単一化に基づく、ゲームのための言語処理インタフェース

Unification Based Language Processing Interface for Computer Games

松原康夫†，何　　航‡
Yasuo　Matsubara†　and　Ka Kou‡

Abstract: Game is a system, which interacts with players through various interfaces. So far, physical interfaces, like a lever, a button and so on, are mainly used. As a possibility, the interaction through a natural language may extend the scope of games.

However, we could not find out any convenient tool to actualize this plan. Then, we have constructed, as an experiment, a tool which processes a statement based upon a LFG-like rules and produces a functional structure which can be used to progress the game. As an application, we realized a world of blocks.

１．はじめに

コンピュータゲームは、様々なインタフェースを通して人間とコンピュータが相互作用するものであるが、これまではレバーやボタンなどの物理的な操作による相互作用が主なものであって、自然言語による相互作用はあまり用いられてこなかった。しかしながら、自然言語による相互作用はゲームの概念を大きく広げる可能性を持つ。

現代の言語処理の研究においては、できるだけ多くの場面で通用する方法を追及し、できるだけ多くの語彙を処理できることを目的としている。しかしながらゲームにおいては限定された文脈と意味論に基づいて処理をすることが求められる一方で、その場での意味の処理をすることが求められる。こうした場合に、限られた語彙と意味論、あるいは限られた文法に基づいて、プレイヤーが入力した文を処理するような簡便なシステムは見当たらない。

本研究では、そうした限定された語彙、限定された文法、そして限定された意味論に基づいて、入力文を処理するシステムを構築した。本システムはかなり限定された種類の文を処理できることを目的としている一方で、その文の意味についてはゲームの場面に応じた処理ができることを必要とする。

こうした目的に合致させるために、CKY 法によって、入力文の形態素解析と構文解析を平行して行い、構成された構文木から、機能的注釈によって F 構造(functional structure)を構成するという、LFG に類似した方法を採用することとした。こうした方法を採ることで、句構造としては若干の制約を受けるにしても、様々な文法を必要に応じて試してみるという柔軟性を得ることができた。

一つの応用として、積み木の世界を構築してみたことで、ゲーム一般への応用の可能性を確認することができた。

†文教大学情報学研究科教授　　‡文教大学情報学研究科修士課程２年

２．木構造の構築

ゲームの各場面においての応用を考えると、形態素解析と構文解析を行い、その場面で必要とされる「意味」を取り出すことが重要である。そのためには LFG のような単一化文法を応用することが一つの方法として考えられる。しかし必ずしも LFG の体系に忠実であることは必要ではない。

また限られた語彙と文法を簡単に処理できることが求められることから CKY 法を利用することが発想される。本システムでは、次のような形の文法を用いることとする。

〔句構造規則〕

（１） 右辺がちょうど二つの非終端記号からなる規則（分岐規則と呼ぶ）

A → B C

（２）右辺がちょうど一つの非終端記号からなる規則（保存規則と呼ぶ）

A → B

（３） 右辺が終端記号である辞書規則（この規則の左辺の記号を前終端記号と呼ぶ）

A → α □

CKY 法のアルゴリズムはよく知られている。通常は n 個の形態素列からなる入力文を処理するために n×n の三角行列を用いる。t[i,j]には、i 番目の形態素から始まる長さ j の形態素列を導出できる、あらゆる非終端記号を格納するのであった。本システムにおいては長さが n 文字の入力文に対して、n×n の三角行列を用いる。そして t[i,j]には i 番目の文字から始まる長さが j の文字列を導出できる、あらゆる非終端記号を格納するのである。

入力文は 1 次元配列 s[n]に格納されて提示される。s の添え字は 0 から n-1 までである。s[0]が初めの文字を保持し、s[n-1]が最後の文字を保持する。三角行列の要素 t[i,j]において i は 0≦i≦n-1 の値をとり、j は 0≦j≦ n-i-1 の値をとる。

CKY 法はボトムアップの過程とトップダウンの過程からなる。ボトムアップの過程を終えたときに要素 t[0,n-1]に、文記号 S が存在すれば構文解析は成功である。

通常の CKY 法では、最初に一番下の段に形態素を導出する前終端記号を登録することから開始するのであるが、本システムの方法では、辞書規則の右辺が、入力文中の部分文字列に一致するすべての前終端記号を、必ずしも一番下の段ではなく、しかるべき高さの位置に登録する必要がある。つまり以下のようにする。

i 番目から始まる長さ j の文字列 s[i],s[i+1],・・・s[i+j-1]が、 辞書規則 「 A → α 」の右辺 α と一致するとき、 前終端記号 A を行列要素 t[i,j]に登録する。

