

計算機言語の構造解析と複雑度評価

4Y-9

大内 東、加地 郁夫、 John N. Warfield
(北海道大学工学部)、 (George Mason University)

1. はじめに

コンピュータソフトウェアに関して、生産物や生産プロセスに対する評価や比較などの定量的な尺度がないことが言われている。さらにソフトウェアを利用している人々や、生産にたずさわる人々の認知的負担もほとんど考慮されていないとも言われている。本研究はこれらの不十分な点につき考察し、なんらかの解決策を見出すことを目的としている。

ソフトウェアの開発にたずさわる人々が学び、利用する、もっとも重要なソフトウェアは計算機言語である。コンピュータソフトウェアの保守費が高くつくことの理由の一つに計算機言語の複雑さがある。本稿では計算機言語のシンタックス構造を解析する方法と、その複雑度を評価するための近似的基準を提案する。さらに提案する方法と基準によりALGOL-60, Ada, Pascal, Language C の比較を行なう。

2. 構造解析

Backus-Naur形式(以後BSFと略す)は計算機言語のシンタックスを明確に記述するために利用される。とくにBSFは再帰的定義と基本要素間の定義を明確に表現するのに適している。しかしながら、BSFは、全体構造を把握するためには適していない。これを補うためにグラフィック表現を利用すると良い。

2.1 シンタックス構造グラフ

BSFでは左辺項を右辺項で定義する。従って、以下の様に考えることができる。

仮定1: 学習という立場から見ると、BSFの右辺に現れる項は、左辺に現れる項の必要条件である。

仮定1にもとずき有向グラフを定義し、このグラフをシンタックス構造グラフ(SYN)と呼ぶ。ある項の下位あるいは同レベルにある項はその項の必要条件となっている。

2.2 サイクル構造グラフ

SYNにサイクルが存在する場合は、そのサイクルにあるいずれの項を理解するためには他のすべての項を理解しなければ成らないことを示している。いわゆる”鶏と卵の関係”を表す。次の仮定を置く。

仮定2: サイクルの複雑度はそのサイクルに含まれる要素数の増加に伴い指数関数的に増す。

SYNのサイクルのみに注目し、他の要素はすべて無視して得られる構造グラフをサイクル構造グラフ(CYC)と呼ぶ。CYCのノードはSYNの強連結成分に対応し、アークは強連結成分間の接続関係を表す。ノードの数字はそのサイクルを構成する要素数を表す。CYCはサイクルフリーな階層グラフとなる。

2.3 複雑度

サイクルフリーな階層構造における複雑度に対し、サイクルのある構造の複雑度は指数関数的に増大するということから、言語のシンタックスの複雑度を以下のような考えに基づき近似的に計算する。複雑度の評価基準として、シンタックスのBSFから得られるSYNの中に存在するサイクル構造のみに注目する。すなわち、CYCを用いてシンタックスの複雑度を以下のように定義する。

CYCの第kレベルの複雑度を C_k を $C_k = 2^{n_k - 1}$ で定義する。但し、 n_k は第kレベルのサイクルの要素数である。

仮定3: 全体の複雑度はCYCの最長距離を持つ経路における各レベルの複雑度の積で表される。

全体の複雑度として、仮定3の結果を10を低とする対数で表す。

$$C = \log_{10}(C_1 \times C_2 \times \dots \times C_L)$$

3. 計算例

ALGOL 60, Ada, Pascal, Language C に対して、構造解析と複雑度の評価を行なった。但し、BSFの左辺に表れた項のみを用いている。例としてFig.1 ~

