自由文による思考プログラミング

中村 圭介[1,a)]

概要：Prolog は分野に依存せず論理的な知識をプログラムとして表現できる点で「汎用」ではあるが，1) 知識を自由文（各国自然言語の全体を含む）では定義できない，2) 学際的な問題に対応し難い，ことがより実用的な汎用性への課題である．そこで，自然言語表現，関連知識の動的ネット引用，ローカルな価値観やタブーの定義等をより簡易に扱う自由文思考プログラミング言語「中学＄ Prolog」を開発した．単一化は，より利用者に分かり易い，自由文に変数を埋め込んだ「言葉方程式」とのパターンマッチングとする．また, 問題毎の文脈管理やプログラム間の動的な遷移・引用機能も実装し, より汎用的な QA/RPG サービス用に自由文や推論規則を活用する為の統合実行環境「HUMANOTE」を開発した．これらの仕組みにより，より実用的な汎用性を目指す論理型ＡＩへのアプローチを提案する．

キーワード：Prolog, 汎用性, 自然言語, 単一化, 知識の動的引用, 価値観, タブー, 文脈管理

1. はじめに

Prolog は分野に依存せず論理的な知識をプログラムとして表現できる点で「汎用」ではあるが，以下の２点が, より実用的な汎用性を獲得するための課題ではないかと考えている．

- 知識を自由文（各国自然言語の全体を含む）では定義できない
- 文法や語彙（オントロジー依存）の統一が困難であり学際的な問題に対応し難い

そこで，Prolog の基本的な考え方（特に, 述語論理において構成要件を定義するホーン節（horn clause））を参考にしながらも, 自然言語表現，関連知識の動的ネット引用，ローカルな価値観やタブーの定義等を一般中学生以上がより簡易に扱えることを目指し，自由文思考プログラミング言語「中学＄ Prolog」を開発した**?**.

特に, 単一化 (unification) は，利用者に, より分かり易くなるよう，自由文に変数を埋め込んだ文字列（以降,「言葉方程式」と呼ぶ）と, 事実 (fact) 又は規則ヘッド (rule head. これも「言葉方程式」であってよい）とのパターンマッチングとした．

また, 利用者の問題状況毎の文脈管理や各問題状況を解決するプログラム間の動的な遷移・引用機能も実装し, より汎用的なＱＡサービス／ＲＰＧサービス用にネット上の自由文や推論規則を活用する為の統合実行環境「HUMANOTE」を開発し, オープンベータ版[*1] を配信している．

以降では，代表的な６つのデモ用サンプルプログラムの意味や動作を説明し，これらの動作原理を組み合せた，より実用的な汎用性を持ちうる論理型ＡＩへのアプローチを提案する．

2. 例題１：言葉方程式と構成要件

例題 1 としてプログラム（./1.jpl）上の簡単な

[1] ナレルシステム株式会社
[a)] keisukebecome@icloud.com

[*1] github.com/keisukebecome/humanote_for_public/

「推移律」の定義と質問（実行）を例に説明する.

```
///中学$Prologプログラム（./1.jpl）
9: アイスクリームは冷たい
10: 冷たいは涼しくなる
11: $Aは$B :- $Aは$C ; $Cは$B ;
12: #例題1 :- アイスクリームは$X ;

///利用者からの質問（「#例題1」起動）
?アイスクリームは$X

///HUMANOTEからの回答
:$X = 冷たい
  ?アイスクリームは$X
    9 アイスクリームは冷たい

:$X = 涼しくなる
  ?アイスクリームは$X
    [11] $Aは$B :- $Aは$C ; $Cは$B ;
      -
      9 アイスクリームは冷たい
      -
      10 冷たいは涼しくなる
      =
探索時間 = 466msec
```

9 行目は, 一般的な日本語（自然言語. ただし, わざと句点を省いている）である. 10 行目は, 文法的にはむしろ「冷たいものは」とすべき「標準的でない」自然言語であるが, 当処理系では「現場の略文」として「標準的でない」ことをエラーとせずに解釈する.

