

TR-080

状況意味論に基づいた OA システムの試作

加藤 恒雄
(日本電信電話公社)

September, 1984

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

目次

1. はじめに
2. 状況意味論の概要
 - 2.1 基本的考え方
 - 2.2 状況とその関係としての意味
 - 2.3 言葉の意味
3. 状況意味論の理論
 - 3.1 抽象的状況
 - 3.2 イベント型
 - 3.3 制約
 - 3.4 文の意味
 - 3.5 普通名詞の意味
4. aliasとそのインプリメント
 - 4.1 aliasとは
 - 4.2 インプリメント
5. QAシステムへの応用
 - 5.1 システムの概要
 - 5.2 QAシステムにおけるalias
 - 5.3 状況構造
 - 5.4 関連の評価と整合性チェック
 - 5.5 問題と今後の課題
6. 考察
7. おわりに

謝辞

参考文献

1. はじめに

本レポートは、状況意味論(situation semantics)を基にした文解析システムAliassのPrologによるインプリメントと、それを応用した簡単なQAシステムの実現についての報告である。状況意味論は自然言語の研究にたずさわる多くの人々の関心を集めている新しい意味論であり、Aliassはその説明のために提案された英語に近い人工言語である。また、本メモで報告するシステムは、計算機による自然言語理解や談話理解に状況意味論を応用する可能性を検討するためのものである。

今回のシステムは極めて小規模なものであり、これだけでは状況意味論の能力を計るのに充分とは言えないが、以下の点で状況意味論は談話理解システムの思想的基盤とするのに適切でないかという感触を得た。

- (1) 言語的には等しい文でもその解釈が異なるという前提を持っているため、focus や speech actなど従来語用論の問題として意味論とは別に考えられていたものが統一的に扱える。
- (2) 言語の意味に止まらず、それが発話される世界の構造についても理論が及んでいるため、談話理解システムに必須である推論規則その他が同じ枠組みのなかで扱える。
- (3) 今回のシステムでは扱っていないが、believe, knowなどattitude reportを含んだ文の解釈に問題が少なく、mental stateなどをも統一的に扱えると思われる。
- (4) 以上3点とは若干レベルが異なるが、意味や世界の構造を状況間の関係という概念でとらえているため、logic programming と親和性が高い。

本レポートの構成を以下に示す。

2. では、状況意味論の概要について述べる。ここでは、主に本理論の思想的側面に着目し、状況意味論のbasic concept と言えるものについてまとめる。
3. では、状況意味論の形式化された部分について説明する。状況意味論で用いられる用語の説明などは本章で行なわれる。
4. では、Aliassについて説明し、そのインプリメント手法について述べる。
5. では、Aliassを応用した簡単なQAシステムについて述べ、それを基に文の理解という観点から状況意味論について検討する。
6. では、全体のまとめを行なう。

なお、本レポートの前半は文献(1)を中心にまとめたものである。また、状況意味論の用語に関しては、現在の所、定まった訳語がないため、これらについてはなるべく原語を併記するようにした。

2. 状況意味論の概要

2.1 基本的考え方

状況意味論はBarwise とPerry が1981年頃から提唱し始めた理論である。本理論は自然言語の意味論として提案されているが、後に述べるような理論の性質上、意味というものの一般や世界についての理論を含んでいる。このような広い視野に立った意味論であるということが状況意味論のひとつの特徴であると言えよう。

状況意味論は言葉の外延的意味、つまり、言葉が持つ物事を分類する能力を重視する考え方を発展させたものである。その意味で、この理論は一階述語論理による外延的モデル理論の発展形である。

周知のようにこのような素朴意味論には、幾つかの重大な問題がある。ひとつは明けの明星のパラドックスとして知られているもので、

Morning star is in the sky.

Evening star is in the sky.

というふたつの文が同じことを意味するようになってしまうという問題である。もうひとつは、例えば、the planet closer to the sun than mercury というような現実の世界には存在しないものを示す表現が意味を持たないようになってしまうことである。更に、Helanie believes that Jonny is asleep.のようなattitude report を含んだような文はHelanie の心理的状态を分類しているのであって、外延的な物事を分類しているようには考えられないという問題もある。

状況意味論では、これらの問題を言葉の効果(efficiency of language)という概念を用いて解決する。言葉の効果とは、つまり、異なった人によって、異なった時空間の領域で発話者を取りまく世界との異なった関連(connection)で用いられた言葉はその言語的な意味が等しくても異なった解釈(interpretation)を持つということである。例えば、I am right, you are wrong. という発話はそれが誰によって発話されたかによって、異なった主張をしていることが明らかであるが、これなどは、効果のある文のよい例である。

