自然語仕様から代数的仕様への，文脈を考慮した変換 －ＯＳＩセション層プロトコル仕様を例にして－

3L-3

島袋 潤　萩本 和彦　関 浩之　藤井 護　嵩 忠雄
大阪大学基礎工学部情報工学科

1．まえがき

我々は，自然語によって記述されたプログラム仕様を代数的仕様に変換する変換法の検討と，その方法を用いて変換を行うシステムの作成を行っている[5]．本方法では，自然語の各文（あるいは段落）から‘論理式 == true’の形の公理への変換規則を定義する．また，自然語文に現れる個々の語句の意味は，抽象データ型の述語や関数として，変換規則とは別に公理によって定義する．変換規則によって得られる公理，語句の意味を定義する公理，および論理式の従う論理体系を定義する公理の和集合からなる代数的仕様を，もとの自然語仕様に対する代数的仕様とする．

変換規則は，構文規則，中間表現構成規則および公理生成規則から成る．まず，GPSG[2]を用いて記述された構文規則に従って自然語文を構文解析し，構文木を得る．次に，構文木を，HPSG[3]の語彙構造に似た中間表現と呼ばれる形式に変換する（中間表現構成規則）．中間表現には，論理式への変換に際して必要となる情報のうち，例えば名詞句に対応するデータ型や，動詞句に対応する述語やその引数のデータ型に関する制約条件等が格納されている（詳しくは，[5]）．なお，$\Gamma=\gamma_1\cdots\gamma_n$を自然語文の系列，各$\gamma_i$ $(1\leq i\leq n)$に対応する中間表現をA_iとするとき，中間表現の系列$A_1\cdots A_n$を，Γに対応する中間表現の系列と呼ぶ．最後に，自然語文の系列（段落）に対応する中間表現の系列に対し，局所的な構文からの情報のみでは扱えない問題についての決定を行うことにより，もとの自然語文に対応した公理を得る（公理生成規則）．

現在，例題として，ＯＳＩセション層プロトコル仕様[1]（以下，原仕様と呼ぶ）の中のSPM（セションプロトコルマシン，セション層の通信主体）の送受信動作を規定している部分（７章，約250文）を変換の対象とし，変換規則の精密化と，そこに現れる語句の意味定義を進めている[6]．この仕様を選んだ理由は，構文の曖昧さが比較的少ないこと，仕様としての規模は大きいが，用いられている構文，語句の種類が比較的少ないこと，専門的な分野であるため，語句の意味が定義しやすいこと等である．以下，2.では公理生成規則について，原仕様の変換に際して考慮した点を述べる．次に3.で原仕様に現れる語句の意味定義について述べ，4.で変換システムの概要について述べる．

2．公理生成規則

中間表現から公理への変換に際しては，(1)指示代名詞の先行詞の決定，(2)限量子の有効範囲の決定，(3)自然語文では陽に指定されていない引数の指定が必要である．

(1)については，指示代名詞と先行詞の候補のデータ型を比較することによって先行詞を決定する．これによって一意に決定できない場合には，ユーザに問い返して決定する．今回対象とした約250文では，一箇所以外はデータ型を用いて一意に決定できた．先行詞を一意に決定できなかったのは形容詞的用法で用いられているthisである．これに対しては，時制を表す語句がもつ，事象の前後関係に対する制約条件を考慮して先行詞を一意に決定できる．

(2)について，構文により決定できない場合には問い返すが，今回対象とした部分では，曖昧さの問題はなかった．

(3)については，原仕様中の例を用いて説明する(表1)．この一連の文は，REFUSE SPDUというデータ単位を送信するときのSPMの動作を記述したものである．このうち，例えば文(d)は，文(c)で述べられている，タイマの起動という事象があって初めて意味をなす．このことから，各自然語文から変換規則によって得られる論理式は，その文が前提としている系の“状態”（状況と呼ぶ）を引数としてもつように公理生成規則を定義する．この引数は，自然語文では陽に指定されていない引数である．形式的には，データの送受信や変数の値の更新等のような単一の事象を表す式の有限系列を値にとるデータ型として“事象の系列”型を導入する．そして，この引数のデータ型は“事象の系列”型であると定義する．上述のように，文(d)の直前の状況は，その一つ前の文(c)の直後の状況である．一方，文(e)の直前の状況は，文(d)の直後の状況ではなく，文(c)の直後の状況である．そこで，自然語文の系列Γに対応する中間表現の系列$\alpha=A_1\cdots A_n$に対して，Γの各文の接続関係（各文の直前の状況が，それ以前のどの文の直後の状況となり得るか）を組織的に求めるために，次のような有向アサイクリックグラフ（αの依存グラフと呼ぶ）と述語を用いる．なお，自然語文γの直前（直後）の状況を，γに対する中間表現の直前（直後）の状況とも呼ぶことにする．

