

名前の統括管理システム

5G-3

NameMasterの実装

森 厚・神田陽治・田中英彦
(富士通) (東京大学・工学部)

1. はじめに

名前統括管理システムNameMasterはORAGANシステムにおいて、プログラム中に現われる名前を管理するソフトウェア・コンポーネントである。ソフトウェアの高生産性を実現するには、プログラムの可読性は必要条件であり、NameMasterは、この可読性の向上を目的としている。

本論文では、VAX-11/730のUNIX上で、開発を進めているNameMasterの実装部分について論ずる。

2. データの内部表現

名前は、複合語であり、複合語は、単語から成る。NameMasterとDinnerBellでは、格納効率を考慮し、名前はすべて、内部表現でもちまわっている。内部表現には、Fig.1 に示すワードナンバーとシンボリックナンバーの2種類がある。ワードナンバーはその単語と1対1に対応しており、その単語自身と品詞を表わす番号である。一方、シンボリックナンバーは複合語を表わす番号である。

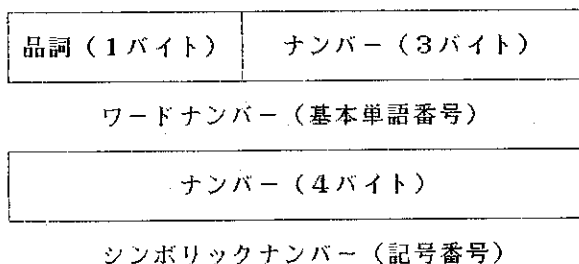


Fig.1 内部表現

3. 構造

NameMasterはコンパイラの一部として動作するため、DinnerBellと密接に情報を交換しあう。Fig.2に、DinnerBellとNameMasterの構造を、Fig.3に表現の変化を示し、データ、制御の流れを両図をあわせて説明する。

P.1では、プログラムから単語を取り出し、frontend部に受け渡す。frontend部では、これをワードナンバーに変換して、P.1に返す。(1)

P.1では、品詞をトークンとして、ワードナンバーと共にPI.yに受け渡す。(2)

PI.yでは、品詞(トークン)を基に名前の構文解析を行なう。構文に誤りがあれば、「シンタッ

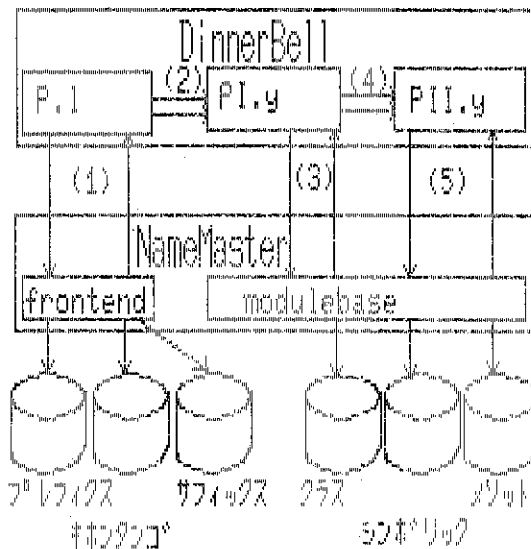


Fig.2 DinnerBell, NameMasterの構造

クス・エラー」として処理し、そうでなければ、1つ、または、複数のワードナンバーからワードナンバー列を生成する。生成されたワードナンバー列は、modulebase部に渡され、シンボリックナンバーに変換される。(3)

PI.yでは、このシンボリックナンバーとトークンをPII.yに受け渡す。(4)

PII.yでは、言語の構文解析を行なうと共に、そこで使われている名前が正しいか、否かをmodulebase部に尋ねる。(5)

modulebase部からの返答が否であれば、「セマンティック・エラー」とする。

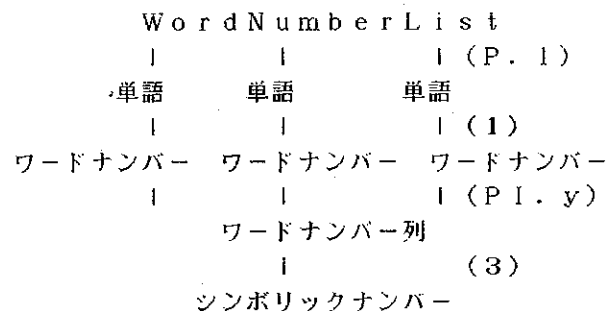


Fig.3 表現の変化

3.1 辞書

NameMasterの辞書は、frontend部で使用する基本

単語辞書、プレフィクス辞書、サフィックス辞書と、ModuleBase部で使用するクラス辞書、メソッド辞書、及び、シンボリック辞書から成っている。基本単語辞書、プレフィクス辞書、サフィックス辞書は、ユーザが管理する。それに対し、クラス辞書、メソッド辞書、シンボリック辞書は、modulebase部が管理し、コンパイル時に動的に更新する。コンパイル過程でエラーを発見すると、modulebase部はリカバリ処理を行い、コンパイル前の状態に戻す。

