

## 3R-1

## 日本語による論理仕様記述の試み

浅尾 清, 大平 駿介, 伊藤 誠, 蓼沼 良一

山梨大学

## 1. はじめに

我々の研究室では論理設計の知的支援システムの開発を進めている。その知的支援システムの構築の一部として、日本語による仕様書からの回路合成の試みについて報告する。

論理回路の自動合成システムは多数発表されているが、設計仕様の記述は人工言語であり、そのシステム特有の表現形式である。仕様を日本語で記述できれば、誰にも理解できる仕様書が作成できると考えられる。また、人工言語では記述上のあいまいさは生じないが、自然言語ではあいまいさが生じ易く、それを補うための常識(知識)が必要になってくる。

## 2. ブロック図レベルの回路記述

仕様記述の段階では、データフローレベルのブロック図で記述する。ブロック図の各要素は並列実行可能であり、各要素はRTL、構造的機能素子、または、日本語で記述される。RTLの関数、機能素子、日本語の動詞に関する知識は、外部のフレーム型DBより会話型で提供される。

以下、主として日本語記述のトランスレータについて述べていく。

## 3. 日本語記述からの素子生成システムの概要

システムは、日本語解析部、展開合成部、シミュレータ、説明合成部、知識ベースの5つにより構成される。知識ベースマネージャー以外はすべてPrologで記述した。

日本語文解析部は仕様書を入力として、仕様書の文の意味を表した動作記述を出力する。

展開合成部は、動作記述から回路記述を作り出す。

シミュレータ部は、イベントドリブンの多値シミュレータをProlog上に実現したものである。

説明合成部は、回路記述からその回路の動作

の説明文を出力する。

展開合成、説明合成のルールは知識ベースマネージャーにより管理される。

## 4. 仕様書の記述と日本語文解析部

仕様書の記述の一般形は次のとおりである。

- ・回路名称
- ・入出力信号の記述
- ・素子、信号の記述
- ・処理記述

処理記述の一般形は、

(状態)で(条件)のとき

(動作)して(状態)へいく

という形である。合成の記述レベルは、まだ知識ベースが不十分であるため、低レベルである。また、現在の段階では、Prologのシステム上の制約から、仕様書はローマ字の分かち書きになっている。

解析部はDCGを用いて記述し、トランスレータによりPrologプログラムに変換した。解析結果は動作記述として、fact形式のリストとしてassertされる。

例として乗算器の仕様書を以下に示す。

名称 mul

[i13, i12, i11, i10]と[i23, i22, i21, i20]は  
入力である。

laは出力である。

uaとlaはシフトレジスタである。

bはレジスタである。

cはカウンタである。

qはD型フリップフロップである。

siはuaのシリアルインプットである。

cyoはキャリーである。

t1でuaをクリアしてt2へいく。

t1でlaに[i13, i12, i11, i10]を保存してt2へいく。

t1でcをクリアしてt2へいく。  
 t1でbに [i23, i22, i21, i20] をセットしてt2へいく。  
 t2でla0が1のときuaとbを加算してuaとし  
 t3へいく。  
 t2でla0が1のときqにcyoをセットしてt3へ  
 いく。  
 qかつla0のときsiとする。  
 t3でuaとlaを右シフトしてt4へいく。  
 t3でcをインクリメントしてt4へいく。  
 t4でc2が1でないならt2へいく。

### 5. 展開合成部

展開合成部は、動作記述を回路記述へ展開する。fact形式でassertされている動作記述のリストを取り出し、univにより複合項とし、それをゴールとして実行する（展開ルールの適用）ことにより動作記述を書き換えていく。これを、書き換えが不可能になるまで繰り返し行うことによって、展開合成を行う。展開合成のルールは、トップダウンになっている。フリップフロップやレジスタのように、1つの動作記述から全てのデータ・制御情報を取り出せないものについては、展開合成時に各種信号をassertしておいて、書き換え不能になったところでassertしてある各種信号を取り出し、回路記述を合成するようにした。乗算器の仕様書からの合成結果を示す。

```
circuit :-
    and2(q, la0, si),
    and2(la0, t2, ct12),
    inv(c2, d1),
    and2(d1, t4, ct11),
    add([ua3, ua2, ua1, ua0], [b3, b2, b1,
        b0], [cyo, d23, d22, d21, d20]),
    inv(ct12, s1),
    or2(t3, ct12, cp1),
    inv(t1, bc11),
    shift(si, [d23, d22, d21, d20], s1,
        bc11, cp1, [ua3, ua2, ua1, ua0]),
    or2(t3, t1, cp2),
    shift(ua0, [i13, i12, i11, i10], bc11,
        hi, cp2, [la3, la2, la1, la0]),
    reg([i23, i22, i21, i20], hi, t1,
        [b3, b2, b1, b0]),
    cnt(bc11, t3, [c3, c2, c1, c0], cntmax),
```

```
dff(cyo, ct12, hi, c1, q, bql).
```

### 6. シミュレータ部

より柔軟な論理シミュレータの開発を目的として、Prologによるシミュレータを試作した。このシミュレータは、各信号の現在の値とこれから変化する信号をassertによって登録することにより、イベントドリブンを実現した。また、エッジトリガを表現するために0, 1, up, down, unknownの5値を扱う多値シミュレータになっている。

### 7. 説明合成部

説明合成部は、入力信号より出力信号まで各素子間の配線をたどりながら、素子単位に用意された説明文（形式は仕様記述と同様で、外部DBより提供される）をテンプレートに用いて、回路全体の説明文を合成する。説明のレベル（詳細さ）は指定できる。

### 8. おわりに

論理回路設計のような限定された世界では、自然言語処理もある程度は可能であることが確認できた。しかし、本システムの自然言語処理は、分かち書きされた特定の記述形式を前提としており、完全な自然言語処理にはなっていない。また、制約のある日本語記述は人工言語よりあいまいさが多くなることが予想されるが、本システムは使用できる表現が限られているので、入力用のテンプレートを用意しておけば入力が楽になるとともに、あいまいさも減少すると思われる。また、記述の抽象度を上げて入力や展開合成時に、知識の検索や推論によるデフォルト処理すれば知的支援システムにより近づいていくと考えられる。

#### 【参考文献】

- [1] W. F. Clocksin, C. S. Mellish (中村克彦訳): "Prologプログラミング", マイクロソフトウェア, 1983
- [2] 溝口文雄 他: "Prologとその応用2", 総研出版, 1985
- [3] 渡辺茂 他: "CAD用データベースシステムの開発", 情報処理学会第33回全国大会論文集, pp889-890, 1986-10