・依存グラフ：各頂点はαの中間表現と一対一に対応し，Aの直後の状況がBの直前の状況の一つであるときかつそのときのみ，頂点AからBへの枝が存在する．各頂点の子供には，全順序（第1子，第2子，…）がついている．

・relate(A,s,t)：Aの直前の状況をsとしたとき，tがAの直後の状況となり得るときかつそのときのみtrueとなる述語．

・precond(A,s)：状況sが，Aの直前の状況に対して仮定している条件（条件文の条件節等に対応）を満たしているときかつそのときのみtrueとなる述語．

これら二つの述語は，自然語文の各語句に対応する関数

表１　原仕様中に現れる自然語文の例

(a) An [S-CONNECT(reject)response] [results in] a [REFUSE SPDU].
(b) The SPM [waits for] a [CONNECT SPDU] if the [Transport Disconnect parameter] indicates that the [transport connection] can be reused.
(c) Otherwise the SPM starts the timer,TIM and [waits for] a [T-DISCONNECT indication].
(d) The SPM requests [transport disconnection] with a [T-DISCONNECT request] if the timer expires before receipt of a [T-DISCONNECT indication].
(e) The timer is cancelled on receipt of a [T-DISCONNECT indication].

※[α]は，αを一語とみなすことを表す．

A Translation from Natural Language Specifications into Algebraic Specifications Using Contextual Dependencies
Jun Shimabukuro, Kazuhiko Hagimoto, Hiroyuki Seki, Mamoru Fujii and Tadao Kasami
Osaka University

や述語毎に形式的に定義する．例えば，表1の文(e)中の語句 on，cancelledに対応する述語on(A,a,s)，cancelled(tm,s)については，

relate(on(A,a),s,t) ≙ ok(sa) ∧ relate(A,sa,t)
precond(on(A,a),s) ≙ ok(sa) ∧ precond(A,sa)
relate(cancelled(tm,*),s,t) ≙ t = s cancelled'(tm)
precond(cancelled(tm,*),s) ≙ true

（Aは主節に対する中間表現，aはonの目的語の句が表す事象，canselled'(tm)は“タイマtmのキャンセル”という単一の事象，*は状況を表す引数の位置を表す）

ここで，ok(s)は，状況sがプロトコルで許された事象の系列であるときのみtrueとなる述語である（4.参照）．

また，relateの定義を中間表現の系列に対する述語に拡張して，次の述語を定義する．

$relate^{+}(A_1 \cdots A_{i-1}A_i, s, t) \triangleq$
$\exists u[relate^{+}(A_1 \cdots A_{i-1}, s, u) \wedge relate(A_i, u, t)]$

原仕様の７章に対して，文の接続関係についての分析を行った結果，依存グラフは以下の性質をもつと考える．依存グラフの根から頂点A_iへの頂点列を$A_1 \cdots A_i$とする．

［性質１］ 依存グラフを子頂点間の全順序関係に従って先行順走査によるリスティング（preorder listing）をすると，もとの自然語文の系列が得られる．ただし，入射枝を複数もつ頂点は最後の枝で走査するものとする．また，(a)入射枝を一つもち，出射枝をもたない頂点は，入射枝とともに消去する，(b)入射枝，出射枝を共に一つもつ頂点は，一つの枝で置き換える，(c)二頂点間に複数の枝が存在すれば，それらを一つの枝で置き換える，の三つの操作を有限回繰り返すと，単一の頂点のみからなるグラフが得られる．（直観的には，後方の文が前方の文の系列に割り込むような接続関係はなく，入れ子構造をもつことを表している．）

［性質２］ 頂点A_i($1 \leq i \leq n$)から頂点$B_1, \cdots, B_m$への枝が存在し，A_1の直前の状況をs_0としたとき，

$relate^{+}(A_1 \cdots A_i, s_0, t) \supset$
$\{precond(B_1, t) \vee \cdots \vee precond(B_m, t)\} \equiv true$

が成り立つならば，A_iはそれ以外の頂点への出射枝をもたない．（枝分かれによる場合分けが，既に全ての場合を尽くしているならば，それ以上の枝分かれはない．）ここで，≡は，もとの自然語仕様に対する代数的仕様によって定まる合同関係[4]を表す．