以下に、個々の辞書の機能と役割を示す。

(1) 基本単語辞書

基本単語とワードナンバーの対応関係を管理する辞書であり、同義語の管理もここで行なう。

(2) プレフィクス辞書

単語の変化に対応するための辞書である。

(3) サフィックス辞書

プレフィクス辞書と同様に、単語の変化に対応するための辞書である。

(4) クラス辞書

名前式からクラス候補を捜し出すのに用いる。クラス候補のシンボリックナンバーの一覧を管理する。

(5) メソッド辞書

プログラムの文脈検査のための辞書であり、システム全体のクラスのメソッドヘッダーを格納している。

(6) シンボリック辞書

ワードナンバー列とシンボリックナンバーの対応辞書である。つまり、従来のコンパイラがもっていた記号表と同じ役割をする。従来の記号表と大きく異なるのは、「記録の対象がシステム全体のプログラムなので、恒久的である。」と、いうことである。

3.2 処理系

DinnerBellのパーザ部はNameMasterとの関係により、従来のコンパイラとは異なった処理系を有する。従来のパーザは、主に字句解析部と構文解析部から構成されていた。NameMasterでは、この字句解析部をさらに、単語解析部(P. 1)と名前解析部(P I. y)に分割し、より深く、名前の検査を行なっている。

NameMasterは、frontend部とModuleBase部から構成されている。

frontend部はDinnerBellの単語解析部から、受け取った単語をワードナンバーに変換する。基本単語辞書と、プレフィクス辞書、及びサフィックス辞書を併用することで、基本単語の変化による辞書の巨大化を防いでいる。

ModuleBase部は大きく分けると、2つの機能をもっている。一つは、ワードナンバー列からシンボリ

ックナンバーを生成し、シンボリック辞書に登録する機能であり、もう一つは、クラスの候補を捜しだし、プログラムの文脈情報に基づいて検査する機能である。

後者を簡単に説明する。まず、スロット名のシンボリックナンバーからクラス辞書のクラス名候補を得る。次に、すべてのスロットのクラス候補の組み合わせに対して、メソッド辞書を調べて、仮想的にメッセージが流れるかどうか調べる。最後に、プログラム全体で、意味のとおりクラス候補の組があるかどうか調べ、ただ一つ残った時に、記号性があると判断する。仮想的なメッセージの流れはFig.4のように調べる。まず、m: C1が流れ出すと仮定し、レシーバーとなっているスロットのメソッドを取り出す。C0のメソッドによって、これがn: C2にかわる。この繰り返しで、最後のレシーバーまでメッセージが流れれば、クラス候補から(a)が残り、メソッドが選ばれ(c)、(b)が形成される。

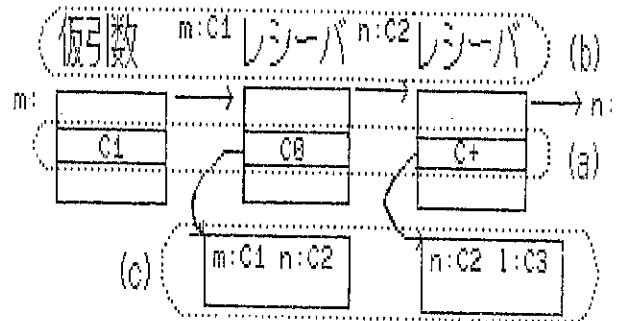


Fig.4 名前の正当性検査

4. 開発環境

DinnerBellとNameMasterはVAX-11/730のUNIX上で、開発を進めている。DinnerBellの単語解析部(P. 1)はLex, 名前解析部(P I. y)、及び構文解析部(P I I. y)は yaccで記述している。また、NameMasterはC言語で記述している。

5. おわりに

NameMasterの場合、実行速度は辞書検索の処理効率に大きく依存するため、そのオーバーヘッドの軽減策は重要な研究テーマである。特に、名前の検査ではORAGAシステム全体のクラス、メソッドを検索するため、その処理時間は多大なものを要する。よって、今後の検討が必要である。

<参考文献>

本大会5G-2