

1L-4

コンパイラ教育支援システムにおける
属性文法に基づく意味解析系提示ツールの作成

梅田雅之

中西健一郎

和田幸一

川口喜三男

名古屋工業大学 電気情報工学科

1. はじめに

コンパイラの学習は、プログラミング言語についてより深く知ろうとするとき欠かせないものであると同時に、その技法については一般性があり、種々の分野のソフトウェア開発に利用することが出来る等、情報系の教育において重要な分野である。しかしながら、このコンパイラの内部動作は複雑多岐であり、その教育はなかなか難しい。現在の教育現場ではOHP、テキスト等の静的資料を用いて教育が行なわれているが、これに加え具体的な動作の様子を視覚的に示す動的資料を提供すれば、コンパイラの教育において極めて有効な手助けとなる。このような考えから、我々はコンパイラ教育支援システム[1]の構築をすすめている。

2. 属性文法に基づく意味解析系提示ツール

今回、コンパイラ教育支援システムの一環として、コンパイラの中の意味解析系に注目し、意味記述の定式化として代表的な属性文法を取り上げ、属性文法に基づく意味解析系の動作を視覚的に解り易く表示するツールを作成する。

属性文法は文脈自由文法の各生成規則に意味規則を付け加えたものである。属性文法にはその評価方法と能力によって様々なクラスが存在するが、現実的には属性文法に幾つかの制限を加え入力を一回のパスで効率の良い属性評価を行う手法が多く用いられている[2]。今回、そのようなクラスの代表として下降型構文解析と同時に意味解析の可能なLL(1)を基底文法とするL属性文法と上昇型構文解析と同時に意味解析の可能なSLR(1)を基底文法とするLR属性文法を対象とした。

このツールでは属性文法で記述された文法と解析するソースを入力とし、与えられた文法に従ってソースを解析しその様子を視覚的に提示する。以下でそれぞれの提示対象とその提示方法について述べる。

LL(1)を基底文法とするL属性文法

LL(1)を基底文法とするL属性文法は構文解析部分に

下降型構文解析系のLL(1)構文解析を用い、意味解析にこれと同時に属性の評価が行なえるL属性文法を用いる。

属性の評価は新たな記号に処理が移ったときにその記号の相続属性を評価し、全ての記号の処理を終えて戻ってきたときにその記号の合成属性を評価する。この様にLL(1)構文解析とL属性文法の評価が同調して実行されていく。この動作を視覚的に解り易く表示することを考えると、提示する対象として以下のようなものが挙げられる。

入力記号列、先読み記号: どのような入力に対して解析を行なっているのか、そして現在入力のどこを解析しているのかという状況を把握するために必要である。入力記号列は与えられた入力そのものをテキストで表示する。先読み記号は入力記号列の中で先読み記号に相当する部分を強調表示することによって示す。

適用された生成規則とその構文図: 構文解析がどの生成規則にそってすすめているのか、これからどの部分を解析しようとしているのかといった構文解析の動作を把握するために必要である。

その時点までに適用された生成規則列: 解析の経過を把握するために必要である。

First集合: LL(1)構文解析系はFirst集合を使って解析を進めるため、構文解析の動作を理解するためにはFirst集合が必要である。

構文解析木: 構文解析の進行の様子を図的にあらわすものであり、これによって構文解析の進行の概観、解析の経過が視覚的に把握できる様になる。構文解析木は自動的に出来る限りコンパクトにレイアウトして表示するが、入力のサイズが大きくなると構文解析木はかなり大きくなり画面に入り切らなくなる。これに対しては画面をスクロールして見ることもできるが、必要のない部分を縮約して出来る限り一度に広い範囲が見られるようになっている。

属性の依存関係: 属性の依存関係は各属性がどの属性から決定されるか(決定されたか)をあらわすものである。この依存関係を図的に表示すれば属性の評価の概観を視覚的に把握することが出来る。依存関係を有向グラフであらわし、属性の評価が構文解析と同調して行なわれることを強調するために、構文解析木に重ね合わせる様にして表示する。

属性の値: 意味解析は属性文法においては属性を評価することに相当する。すなわち、評価された属性の値は意味解析の結果に相当するものであり、意味解析の状況

Tools for displaying semantic parsers based on attribute grammars in a visual-based support system for teaching compiler design

Masayuki UMEDA, Kenichirou NAKANISHI,
Koichi WADA, Kimio KAWAGUCHI
Nagoya Institute of Technology