11 行目は, 助詞「は」以外のひらがな「は」が多用されていない短文集合に対し, ある程度有用な「推移律」の定義を, （何も常識を知らない）論理プログラミング解釈系「HUMANOTE」に与える推論規則である. 「:-」の左辺（「$A は$B」）は, その「$A」と「$B」が埋まった場合の結論を示す. 一方右辺は,「$A は$C」と「$C は$B」をＡＮＤ結合した連立言葉方程式であり, 結論「$A は$B」であるための構成要件を定義したものである.

なお、Prolog と異なり条件の間をコンマ「,」ではなくセミコロン「;」で区切っているのは, コンマ「,」をセミコロン「;」よりも多用する英語等に対応するためでもあり, 手続的解釈におけるコマンド順的な意味合いを示すためでもある.

連立言葉方程式として AND 結合された 2 つの言葉方程式の同一の変数（「$C」ならば同じ「$C」）には, 数学の連立方程式と同様に, 一つの解については同一の言葉（数でもよい）が代入されなければならない.

「利用者からの質問」の意味は, 「言葉方程式『アイスクリームは$X』の変数$X を根拠をもって埋める（束縛する, 代入する）ことができる文字列を全て求めなさい」となる.

「HUMANOTE からの回答」は, 2 つの解「$X = 冷たい」「$X = 涼しくなる」を示しており, それぞれの下に回答の元となった言葉方程式「?アイスクリームは$X」と, それぞれの解の「根拠ツリー」が示される. なお, 根拠ツリーの各行先頭の番号はプログラムの行番を示し,「[11]」は 11 行目が推論規則であり, 事実（fact）ではないことを示す.

3. 例題２：ネット知識等の動的引用Ａ（大域スコープ LinkedOpenRule）

ついで, 質問の実行時（すなわち「動的」に）, 元プログラム（./2.jpl）中に, 別プログラム（例題 1 の./1.jpl）の内容を読み込んで相互に影響させて回答する動的引用機能を説明する.

```
///中学$Prologプログラム（./2.jpl）
9: ///@1.jpl
10: 好きなものはアイスクリーム
11: 私が食べているものは$A
       :- 好きなものは$X ; $Xは$A ;
12: 例題2 :# 私が$X ;

///利用者からの質問（「例題2」起動）
?私が$X
```

↓質問実行時に 9 行目を 1.jpl に展開して回答

```
 9: ///◆INC_START 2018/11/29_23:05:18
1.jpl
・・・略
17: アイスクリームは冷たい
18: 冷たいは涼しくなる
19: $Aは$B :- $Aは$C ; $Cは$B ;
20: #例題1 :- アイスクリームは$X ;
21: ///◆INC_END  1.jpl
```

```
22: 好きなものはアイスクリーム
23: 私が食べているものは$A
        :- 好きなものは$X ; $Xは$A ;
24: 例題２ :# 私が$X ;

///HUMANOTEからの回答（根拠ツリーは略）
:$X = 食べているものは涼しくなる
:$X = 食べているものは冷たい
探索時間 = 4007msec
```

ここでは, 例題１の内容 (./1.jpl) が例題２の内容 (./2.jpl) の９行目以降に読み込まれた上で推論（質問「私が$X」の探索）が行われる.

なお, この例題２の実行では, 無限ループの制御と高速化キャッシュ?を用いているため, 根拠ツリーは省略している.

この例では, 知識を含んだプログラム（1.jpl）を, HUMANOTE がインストールされたローカル PC のカレントフォルダ (./) から読み込んでいるが, 「@https://hoge.net/hoge/1.jpl」, 「@http://192.168.1.1/hoge/1.jpl」等と, インターネット URL や社内 LAN 上の共有パス等を指定することにより, そのインターネットや LAN 等からも他社や仲間等が随時更新した最新バージョン（ニュース, 最新法則, 等）の知識を含んだプログラムを決まった時間にダウンロードして各社/各自の推論規則を適用した結果を音声合成したりすることもできる.

また, プログラム「1.jpl」がさらに別のプログラムを引用している場合, そのファイルもプログラム「1.jpl」から再帰的にインラインで引用される. これにより, 引用する知識の保守を信頼する編集者に任せることも容易になる.

なお, 本稿では詳述しないが, 動的引用 (URL やパス指定を含む）が推論規則内の条件式の位置で宣言され, 引用が有効であるスコープをその宣言以降の「ボディ内のみ」とする動的引用も実装しており (?を参照)」, このタイプの動的引用には, 以下のような目的がある.