Fig.4 にはこれら4種類の言語のCYCを示す。また、Fig.5 にはAlgol 60のSYNを示す。これらの図から、言語の複雑度は以下のように計算できる。

ALGOL 60:

$$\log_{10} (2^{11} \times 2^1 \times 2^{13}) \sim 8$$

Pascal:

$$\log_{10} (2^{18} \times 2^{11} \times 2^4) \sim 10$$

Ada:

$$\log_{10} (2^{33} \times 2^2 \times 2^5 \times 2^{25}) \sim 20$$

Language C:

$$\log_{10} (2^2 \times 2^7 \times 2^5) \sim 4$$

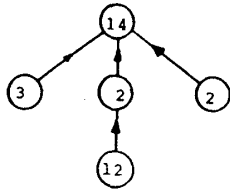


Fig. 1

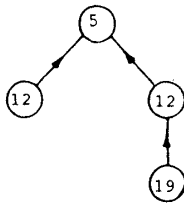


Fig. 2



Fig. 3



Fig. 4

Fig.1 CYC for ALGOL 60.
 Fig.2 CYC for Pascal.
 Fig.3 CYC for Ada.
 Fig.4 CYC for Language C.
 The number in the circle represents the number of elements in the cycle.

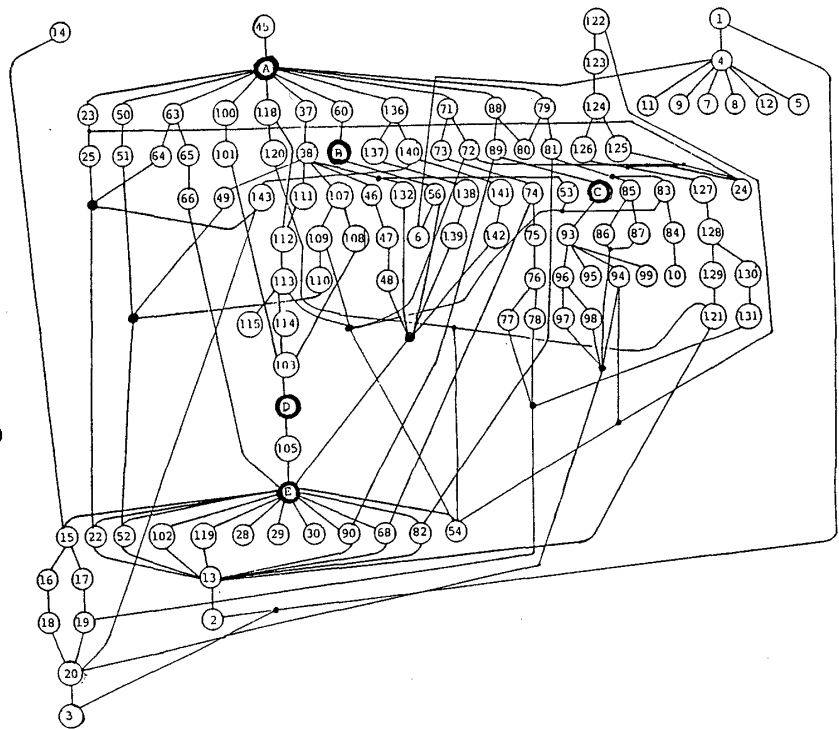


Figure 5. A relation map for the language ALGOL 60. Numbers represent individual terms from this language. Letters represent cycles containing multiple terms. To understand a given term, one must understand all those lying below it that feed into it, as well as all members of a cycle in which it may lie.

4. おわりに

計算機言語のシンタックス構造解析からサイクル構造を抽出し、これを用いて複雑度を定量化した。本文で提案した構造解析の方法と、複雑度の評価基準は計算機言語のみならず一般のソフトウェアに対しても応用できる。しかし、提案した複雑度評価基準は、数多くあると思われる基準の一つである。他の評価基準について考察して行くことが今後の課題である。有益な議論をして頂いた富士通・国際情報社会科学研究所、杉山公造氏に感謝します。

[参考文献]

- [1]D.Thurnau,et.al.,”ALGOL Parogramming,2d Rev.E d.,Detroit: Burroughs,1964.
- [2]J.N.Warfield,”Structural analysis of a computer language”, IEEE The 7-th Southeastern symposium of system theory,1985.
- [3]J.N.Warfield,”A complexity metric for high-level software languages”,1987 IEEE International Conference on SMC,1987.