

# 構造化されたチャートのプログラムサイズの最小化に関する研究

2T-9

佐藤智昭 大原茂之  
東海大学

## 1 はじめに

プログラムの構造化はプログラムの開発と保守の効率を高めることに寄与してきた。その反面、プログラムの構造化は、プログラムを冗長にすることがある。この欠点は、例えばメモリ容量が制限されているシステム開発<sup>1), 2)</sup>の場合には馴染まない。そこでプログラムの開発保守においては構造化プログラミングの利点を生かし、システムに実装する段階でプログラムの冗長性をなくすために構造化を系統的に崩してプログラムのサイズを最小化することが考えられる。ここではプログラムの構造化にTSチャート<sup>3)</sup>を用いる場合について、最小化する手法について述べる。

## 2 本文

### 2.1 書き換え規則

ここではTSチャートを最小化する規則を与える。

【規則1】 図1.1の構造を満たすTSチャートを図1.2の構造に書き換える。

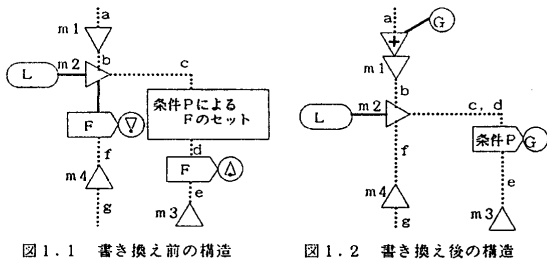


図1.1 書き換え前の構造 図1.2 書き換え後の構造

(実行順序等価の証明) 以下に図1.1と図1.2の実行順序が等価であることを示す。

図1.1 → 図1.2

[ ] 内の数字は一連の手続きの処理順序を示すものであり、また [ ] 内の " ." は被制御構造であることを示す。

(図1.1)

- [1] a, m1, b
- [2] Lによるループ
- [2.1] m2, c, (条件Pが真ならばFのセット), d
- [2.2] Fがセット状態ならばm3に分岐, m1に分岐
- [2.3] Fがセット状態でないならばe, m3
- [3] f, m4, g

(図1.2)

- [1] a, m1, b
- [2] Lによるループ
- [2.1] m2, c, d
- [2.2] 条件Pが真ならばGOTO合流記号に分岐, m1
- [2.3] 条件Pが偽ならばe, m3
- [3] f, m4, g

図1.2 → 図1.1についても同様に示すことができる。

【規則2】 図2.1の構造を満たすTSチャートを図2.2の構造に書き換える。

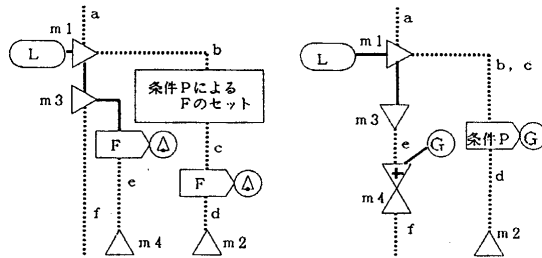


図2.1 書き換え前の構造 図2.2 書き換え後の構造

(実行順序等価の証明) 以下に図2.1と図2.2の実行順序が等価であることを示す。

図2.1 → 図2.2

(図2.1)

- [1] a
- [2] Lによるループ
- [2.1] m1, b, (条件PによるFのセット), c
- [2.2] Fがセット状態ならばm2に分岐, m3, m4に分岐
- [2.3] Fがセット状態でないならばd, m2
- [3] Fがセット状態でないならばm3, e, m4
- [4] f

(図2.2)

- [1] a
- [2] Lによるループ
- [2.1] m1, b, c
- [2.2] 条件Pが真ならばGOTO合流記号に分岐, m4
- [2.3] 条件Pが偽ならばd, m2
- [3] 条件Pが偽ならばm3, e, m4
- [4] f

図2.2 → 図2.1についても同様に示すことができる。

【規則3】 図3.1の構造を満たすTSチャートを図3.2の構造に書き換える。

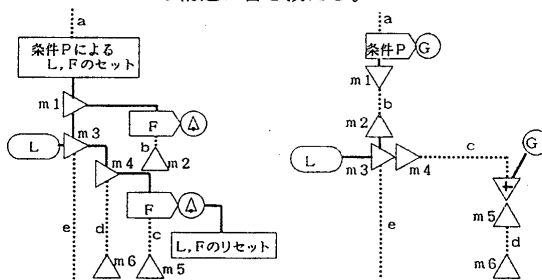


図3.1 書き換え前の構造 図3.2 書き換え後の構造

(実行順序等価の証明) 以下に図3.1と図3.2の実行順序が等価であることを示す。

図3.1→図3.2

(図3.1)