ここで、入力文中にある文字列は互いに文字列が重なるものも含めて、すべての前終端記号を登録することが必要である。これによって、可能なすべての形態素を登録することになる。

このようにして、可能なすべての形態素を登録し、保存規則の処理を行った後は、通常の CKY 法におけるのとまったく同様に、一番下の段からボトムアップに行列要素に非終端記号を登録してゆく。その結果 t[0,n-1]に文記号 S が登録されていれば、構文解析は成功である。

トップダウンの過程は、t[0,n-1]に登録された文記号 S をルートとして、木構造を構成するものであり、通常の CKY 法と同じである。

３．F 構造の取り出し

LFG におけるのと同様に、本研究における句構造規則には機能的な注釈(functional annotation)が付随する。実際に積み木の世界を構築するときの例を用いて記述法を示す。次の文を解析することを考える。

入力文:「赤いピラミッドをブロックに載せよ」

この文を解析するために必要な最小限の文法規則を図 1 に示す。

この文法に基づいて入力文を解析すると、図 2 のような解析木が得られる。

S → VP[↑=↓];
VP → JP[↑OBJ1=↓,↓CASE=を] VPO[↑=↓];
VPO → JP[↑OBJ2=↓,↑CASE=に] VP2[↑=↓];
JP → NP[↑=↓] J[↑=↓];
VP2 → V2[↑=↓];
NP → ADJ[↑=↓]　NP[↑=↓];
NP → N[↑=↓];
ADJ → 赤い[↑COLOR="赤"];
N → ブロック[↑PRED="物",↑TYPE="ブロック"];
J → を[↑CASE=を];
J → に[↑CASE=に];
N → ピラミッド[↑PRED="物",↑TYPE="ピラミッド"];
V2 → 載せよ[↑PRED="載せよ(↑OBJ1,↑OBJ2)"];

