研究ノート

応用人工知能
—自動プログラミング・エキスパート—

豊　田　順　一*

近頃，人工知能，とりわけエキスパートシステムに関する記事が新聞に載らぬ日はない．どの会社も急遽 AI の研究グループを組織して，あわてふためき，セミナー屋の主催するセミナーに若手を出席させたりしている．大学の先生方も先生方である．僅かな金，多くても高々5万円の金で少々の知識を切り売りする．言ってみれば売春婦と変らないような事をやっている若手もいる．

少し苦言を呈させて頂いたが，自戒の念を込めて書かせて頂いた．

私共の部門では，AI の研究のうち自然言語理解の応用として，自然言語で書かれた仕様からプログラムを生成する研究を行っている．自然言語自身の持つ問題と共に，仕様が手続き的あるいは宣言的に記述されるかによって種々の問題が生ずる．仕様を手続き的に書けば，制御構造を書き込む必要があり読みづらい．それを宣言的に書けば，制御構造を考える必要のない代わりに，読み手の常識を仮定して多くの書くべき事項が省略され，二通り以上の解釈が可能となる．したがって，プログラム合成に際しては，自然言語記述からユーザの意図を抽出し，ユーザの生きている世界に関する様々な常識，知識を利用することにより，ユーザの考えている処理の筋道を補足してやる必要がある．ここでは，プランとゴールの概念によって自動プログラミングを行う．

ゴールは行為の目的を示すものであり，プランはゴールを達成するための計画を表す．例えば，「おなかがへった」という発話に対して，ゴールは「空腹を満たす」ことであり，そのゴールを達成するプランとして「レストランに行く」とか「ホカホカ弁当を食べる」がある．プログラミングでは，ゴールは仕様に書かれているはずの意図（要求），プランはその意図を達成するための手続き（プログラム）に対応している．したがって仕様からゴールを抽出し，ゴールを達成するためのプランを推論することができれば，自然言語からプログラムが合成されたことになる．

このためには，以下のようなゴール・プランに基づく手順が必要である．

①まず，文法，辞書などの「自然言語に関する知識」を用いて，入力仕様文から文の意味表現を抽出する．このためには，我々が開発した Integrated Parser を用いる．この IP は統語・意味・文脈処理のすべてを融合したパーサである．

②得られた意味表現からユーザの意図を推論する．この推論には，意味表現とゴールの対からなる「ゴール推論知識」を用いる．

③次にゴールを達成するためのプランを推論し，得られたプランから更にプリミティヴなプラン推論するというように段階的にプランを詳細化，実行可能なプラン列に変換する．この過程では「プランニングのための知識」を用いる．図1にプラン詳細化の概念を示す．

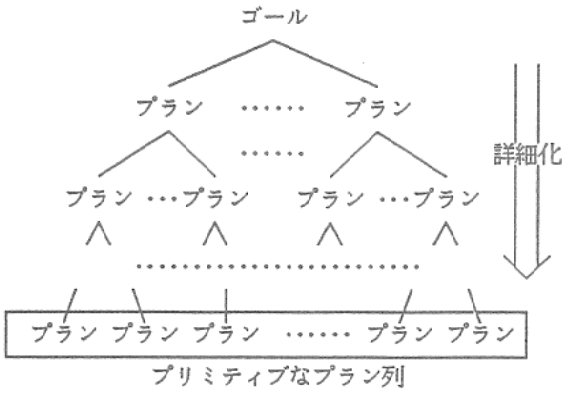
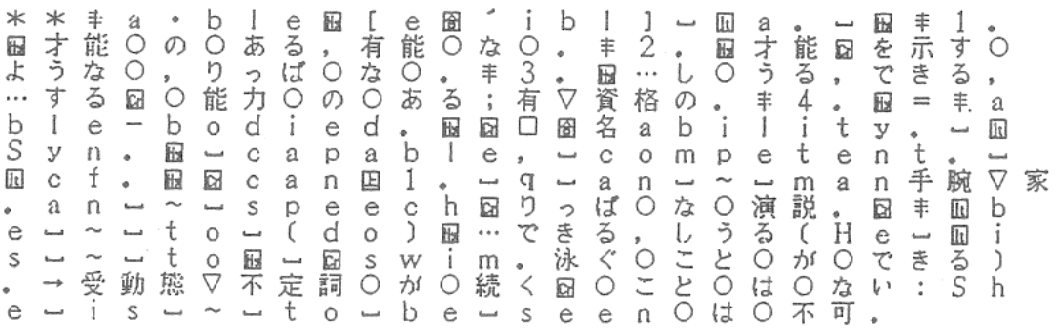


図1　プランの詳細化

*豊田順一（Jun'ichi TOYODA），大阪大学産業科学研究所，豊田研究室，教授，工学博士，人工知能



(a) 磁気テープ上の辞書データ

＊＊a・ble[éibl] *a*. 1. 才能のある，有能な．2. 才能を示すような，りっぱな．3. …しうる，できる，…する能力のある；有資格の．4. =able-bodied.
図ability. *Syn.* capable, competent. *cf.* can[1]. q an ~ man手腕家. an ~ speech りっぱな演説.
be ~ *to* (do)…できる，しうる(He *is* ~ *to* swim. 泳ぐことができる)．→受動態不定詞が続くことはない：She is ~ to be seenは不可.