従来の意味論は、その多くが効果のない文、つまり、どこで誰によって発話されても同じ意味しか持ちえなかった文を中心に扱っていたが、状況意味論では、効果を持つ文を中心に扱い、その同じ効果という概念で素朴意味論の有していた問題の解決を図っている。当然のことながら、現実の世界で多く見られるのはこのような効果のある文であるから、どちらの意味論が強力であるかは明らかであろう。

効果のある発話においては、発話の解釈はそれに利用された表現の意味と発話に関する多くの付加的な事実依存する。発話の真理値は、その解釈が事実の合うかどうか依存する。つまり、世界的事実は発話から真理値を求める間に2回参照されることとなる。1回は与えられた発話の解釈を決定し、もう1回は得られた解釈の真理値を決定する。ここで、状況という概念が導入される。状況の概念によって、発話やその解釈、更には世界的事実までを統一的に扱おうというわけである。

これによって、発話と真理値の間に介在する現実をふたつの部分に分離することができる。ひとつは発話に関する事実であり、もうひとつは、発話によって描かれた状況(*described situation*)に関する事実である。後者は、解釈された発話の真理値に関連するものと言ってよい。一例として、私が*You are sitting.*と発話した場合を考えてみる。発話の解釈つまりこの発話によって描かれる状況は、私が誰で誰に向かって話しているかという事実によって決定する。これらの事実は発話の状況(*discourse situation*)である。この解釈の真偽は、あなたに関する事実つまり現実の状況によって決定されるのである。このように考えると、発話の意味とは、発話の状況とそれによって描かれた状況との関係としてとらえることができる。この考えをより一般化し、意味そのものを状況間の関係としてとらえようとするのが、状況意味論の基礎であり、彼らは、これを意味の関係理論(*relation theory of meaning*)と呼んでいる。

2.2 状況とその関係としての意味

彼らは現実を状況からなるものとしてとらえている。ある時空間の領域で、個体がある性質を持っていることや、それらが関係づけられていることなどがすべて状況である。つまり、我々は常に状況の中になることになる。我々はそれらを認識し、それらから得られるものを利用し、それらに対して態度(*attitude*)を持つ。

彼らはこのような状況を以下に示すような手法で抽象化理論化する。まず、現実の状況(*real situation*)から理論の構成に必要な素材を抽出する。個体(*individual*)、性質(*property*)、関係(*relation*)、領域(*location*)がそれである。これらは、時空間的に変化しないものと考えられ、不変物(*uniformities*)と呼ばれる。これらを集合論を基に組み合わせて作られるものが抽象的状況(*abstract situation*)である。

抽象的状況は状況型(*situation type*)と領域からなる。例えば、Holly が今朝庭で吠えたという状況とHolly が昨日庭で吠えたという状況には何か共通のものがある。状況意味論では、これを、状況型は等しく、領域が異なるふたつの状況とみる。

つまり、状況型は n 項関係 ($n=1$ のときは性質となる) と n 個の個体から値1(true), 0(false)への部分関数(partial function)である。

```
s(bark, Molly)=1  
s(bark, Jackie)=0
```

は、Molly が吠えていて、Jackieが吠えていないような状況型を表わしている。これを略記して以下のように表わすことがある。

```
in s:barks, Molly;yes  
barks, Jackie;no
```

出来事の経過(course of event, 以下coe と略す) は、領域から、状況型への関数である。coe は幾つかの領域で何が生じているかを表わしているものと考えることができる。

```
In e, at l:barks, Molly;yes  
at l1:shouts at, Mr. Levine, Molly;yes  
at l2:barks, Molly;no
```

というcoe e は、Molly はある時ある場所で吠え、Mr. Levine にしかられ、静かになったという経過を表わしている。

状況型とcoe は共にpartial なものとして定義されている。従って、そこに明示されていないものについては、それが起っているかいないかについては、何も述べていないし、そこで起っていることすべてが明示されているわけでもない。このことは状況意味論の特徴のひとつであり、状況意味論を強力なものとしている理由となっている。

以上のように、これら抽象的状況は、集合論を基に定義される抽象物である。状況意味論が基礎としている集合論は有限集合のみを扱うKPU であり、このことが計算量の点でこの理論を有利なものとしている。

状況意味論における意味とは、ただ言語の意味に止まらず、より遍在的な意味一般を指している。先にも述べたように意味とは状況間の関係である。彼らが意味としてとらえているものの背景には、認識論的なもの、実用論的なものがある。つまり、意味とは、生物が生きてゆくために必要とする情報なのである。