図 1. 文法規則

各ノードに付随した g4 などの名前は F 構造の名前である。機能的注釈をボトムアップに適用することで最終的にルートノードに付随した F 構造 g0 を得る。

図３．に、今回の実験のためのシステムの構成を示す。

ボトムアップに F 構造を構成してゆく過程で矛盾が生ずれば F 構造は null 値となる。

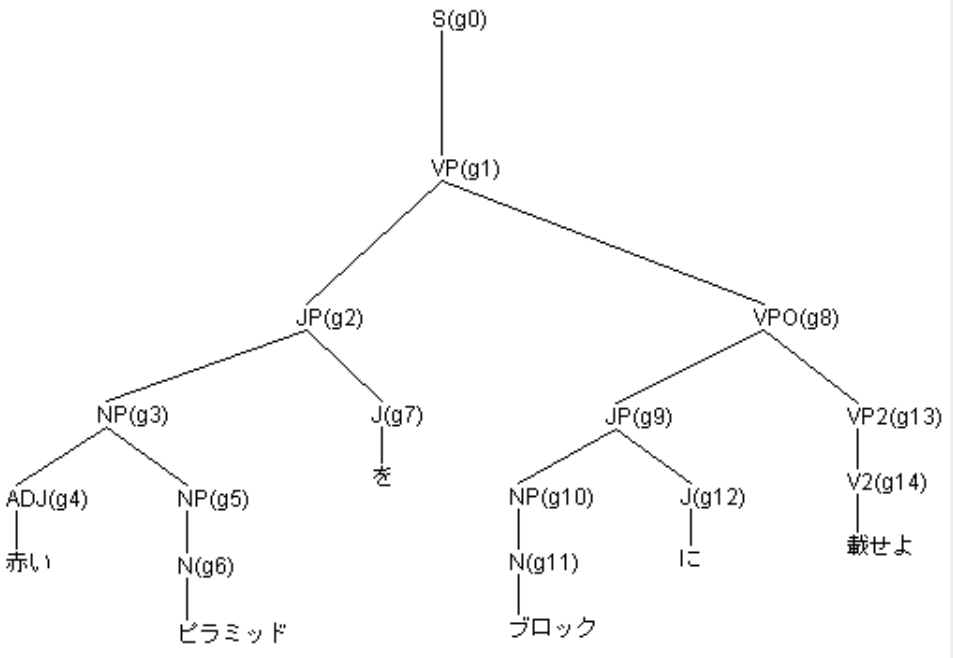


図２．解析木

入力文解析部は、最終的にルートノードに付随する F 構造が null でないときのみ、その F 構造を意味処理部に渡す。

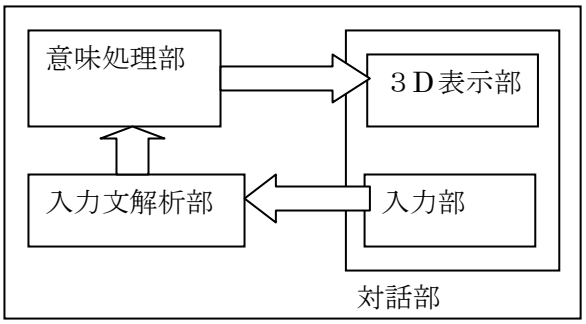


図３．実験システムの構成

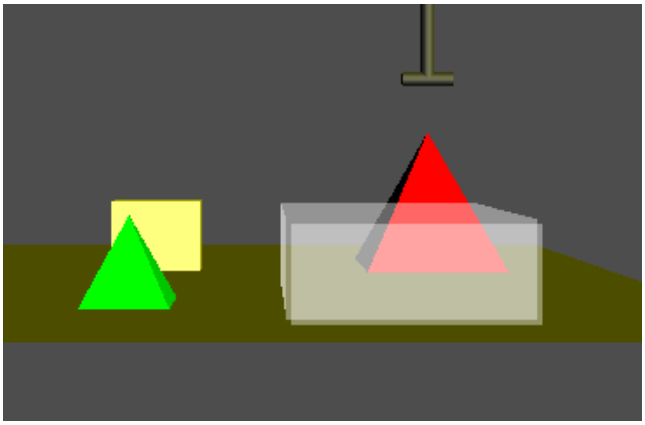
意味処理部は渡された F 構造に基づいて内部状態を変更し、それを３D 表示部に反映させる。積み木の世界には、形状、色、大きさなどの属性を有する物体が存在する。実際には赤くて大きいピラミッド、緑で小さいピラミッド、黄色いブロック、そして半透明な箱

などがある。そして各物体は地面にあるか、またはブロックの上にあるか、あるいは箱に入っているなどの状態をとる。

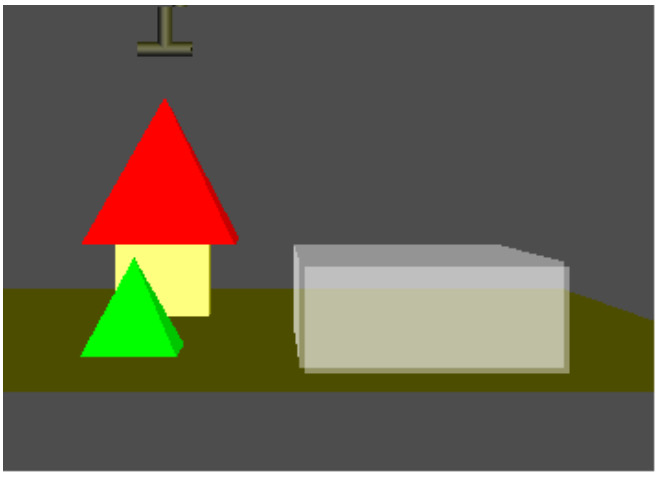
上記の入力文に対しては、図 4 のような F 構造が出力される。これを受け取った意味処理部は図 5. (a)の状態から(b)の状態へと 3 次元表示を動的に変化させる。

```
g0:
[
    PRED  "載せよ(↑OBJ1,↑OBJ2)"
    CASE  に
    OBJ1  g2:
          [
              PRED  "物"
              CASE  を
              COLOR  "赤"
              TYPE  "ピラミッド"
          ]
    OBJ2  g9:
          [
              PRED  "物"
              CASE  に
              TYPE  "ブロック"
          ]
]
```

図４．ルートノードの F 構造



(a)移動前



(b)移動後

図 5. 表示画面の変化

４．まとめ

積み木の世界に適用することで、本研究の方法が、ゲームの場面におけるように、限定された語彙や文法および限定された意味論に対しては有効に応用できる可能性を確認することができた。

結果として、SHRDLU に類似した機能を日本語で実現したことになるが、今のところ、限られた機能を実現しただけであり、今後増やしてゆく予定である。

さらに追求すべき課題としては、入力文の解析部を独立した汎用のモジュールとして取り出すこととともに、その場で必要な文法をすべて最初から記述するのではなく、ある程度の基本となる文法規則群を用意しておいて、必要な部分だけを補うようにすることがある。また形態素の変化や、同じ意味の形態素を容易に扱えるようにする必要がある。

参考文献

[１]Joan Bresnan,"Lexical Functional Syntax",Blackwell(2000).

[２]田中穂積,"自然言語処理―基礎と応用",電子情報通信学会(1999)

[３]http://hci. stanford.edu/~winograd/shrdlu/

[４] http://www.semaphorecorp.com/misc/shrdlu.html