- [1] a, (条件Pが真ならばL・Fのセット)
- [2] m1, Fがセット状態ならばm2に分岐, (ループ)
- [2.1] (m3, m4, L・Fのリセット), m5, d, m6
- [3] m1, Fがセット状態でないならばb, m2
- [4] Lによるループ
- [4.1] m3, m4, c, m5, d, m6
- [5] e

(図3.2)

- [1] a
- [2] 条件Pが真ならばGOTO合流記号に分岐
- [2.1] m5, d, m6
- [3] 条件Pが偽ならばm1, b, m2
- [4] Lによるループ
- [4.1] m3, m4, c, m5, d, m6
- [5] e

図3.2→図3.1についても同様に示すことができる。

【規則4】 図4.1の構造を満たすTSチャートを図4.2の構造に書き換える。

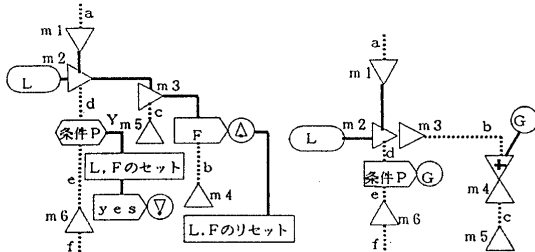


図4.1 書き換え前の構造 図4.2 書き換え後の構造

(実行順序等価の証明) 以下に図4.1と図4.2の実行順序が等価であることを示す。

図4.1→図4.2

(図4.1)

- [1] a, m1
- [2] Lによるループ
- [2.1] m2, m3
- [2.2] Fがセット状態ならば(L・Fのリセット), m4, c, m5
- [2.3] Fがセット状態でないならばb, m4, c, m5
- [3] d
- [4] 条件Pが真ならば(L・Fのセット), m1に分岐
- [5] 条件Pが偽ならばe, m6, f

(図4.2)

- [1] a, m1
- [2] Lによるループ
- [2.1] m2, m3
- [2.3] b, m4, c, m5
- [3] d
- [4] [2.2] 条件Pが真ならばGOTO合流記号に分岐, m4, c, m5
- [5] 条件Pが偽ならばe, m6, f

図4.2→図4.1についても同様に示すことができる。

## 2.2 書き換え規則の適用

ここではこの書き換え規則を用いてTSチャートを最小化する例を示す。

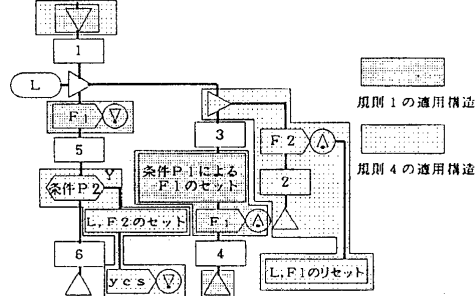


図5.1 書き換え前の構造

図5.1のTSチャートにおける濃いドットの網掛けの構造に規則1を、そして薄いドットの網掛けの構造に規則4を適用する。その結果を図5.2のTSチャートに示した。

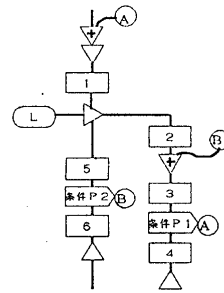


図5.2 書き換え後の構造

この例では書き換え規則を適用することによって分岐または選択の処理は図5.1の六つから図5.2では二つに減少した。また、図5.1のフラグ・ループ条件のセット・リセットの処理は、図5.2においてないことがわかる。

## 3 おわりに

本報告ではTSチャートの最小化手法について述べた。しかし、今回の内容にはまだ多くの制限があるため今後はこれらの制限を緩めることと、さらに機械的に最小化していく手法について検討したい。

## 謝辞

本研究を進めるに当たり、日頃お世話になっている本学電子工学科主任飯田昌盛教授、制御工学科主任小高明夫教授に感謝の意を表します。

## 参考文献

- 1) 佐藤、水本、大原、小高：ワンチップマイコンの初期化に関する一考察、情報処理学会第44回全国大会(1992)
- 2) 水本、岩男、佐藤、大原、小高、宇賀神：プログラマブル周辺デバイス応用支援システム、情報処理学会研究報告(1991)
- 3) 大原茂之：木構造化チャートによるプログラム開発・保守、情報処理学会論文誌Vol.27 No.10(1989)