(b) 辞書イメージのデータ

図2 入力データと希望する出力データ

```
仕様文：
    辞書項目 中 に ある 最初 の ボールド体 は 見出し語 を 示す 。

Prolog プログラム：
    program(辞書項目(_42587),_50510,_50511) :-
        not exist_um(flag(ボールド体),_50510),
        pick_up(_42587,[キ,_41418,■],_41653),
        put_um(flag(ボールド体),_50510,_50511),
        delete_um(辞書項目(_42587),_50511,_50566),
        put_um(辞書項目(_41653),_50566,_50567),
        make_pointer(_41027),
        put_um(top=見出し語(_41418,_41027),_50567,_50568),
        goal(_50568,_50511).


仕様文：
    見出し語 の 後 に あり 鍵括弧 で 囲まれた 記号 は 発音 を 示す 。

Prolog プログラム：
    program(辞書項目(_55334),_65534,_65535) :-
        get_um(_53404=見出し語(_53274,_53407),_65534),
        pick_up(_55334,[[,_55083,]],_55579),
        delete_um(辞書項目(_55334),_65534,_65574),
        put_um(辞書項目(_55579),_65574,_65575),
        make_pointer(_54662),
        put_um(_53407=発音(_55083,_54662),_65575,_65576),
        goal(_65576,_65535).


仕様文：
    辞書項目 の 先頭 に ある 改行 は 無視する 。

Prolog プログラム：
    program(辞書項目(_17564),_21268,_21269) :-
        head_on(_17564,[■]),
        pick_up(_17564,[■],_17760),
        delete_um(辞書項目(_17564),_21268,_21304),
        put_um(辞書項目(_17760),_21304,_21305),
        goal(_21305,_21269).
```

図3 プログラム合成例

```
見出し語 = a b l e
    解説 = 受動態不定詞が続くことはない：S h e  i s  a b l e  t o  b e  s e e nは不可
    品詞 = 形容詞
        成句 = b e  a b l e  t o  (d o)
            用例 = H e  i s  a b l e  t o  s w i m.
                日本語訳 = 泳ぐことができる
            日本語訳 = しうる
            日本語訳 = …できる
        用例 = a n  a b l e  s p e e c h
            日本語訳 = りっぱな演説
        用例 = a n  a b l e  m a n
            日本語訳 = 手腕家
        語義 = =a b l e-b o d i e d
            区分 = 4
        語義 = できる
            区分 = 3
        語義 = …する能力のある
            区分 = 3
        語義 = …しうる
            区分 = 3
        語義 = 有資格の
            区分 = 3
        語義 = 才能を示すような
            区分 = 2
        語義 = りっぱな
            区分 = 2
        語義 = 才能のある
            区分 = 1
        語義 = 有能な
            区分 = 1
        発音 = e i b l
            第一アクセント = 1
        参考語 = c a n
        類義語 = c a p a b l e
        類義語 = c o m p e t e n t
    頻度 = 最も高い
    名詞形 = a b i l i t y
```

図4　構造化データ

④また，各プランとその前後のプランでデータがうまく受け渡しされるかどうかを確認するために，プランの実行に必要な「前提条件」と，プランの実行後に得られると予想される「処理結果」を用いる．あるプランの予想される処理結果が，後続のプランの前提条件を満たさない場合，前提条件を達成するために必要なプラン列を新たに生成し，後続のプランの直前に付加する．さらに，仕様中に対象の省略が含まれる場合，プランニング過程で随時常識や文脈情報を参照しながら補足する．

⑤以上の処理を，既に得たプランに対して繰り返し適用し，最もプリミティヴで省略を含まないプラン列にまで詳細化する．

⑥最後に，個々のプリミティヴなプランをそれぞれ対応する目標言語の命令系列に置き換えて，実行可能なプログラムを生成する．この過程では，命令やフォーマットなどの「目標言語に関する知識」と，プログラムの「入出力形式を規定した知識」を用いる．

この考え方を三省堂出版「新コンサイス英和辞典」の磁気テープを原データとして，それを構造化するプログラムを合成した例を示す．ここでいう構造化処理とは，辞書のイメージがそのまま入力されたデータから括弧や制御用コードを手掛かりにして，見出し語，発音などの各データ項目を切り離し，見出し語を頂点とする木構造を再編成する処理のことである．

図2，図3に具体例を示した．

このプログラムを走らせた結果の一例を図4に示す．

本研究ノートでは，日本語文仕様よりPrologプログラムを合成するシステムについて述べた．

ソフトウェアの高次自動合成は今後の課題でもあり，自然言語およびソフトウェア工学の両面から研究を推し進める必要があろう．