生物は、状況の中に生きており、状況を認知してそれに対して適切な行動を取ること(adjust)によりよく生きてゆくことができる。このためには、生物は現実の状況をそこに現れ、認知された不変物を基に分類組織化しなければならない。このような分類組織化を可能にするような関係が意味ある関係(meaningful relation) と呼ばれるのである。幾つか例を挙げる。

煙は火を意味する。

これは火と煙の自然法則的な関係である。

この関係により、煙のある状況から火のある状況を知ることができる。

キスは接触を意味する。

キスと接触は人間によって認知される状況の中の不変物である。この関係は異なった状況間の自動的な連続を意味している。これによって、Holly がBillにキスしたという情報はHolly がBillに接触したという情報を含むことになる。

ベルが鳴るのは授業の終わりを意味する。

ある状態でのベルの音は授業の終りという情報を運ぶことになる。これは状況間の規約的な関係である。このような規約的な関係は誤りを含んでいることがある。

「クッキー」はクッキーを意味する。

Holly が「クッキー」と言うことはそこにクッキーがあることを意味する。「クッキー」という言葉はそれ自体が不変物であり、その言葉を含んだ発話に共通な特徴である。同様にクッキーであるということもクッキーがある状況の共通な特徴である。言語は「クッキー」がクッキーを表わすという点で規約的である。これはふたつの不変物（「クッキー」という言葉とクッキーであるという特徴）の間のシステムティックな関係である。

これらからわかることは状況 s と何らかの形で不変物を分けあっている状況と状況 s_1 と何らかの形で不変物を分けあっている状況との間にシステムティックな関係 M がある時、状況 s は状況 s_1 の情報を含んでいるということである。それらの不変物は物理的なものであってもよいし、言葉のように抽象的なものであってもよい。物理的もしくは抽象的な性質や関係であってもよい。時間や場所やその他の不変物であってもよい。ある状況が他の状況を意味すると言えるのはこのような関係があるときだけである。このような考えから意味が異なった状況の型の間の関係であるという理論（意味の関係理論）が生じる。そして上の例からもわかるように、言葉の意味というものもその枠組みの中のひとつとして意味づけられていることがわかる。

当然のことながら、効果(efficiency)の概念も言語的意味だけでなくすべての意味に拡張される。ある場所である建物のある窓から煙が吹き出しているということは、そのあるビルが火事であるということの意味している。特定のビルから特定の時刻に煙が吹き出しているという状況は決して繰り返されない。次に我々がビルから煙が吹き出すのを見たとき、それはまったく新しい状況である。これら特定された煙の状況は各々特定されたビルが火事であることを示している。これは出来事の意味(Event meaning)と呼ばれる。

しかし、これらの状況は共通なものを持っている。それはビルから煙が吹き出すという共通の型である。この型は火事の意味している。つまり、ビルから煙が吹き出すという出来事の型はそのビルが火事であるという出来事の型を意味している。これは出来事の型の意味(Event-type meaning)と呼ばれる。

出来事型の意味を学ぶことによって、我々は特定の煙の状況から、特定の火事について知ることができる。あるビルから煙は吹き出しているという状況が意味することは、ビルから煙が吹き出しているということの意味とその状況の特殊性によって決定される。これは先に述べた言語の効果の場合とまったく同様である。このことが一般化された効果なのである。

Event-type meaningは従来までの談話理解システムで推論規則と呼ばれていたものと非常に近い役割を持っていることが明らかである。状況意味論において、生物が状況を認知して適切な行動を起こすといわれていることは、談話理解システムが適切な応答をすることと同様であると見なすことができる。自然言語の理論でありながら、このようなもののまでもひとつの理論の中に含んでいることは、本理論によって談話理解というものの全体が扱えることを示唆している。

2.3 言葉の意味

言語的な意味もこれまでに述べたような意味の一般的な構図によってとらえることができる。言葉の本来の役割は情報の伝達にある。つまり、人間は、状況を知覚するひとつの手段として言葉による伝達を利用しているといえる。

例えば、獣医の所にJonny が犬のJackieを連れてきたことを考えてみる。獣医は、彼女のレントゲン写真を見てJackieが骨折していることを知る。Jonny に「彼女は骨折している」と伝える。これによって、獣医とJonny の両者がJackieの骨折を知ることになる。レントゲン写真は獣医にとって意味ある関係である。つまり、そこには、レントゲン写真がこれこれであればそれは骨折を意味するという意味ある関係がある。しかし、Jonny にとってはそうではない。Jonny にとって意味ある関係であったのは、獣医の「彼女は骨折している」という発話と犬の骨折という出来事との関係である。