- 大域的な引用過多による組合せ爆発の回避
- 引用プログラムの事実や規則ヘッドへの識別文字列の付加による無用な単一化の回避
- 引用した内容の主語や前提等の明確化（とりわけ, 個人 PC →組織 LAN →インターネットへと公開されたプログラム側に引用する場合）

この具体例と詳細はマニュアル[*2]を参照されたい.

4. 例題３：再帰的三段論法によるアイディアの合成

以下の例題３では, 既知の出発点と交通手段と到着点の組をいくつか与えられた場合に, 「交通手段を組み合わせて（この例では）豊中からどのような旅が可能か？」を自動探索するものである. この例では, 交通手段系列の再帰的な合成を行っているが, 例えば製造業においては, 出発点をワーク（材料）の加工前状態とし, 交通手段を加工方法（工程）とし, 到着点をワークの加工後状態とした同様の自由文テキスト集合を集めることにより工程系列の合成を行うこともできる. さらに他の分野においても関連する系列やより複雑な構造の再帰的な合成も可能であると考えている.

```
///中学$Prologプログラム（./3.jpl）
 9: 金沢から車で能登に行ける
10: 大阪からＪＲ北陸線で金沢に行ける
11: 大阪から新幹線で仙台に行ける
12: 仙台から徒歩でぎゅうタン屋に行ける
13: 豊中から阪急宝塚線で大阪に行ける
14: 能登から船でへぐら島に行ける
15: $Xから$Yと$Zで$Dに行ける
      :- $Xから$Yで$Cに行ける ;
         $Cから$Zで$Dに行ける ;
16: 例題３ :# 豊中から$X ;

///利用者からの質問（「例題3」起動）
?豊中から$X

///HUMANOTEからの回答（根拠ツリーは略）
:$X = 阪急宝塚線で大阪に行ける

:$X = 阪急宝塚線とＪＲ北陸線で金沢に
          行ける

:$X = 阪急宝塚線と新幹線で仙台に行ける
探索時間 = 6258msec
```

[*2] knowrel.com/jd_prolog/html/howto.html

5. 例題４：一般的な価値観定義によるアイディア合成の制限

コンピュータが知識を組み合わせて再帰的に解を合成することにより, 人間的な価値観や科学的知見等からなる「常識」を逸脱した利用不能なアイディアを生み出してしまうことを防ぐため, それぞれの立場での価値観や物事の程度をテキストで定義し運用できるようにしている.

```
///中学$Prologプログラム（./4.jpl）
 9: <%一般的に大事
10: 家族の結婚式
11: 親友の結婚式
12: 赤ちゃんの誕生日
13: ペットの誕生日
14: >%一般的に大事
15: $Xは$Yより一般的に大事
      :> %一般的に大事 ; $X>$Y ;
16: $Xがあるので旅行を楽しむ
  :- $Xは親友の結婚式より一般的に大事 ;
17: 例題４ :# $X旅行を楽しむ ;

///利用者からの質問（「例題4」起動）
?$X旅行を楽しむ

///HUMANOTEからの回答（根拠ツリーは略）
:$X  = 家族の結婚式があるので
  ?$X旅行を楽しむ
    [16] $Xがあるので旅行を楽しむ
  :- $Xは親友の結婚式より一般的に大事 ;
      -
      [15] $Xは$Yより一般的に大事
              :> %一般的に大事 ; $X>$Y ;
        -
        9 <%一般的に大事
        -
        10 家族の結婚式
        -
        11 親友の結婚式
        =
      =
探索時間 = 278msec
```

この例題４では, 休暇をとって旅行をするために必要な理由を,「親友の結婚式」よりも重要なものに限って探索している. ここでは,「家族の結婚式」がそれにあたるが,「一般的に重要」という項目を「当社において重要」等という項目に置き換えて内容を再定義することにより各国・各社ごとに変わりうる価値観を反映した探索を行うことができる.

なお, 例示はしていないが価値や程度が同じものは「息子の卒業式; 娘の卒業式」のように「;」（セミコロン）で区切って「同列」として定義できるようにしている.

