される．そして最終的に，'at3(at2(a, b), c, d)' が得られる．

なお上記例において型推定を伴わない場合，まずノード n3 において r1，ノード n2 において r2 がマッチングするので，ノード n2 における変換が優先される．しかしながら，この違いは実用上問題はなかった．

4 おわりに

本稿では，C言語の問題向き拡張システム OPTEC について述べた．本システムでは，データ型をパタン変換に導入し，このパタン変換によりプログラム言語のシンタクスとセマンティクスの問題向き拡張を提供する．本システムの実現では，高速なプログラムパタン変換を達成することを目指した．そして，(1) 変換ルールに付加された型を用い，型推定とパタンマッチングを同時にボトムアップで行なう，(2) ルール適用はC言語プログラムによる出力操作として実現，また (3) 出力操作において，サブツリーに対するルール適用を再帰的に呼び出すように構成した．これにより，1パスのパタンマッチングにより適用可能ルールを決定し，続く1パスの再帰的な出力操作により変換処理を実行する高速な変換処理を実現した．現在のシステムでは，例えば 1000 ステップの入力を1秒以内で変換し，実用に耐える変換速度を達成している．なお本システムを用いて問題向き言語を実装し，実際の応用システムの開発に適用中である．また今後は，本システムで実現したプログラム変換技術と視覚的プログラム技術とを組み合わせ，より生産性の高いプログラム開発支援を検討してゆきたいと考えている．

参考文献

[1] 杉本 他, オブジェクト指向方式によるC言語拡張システム：OPTEC(1,2), 情処 38 全, 1988.
[2] 小島 他, C言語拡張システム：OPTEC, 情処 39 全, 1989.
[3] 小島 他, C言語の問題向き拡張用言語変換システム：OPTEC, 情処 41 全, 1990.