

言語処理製品検査システム (LPQES)

1M-5

大川真有美, 西条 寛, 塩谷睦男, 福田昌弘  
富士通 (株)

1. はじめに

従来の言語処理プログラムの検査方法は、マニュアルに基づき言語の各機能を細分し、その組合わせによってテストプログラム(TP)をコーディングし、そのTPを実行することによって機能の信頼性を評価していた。しかし、この方法には以下のような問題があげられる。

- TPに対する対象製品の免疫化による、二度目以降のバグ検出数の低下。それに伴う品質評価の精度の低下。
  - テスト対象機能の十分な網羅性確保のための多大な工数確保。
  - 人手によるTP作成ミス の混入、及びテスト項目の偏り
- このような背景の下に、以下に示すような言語処理製品検査システム(LPQES 図1)を、実際の業務に適用して試行中である。

LPQES は以下のステップで対象製品の検査を行う。

- ① 対象製品の言語仕様を定義した属性文法記述風なメタ言語を入力する。(図2-①)
- ② テストプログラムジェネレータ(TPGEN)は上記の定義から、乱数的に生成規則を選択し、TPを生成していく。(図2-②)
- ③ ②で生成されたTPは自己判定機能を持つため、図2-③で示したような簡便な形式で結果を表示できる。

TPGEN に関しては第36回情報処理学会で、『テストプログラム自動生成の試み』と提して報告した。今回の発表はそのTPGEN に対する改善と、実際の業務に適用し得られた成果、及び今後の課題について報告する。

2. TPGEN の問題点と取り組み

2.1 問題点

従来のTPGEN には、実際の業務への適用にあたり、以下の2点の問題があった。

- (1) 当社エキスパートシステム(ES)で属性文法を記述していたため、当社ES, LISP及びTPGEN の内部論理を知らないこと、検査対象言語の定義ができない。このため生産性を阻害しており、かつ読解性も悪い。
- (2) 記述可能な言語機能が制御、代入、宣言など中核機能に限られており、また制限も多く適用範囲が狭い。

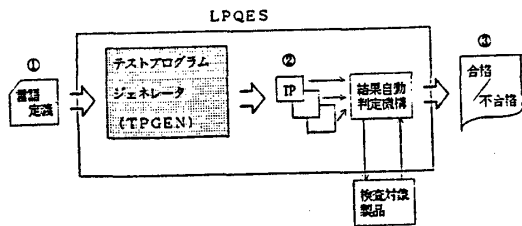


図1 言語処理製品検査システムの概要

<p>① 言語定義例</p> <pre> 文法記述 → (設定 項番値 1) *確認手段の項目               (設定 確認変数 NIL) *確認手段を初期値               &lt;"BNL" 文群 4&gt;* *確認すべきシステム               *プログラムは文群 文群 → 文 (設定 確認変数 NIL)* *文群は文と種と用               *確認変数は、多 文1 → 文% 文2 → 文% 文 → 300 代入文!       【IF(-30) = 100】       IF 文% 代入文 →       【100 : (副作用?)】       変数 "文"式 "BNL"       代入 変数 (値式);         </pre>	<p>② 出力TP例</p> <pre> AAB=294 CALL CHECK(1,AAB,294) AAE=-88--12 CALL CHECK(2,AAE,-76) AAA=-444*189 CALL CHECK(3,AAA,-83) IF AAE:= -263*499 THEN AAB=73*-96 AAA=126/411 ELSE AAD=133/85 CALL CHECK(4,         </pre>	<p>③ 合否判定例</p> <pre> テストプログラム 8本 (194項目) 実行しました。 結果 OK: 194       NG: 0         </pre>
--	--	--

図2 LPQES の入出力結果例

2.2 取り組み

2.1 であげた問題点に対して、我々は以下のように取り組んだ。

(1) 生産性、読解性の向上

日本語記述を導入した読解性の良い属性文法記述用メタ言語を設定し、そのインタプリタを用意した。その結果、従来のTPGEN に比較し記述量が約1/5 に減少した。また属性文法を知る者にとって、記述が自然になり読解性が大幅に向上した。