［性質３］ 頂点A_iからBへの枝が存在するならば，

$relate^{+}(A_1 \cdots A_i, s_0, t) \wedge precond(B, t) \not\equiv false$

が成り立つ．（A_iの直後の状況とBの直前の状況に対する条件が矛盾しない．）

［性質４］ さらに枝を追加すると，性質１～３の少なくとも一つが成り立たなくなる． □

次に，依存グラフまたはその部分グラフGとその直前の状況sから論理式への関数transを次のように定義する．Gの根をA，Aの第i子($1 \leq i \leq n$)を根とする部分グラフをG_iとし，G_iの直前の状況を表す変数をtとする．

trans(G,s) ≙ trans'(A,s) ∧ [relate(A,s,t) ⊃
{trans(G_1,t) ∧ … ∧ trans(G_n,t)}]

（C ∧ Dは，Cの表す論理式，文字'∧'，Dの表す論理式をこの順に連接して得られる論理式を表す．他も同様．）

ここで，trans'(A,s)は中間表現Aに対応する論理式を与える関数で，relateと同様に語句毎に定義されており，先に述べた自然語文で陽に指定されていない引数の位置に状況を表す引数を代入する役割を果している．

以上をまとめると，自然語文の系列（段落）Γに対応する公理は，

trans(G(α),s) == true

である．ここでαはΓに対応する中間表現の系列，G(α)はαの依存グラフ，sはΓの先頭の自然語文の直前の状況を表す変数である．上で述べた性質２および３では，もとの自然語仕様に対する代数的仕様で定まる合同関係を用いて依存グラフの満たすべき条件を述べている．従って，変換規則は，その変換結果である代数的仕様に依存する条件を用いて定義されているという意味で再帰的に定義されている．ただし，原仕様では多くの場合，性質２および３が成り立つかどうかを，もとの自然語文の構文のみに依存する条件を用いて判定することができる．例えば，性質２において，A_1 = if(X,Y)，A_2 = otherwise(X,Z) ならば

precond(A_1,t) ∨ precond(A_2,t) ≡ true

が成り立つ．

３．語句の意味定義

原仕様に現れる語句の意味の定義のために，述語ok(s)を導入する．この述語が，「事象の系列sがプロトコルによって許されたものであるときのみtrueである」という意味をもつように，関数や述語に対応する語句の意味を定義する公理を与える．例えば，表1の文(a)は，「S-CONNECT(reject)responseを受信すると，REFUSE SPDUを送信する」ということを述べている．この中のresult inに対しては，

result_in(d1,d2,s) ==
ok(s in(d1)) ⊃ [ok(s in(d1) out(d2)) ∧
{x≠out(d2) ⊃ ¬ok(s in(d1) x)}]

ここで，xは単一の事象，d1,d2はデータ，sは事象の系列を表す変数であり，in(d1),out(d1)はそれぞれデータd1を受信，送信するという事象を表す式である．この公理は，「result_in(d1,d2,s)が成り立つ」とは，「もし，ok(s in(d1))がtrueならば，ok(s in(d1) out(d2))もtrueであり，かつ，out(d2)以外の任意の事象xに対して，ok(s in(d1) x)はfalseである」ことであると定義している．

４．変換システム

我々は既に，自然語文を公理に変換するプロトタイプシステムの試作を行ってきた．構文解析部は，GPSGによる構文規則を書き下したDCGの規則から得られるPrologの構文解析プログラムを用いている．中間表現構成部はPrologで実現され，公理生成部はその機能の一部がＣ言語で実現されている．今回，変換の対象とした部分（原仕様の７章）の分析を通して，構文解析部，中間表現構成部の拡張および精密化を行った．現在，７章のうち約160文に対しては作業が終了しており，構文解析部はDCGレベルで約170規則，中間表現構成部はPrologの節数が約180個の規模である．なお，今回追加した構文規則は約50規則（DCGレベル）であるが，そのほとんどが，約160文の約1/3についての作業を終えるまでに必要となったものである．また，原仕様の７章の残りの部分（約90文）については，構文規則の追加を行うことなく，辞書項目の追加のみで中間表現への変換が可能である．このことから，実用規模の仕様においても，構文規則の追加が比較的容易に収束することがわかった．また，公理生成部については，依存グラフの構成と語句の意味定義を組み込む作業を進めている．

謝辞 有益なご討論およびご助言を頂いた，本学基礎工学部伊藤実講師ならびに並河英二（現・野村総研），松村享の両氏に深謝致します．

文 献

[1] ISO: “Basic Connection Oriented Session Protocol Specification”, ISO 8327.
[2] G.Gazdar, et al.: “Generalized Phrase Structure Grammar”, Basil Blackwell (1985).
[3] C.J.Pollard: “Lectures on HPSG”, unpublished manuscript, Stanford University (1985-02).
[4] 嵩他，信学論(D)，J69-D,7, pp.1066-1074 (1986-07).
[5] 並河他，情処学研報，NL-65-5 (1988-03).
[6] 萩本他，信学技報，COMP88-56 (1988-11).