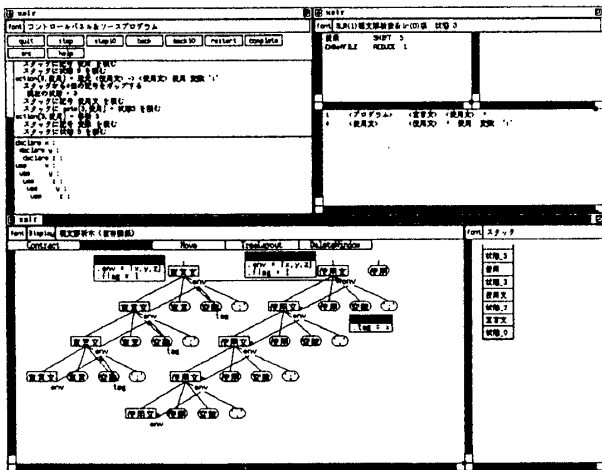


図1 画面図

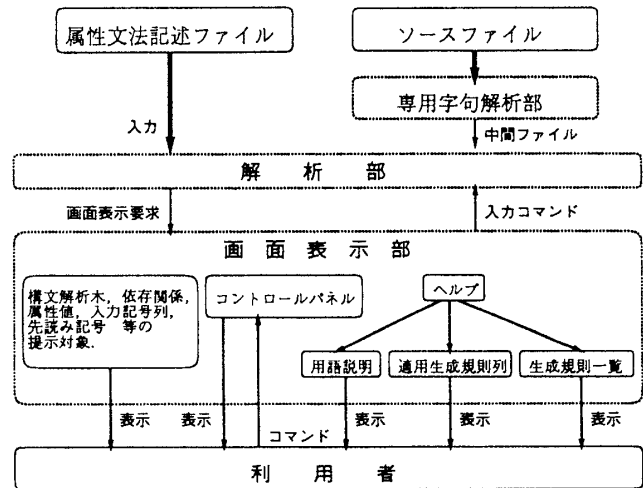


図2 ツールの構成

を把握するためには欠かすことが出来ない。属性の値は各記号毎にまとめて提示する。これも構文解析との関係が解り易いように構文解析木に重ねるようにして表示する。

適用する意味規則: 属性の依存関係と属性の値から属性の評価の概観は掴めるが属性の評価を概観ではなく細かく具体的に把握するために必要である。

SLR(1)を基底文法とするLR属性文法

SLR(1)を基底文法とするLR属性文法は構文解析部分に上昇型構文解析であるSLR(1)構文解析を用い、意味解析部分にこれと同時に属性の評価を行なえるLR属性文法を用いる。

属性の評価は記号がスタックに積まれるときにその記号の相続属性を評価し、還元するときには合成属性の評価を行なう。この様にSLR(1)構文解析とLR属性文法の評価が同調して実行されていく。この動作を視覚的に解り易く表示することを考えると、提示する対象としてLL(1)を基底文法とするL属性文法で挙げた、入力記号、先読み記号、適用生成規則列、構文解析木、属性の依存関係、属性の値に加え以下のようなものが挙げられる。

スタック: SLR(1)を基底文法とするLR属性文法ではスタックに状態、属性を積んで解析をすすめていき、このスタックをどのように使用して解析をすすめているかを把握するために必要である。スタックの実装方法には、数個のスタックを用いる方法が一般的であるが簡単で解り易くするためにスタックは1つとし、1つのスタックに状態と記号(その属性)を積む形式で表示する。

SLR(1)構文解析表: SLR(1)構文解析表の内容、解析系が表のどこを読んでいるのかを把握するために必要である。SLR(1)構文解析表は画面に表示しようとするときかなりの大きさになるので、状態毎に分けて表示する。解析の進行中は解析系の現在の状態を表示し、解析系が読んでいる部分を強調して表示する。

LR(0)項: 各状態に含まれるLR(0)項は、各状態でおこる全ての可能性をあらわしており、現在どのような状態を解析中で、これからどのような入力を期待している

かを把握するために必要である。また、これとSLR(1)構文解析表を見比べることによってSLR(1)構文解析表の意味の概要を把握できる。各状態毎に分けて表示する。解析の進行中は解析系の現在の状態を表示し、可能性のあるLR(0)項を強調して表示する。

適用する生成規則及び意味規則: どの生成規則と意味規則が適用されるかを把握するために必要である。適用する生成規則は強調されているLR(0)項と同じであり、現在の状態のLR(0)項の中で強調することによって表示する。適用する意味規則はLR(0)項の表示と合わせて行なう。それぞれLR(0)の意味規則を表示しその中で適用する意味規則を強調して表示する。

LL(1)を基底文法とするL属性文法、SLR(1)を基底文法とするLR属性文法に基づく意味解析系に対して、今あげた提示対象を画面に表示するツールをXWindow上で作成する。拡張性を考え解析を行なう部分とXWindow上で画面表示を行なう部分を分離し、それぞれを別のプログラムとして実現する。両者間のデータのやりとりはプロセス間通信を用いて行なう。また、自習用に使うことを考え用語等の説明を行なうハイパーカード方式のオンラインヘルプを付加する。このツールの構成を図2に示す。また画面図を図1に示す。

謝辞

本研究の一部は平成4年度内藤科学技術振興財団研究助成金の援助により行なわれた。

参考文献

[1] 中西健一郎, 梅田雅之, 和田幸一, 川口喜三男: アルゴリズムの可視化に基づくコンパイラ教育支援システム, 情報処理学会第48回全国大会, 1994
 [2] 佐々孝政: 属性文法-チュートリアル-, コンピュータソフトウェア, Vol.3, No.4, 1986, pp.377-395