6. 例題５：科学的・道徳的なタブー定義によるアイディア合成の制限

各現場の価値観への配慮だけでなく, 科学的・道徳的なタブー（これも現場により変わりうる）を避けたアイディア合成や解探索が人間と同様にできないと, 実用的な汎用性は得られない.

「中学$Prolog」では, 下の 10 行目の例のように, 構成要件に「(変数による）幅」をもたせた形の「強い否定」によりゴールやサブゴールの成立を監視し次のようなものを否定することができるようにしている?.

- 永久機関の合成等, 科学的なタブーの合成
- ペットをいじめる案等, 道徳的なタブーの合成
- 物語の世界観等と矛盾する命題等の合成

```
///中学$Prologプログラム（./5.jpl）
10: !$Xに$Yを食べさせて儲ける
      :- $Xは家畜 ; $Yは毒 ;
11: $Xを使った$Yの廃棄物処理方法
      :- $Xに$Yを食べさせて儲ける ;
12: $Xに$Yを食べさせて儲ける
      :- $Xは$Yを食べる ;
13: ヤギは家畜
14: ヒ素のついた紙は毒
15: ヤギはナタネ油のついた紙を食べる
16: ヤギはヒ素のついた紙を食べる
17: 例題５ :# $X方法 ;

///利用者からの質問（「例題5」起動）
?$X方法

///HUMANOTEからの回答（根拠ツリーは略）
:$X  = ヤギを使ったナタネ油のついた紙の
         廃棄物処理
```

```
?$X方法
  [11] $Xを使った$Yの廃棄物処理方法
    :- $Xに$Yを食べさせて儲ける ;
    -
    [12] $Xに$Yを食べさせて儲ける
          :- $Xは$Yを食べる ;
      -
      15 ヤギはナタネ油のついた紙を
           食べる
      =
    =
探索時間 = 344msec
```

この例では, 利用者は新しい方法のアイディアをビジネス目的で探索している.

ヤギが（14 行目及び 15 行目）により, ナタネ油のついた紙もヒ素のついた紙も食べることを前提に質問に対する再帰的な自動推論は進む.

その結果,「ヤギにヒ素のついた紙を食べさせて儲ける」というアイディア A と「ヤギにナタネ油のついた紙を食べさせて儲ける」というアイディア B の 2 つが内部的には生成され,「HUMANOTE」のメニューの「否定」オプションが ON の場合には, 10 行目の$X に「ヤギ」が, $Y に「ヒ素のついた紙」が代入されたとき, タブー (!○○）に該当する構成要件の自動検査によりアイディア A が否定され, タブー構成要件として否定されなかったアイディア B のみを生かした推論が継続する結果, 上記のような解を返すこととなる.

7. 例題６：みんなで徐々につなげて広げるＲＰＧ（LinkedOpenRPG）

このような定義能力と動作原理を活用しながら, Prolog と同様の手続的解釈も実行できるようにして, インターネット上のプログラムを移動しながら利用者の文脈（RPG の場合は状態にあたる）を再帰的に更新することにより, 複数の作者が RPG の舞台, 世界観, ゲーム選択肢等を徐々につなげて広げられる Consumer Generated Medea 型の簡単な RPG デモを作成した.

これは, ある RPG 部分 X をプレイすることによって定義された文脈 (ヒットポイントや所持金等の状態 (自由文形式)) を, 別の制作者による RPG 部分 Y のプレイに持ち越せるだけでなく, X の制作者が制作した手続的解釈ルール（各利用者文脈を更新する能力もある）を Y のプレイ場面にも一部持ち越して実行することができるようにしたものである.

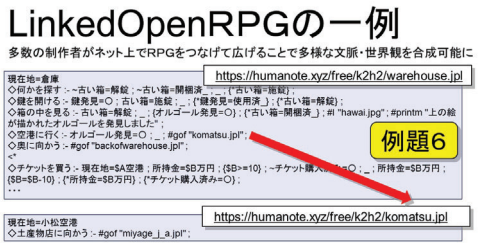

図 1 LinkedOpenRPG のソースの静的リンクの例

この LinkedOpenRPG のサンプルデモは,「HUMANOTE」の前記オープンベータ版において, インターネット接続した後, 中央下の「音声認識」チェックボックスをチェックし, 入力欄から「#gof ”https://humanote.xyz/free/k2h2/warehouse.jpl”」と入力して「設定」メニューから「PLAY 専用」を選択することにより試すことができる.