(2) 適用範囲の拡大

I/O, DB操作のシミュレートも可能になり、通常の言語の殆どの機能を網羅できるまでに至った。適用範囲を拡大するため、今回のTPGEN に導入した工夫の例を以下に示す。

【言語機能の定量制限への対処】

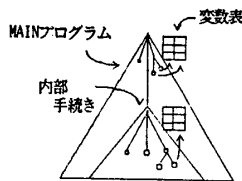
TPGEN では以下のように記述すると、

文→p1 GO TO文 | p2 代入文 | p3 IF 文

GO TO 文、代入文、IF 文のいずれかがそれぞれp1~p3の確率で選択される。しかし、この方法のみではif文が連続して多く選択され、言語の制限(入れ子)を越えることもありうる。今回のTPGEN ではpnの値を動的に変更する機構を導入し、定量制限を越えて選択される確率を0 とすることを可能とした。

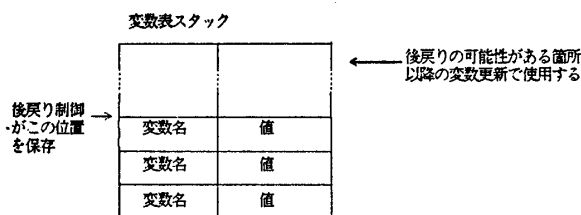
【ブロック内局所変数の実現】

非終端記号の属性の1つに変数表を持たせ、その下方展開中に使用される局所変数の扱いを可能にした。以下に例を示す。



【例外事象発生防止】

ループ演算中の変数オーバーフローなど、生成されたTPの検証中に矛盾が検出されることがある。これに対し上述の変数表をスタック構造にすることにより、例外事象発生時に他の選択で生成を継続する後戻り制御機能を提供している。(下図参照)



3. 成果と考察

88年 6月から10月にかけて検査した言語処理製品の一部にLPQES を適用した。本章ではその結果を従来の人手による検査との比較から考察する。

(1) TP作成工数の削減

従来のTP作成工数の約1/6 となった。図3.1 に工数当たりの実施テスト項目数を示す。TPGEN は原理的には無限個のテストセットからの無作為抽出であり、テスト工数は本テスト項目が多くなるほど大幅に減少する。今回の結果は必要十分なテスト網羅性を得たと判断した時点で、適用を打ち切った場合の数字である。したがってこの結果は通常の検査作業における、平均的な工数削減量と考える。

(2) 工数当たりのバグ検出数

LPQES の適用による単位当たりのバグ検出数は、従来の手法に比較し1.8 倍向上した。

従来のテスト手法で検出される障害は、TPのデバッグ過程での障害が多い。これに対し、TPGEN では言語定義のデバッグ過程では異常系のTP (エラーを含む) は作成されるが、検出障害の多くは正常系である。(図3.2)

この結果、ユーザへの影響の大きい正常系の障害は、従来手法と比べ大幅に減少させた後に出荷可能となる。また(1)、(2) で述べた通り、大幅な工数削減により、検査担当はより高度の判断を必要とする使い易さ、性能などの評価に専念できた。このため従来の工数範囲内で、より緻密な品質評価が可能となった。

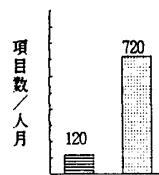


図 3.1 工数当たりの実施項目数

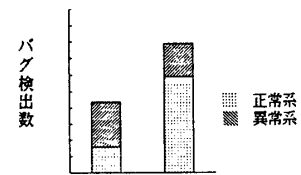


図 3.2 単位当たりのバグ検出

4. 今後の課題

3で述べた通り、工数削減、品質評価の精度の向上などのLPQES の効果について、実際の業務で検証することができた。しかし、これまで述べてきた工数削減や適用範囲の拡大については、まだ改善の余地は多いと考えている。今後は以下の点について取り組んでいく予定である。

性能の向上、MMI の改良による一層の生産性の向上より広範囲な製品への適用による、TPGEN の有効性の検証

参考文献

1)塩谷, 西条: テストプログラム自動生成の試み, 第36回全国大会講演論文集(II), pp.1067-1068 (1988).