8. 論理型汎用 AI へのアプローチ

こうした道具立てにより, 自由文という柔軟な文脈の操作と複数の制作者の知見による世界観の合成を可能にし, 例えば図 2 の形式のような, より実用的な論理型汎用 AI の実現に近づいていけるものと考えている.

図 2 中の「主 PG（プログラム）の動的切替」のために参照する内容として, 現状は以下のものを実装しているが, カメラ画像や利用者音声以外の音にも対応していきたいと考えている.

- 静的リンクへの遷移を伴う選択肢のマウス等による選択
- 利用者の音声（コマンド）の認識
- キーボードによる文脈（自由文）の主張

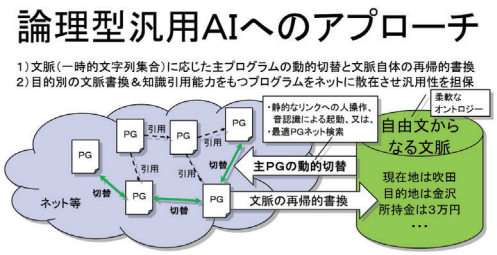

図 2　論理型汎用 AI への我々のアプローチ

9. 今後の課題

また, 今後は, より高い実用性と汎用性（あらゆる分野の知識の選択と合成）に向け, 図 2 における主プログラム（PG) の切替と文脈の再帰的書換（単純追加も含む）をより効果的に行うための研究開発を行っていきたいと考えている. さらに, 小学生から高齢者まで幅広い層への普及を目指し, 使いやすさの改良を行っていきたいと考えている.

謝辞　これまでの開発にあたり, ナレルシステム株式会社の安藤建剛氏, 木戸口卓矢氏並びに株式会社丸誠商会の川畑貴寛氏他の皆様方から多大なご支援を頂きました. また展示会・学会等で問題点等をご指摘・ご指導頂きました知識科学分野の諸先輩方に, この場をお借りして改めて感謝と御礼を申し上げます.

参考文献

[1] 中村圭介: 特開 **2014-211725** 多長一致の無境界単一化法を含む基本特許. 公開公報, 2014.
[2] 中村圭介: 特開 **2017-091119** ネット最新知識の動的引用と引用スコープ. 公開公報, 2017.
[3] 中村圭介: 特開 **2017-102628** 価値観や観点ごとの程度の定義と利用. 公開公報, 2017.
[4] 中村圭介: 特開 **2017-111411** 自由文思考プログラミングによる勉強・研究方法. 公開公報, 2017.
[5] 中村圭介: 特開 **2017-130070** コンピュータからの逆質問による判断. 公開公報, 2017.
[6] 中村圭介: 特開 **2017-211736** 選択式によるユーザインターフェース改善. 公開公報, 2017.
[7] 中村圭介: 特開 **2018-032275** キャッシュによる探索高速化. 公開公報, 2018.
[8] 中村圭介: 特開 **2018-060298** 言葉方程式を用いた自由文の利用方法等. 公開公報, 2018.
[9] 中村圭介: 特開 **2018-063509** 難読文字等の音声の合成又は認識方法. 公開公報, 2018.
[10] 中村圭介: 特開 **2018-072784** 主プログラム遷移による音声認識範囲の制限. 公開公報, 2018.
[11] 中村圭介: 特開 **2018-073015** 自由文の目視による探索枝刈. 公開公報, 2018.
[12] 中村圭介: 特開 **2018-081598** 自由文思考プログラミング環境での広告方法. 公開公報, 2018.
[13] 中村圭介: 特開 **2018-085002** 自由文思考プログラミング環境を用いた通信方法. 公開公報, 2018.
[14] 中村圭介: 特開 **2018-190180** 自由文によるホーン節を用いた前向き推論. 公開公報, 2018.
[15] 中村圭介: 特開 **2018-190182** 科学的・道徳的なタブーによる強い否定. 公開公報, 2018.
[16] 中村圭介: 特開 **2018-190184** 自由文によるアイディアの自動合成. 公開公報, 2018.