この例から言葉の意味が先に述べた一般的な意味と同じように状況間の関係であるということがわかる。つまり、人間の言語の意味とは、発話という出来事とそれによって描かれる対象であるような現実の一面との関係である。そして、この関係はその言語を利用している言語共同体によって決定される規約的なものである。

言語の意味である関係が規約的なものであるということは、それが誤りを含む可能性があることを示している。人々は誤りやその他の理由によってしばしばこの関係における規約を破っている。もし、この関係が誤りを含んでいなかったとしたら人は発話の真理値というものに関心をはらうことはなかったに違いない。そして発話の意味である関係が誤ったものであるか否かはその状況に対して適切な行動をとるために極めて重要な問題となる。このことが、言語の意味を考えるとときにその真理条件が重要な意味を持つ理由である。

正しさの問題は話者にとって言語的な意味の制約であり、宣言的叙述の正しい使い方に対する制約であるのだが、その他にも幾つかの制約がある。例えば、窓を開けてほしくない時に「窓を開けて下さい」ということはその制約を破っていることになる。そう考えることで、speech actの問題が状況意味論のスコープに入ってくることになる。

以上のような考え方によって、単純な宣言文の意味は発話とそれによって描かれた状況の関係であると言える。特定化された場合におけるそのような文による叙述の解釈が描かれた状況(described situation)となる。

I am sitting. という文を考えてみる。その意味は、発話 u と状況 e との間の関係である。この場合、 u において、領域 l で発話している個体 a があり、 e において、 a が領域 l で座っているという関係である。このような関係を以下の様に記述する。

$$\begin{aligned} &u [I \text{ AM SITTING }] e \\ &\quad \text{iff} \\ &\text{There is a location } l \text{ and an individual } a, \text{ such that} \\ &\quad \text{in } u: \text{at } l: \text{speaks, } a; \text{yes} \\ &\quad \text{in } e: \text{at } l: \text{sits, } a; \text{yes} \end{aligned}$$

このような関係の外延は抽象的状況の対の非常に大きなクラスとなる。Fredが発話者であり、その発話がある特定化された領域 l でなされたとした場合、領域 l でFredが座っているようなすべての状況がこの発話の解釈となり得る。Fredが座っているだけの状況もそうであるし、Fredと他の多くの人が座っている状況もそうである。ひとつの発話は普通多くの異なった状況を描くことになる。このため、解釈をそのような状況のクラスとして考えることもできる。

ある特定化された領域で発話がなされたとき、その叙述は描かれる状況が解釈のひとつになるよう描かれた状況を制約する。文の部分も基本的にそれと同様で、そのような制約を作りあげることに関与する。例えば、名詞句の意味は、発話と個体の関係であるし、動詞の意味は、発話と関係や性質との関係である。

以上が状況意味論の背景とも言えるものである。

3. 状況意味論の理論

本章では、状況意味論の理論やそこで用いられる用語の定義について述べる。順序として、まず、状況の理論について述べ、その後言語の意味に関する理論について論じる。

3.1 抽象的状況 (abstract situation)

3.1.1 素材 (primitive)

現実(real situation)から以下のものを抽象的状況の素材として抽出する。

・ 個体 (individual)

$a, b, c, \dots \in A$ A は個体の集合

個体は現実の中のものであり、ある個体が他の個体の部分であってもよい。

・ 関係 (relation)

$r, r_1, r_2, \dots \in R$ R_n を n 項関係の集合とした場合、 $R = \bigcup R_n$

$n=1$ の場合は性質(property)と呼ばれる。 人間であるなど

$n=2$ の場合は 2項関係。 母であるなど。

$n=0$ の場合は状態(situational state) と呼ばれる。雨が降っているなど。

・ 時空間領域 (space-time location)

$l, l_1, l_2, \dots \in L$ L は絶対的4次元時空間の領域の集合

時空間領域の間の関係は外延的なものとし、そこに成立している関係は決定可能とする。

時空間領域の間の関係の基本的なものは次の通り

$l_1 < l_2$ l_1 が完全に時間的に l_2 に先行している。

$l_1 o l_2$ l_1 が時間的に l_2 と重なっている。

$l_1 \theta l_2$ l_1 が空間的に l_2 と重なっている。

$l_1 \subset_t l_2$ l_1 が時間的に l_2 に含まれている。

$l_1 \subset_s l_2$ l_1 が空間的に l_2 に含まれている。

$l_1 \subset l_2$ l_1 が時空間的に l_2 に含まれている。

もっとも大きい領域として lu (universal location)を仮定する。

これは、すべての領域を時空間的に含むような領域である。

以上の素材とそれらを組合せてできる複合体をオブジェクト(object)と呼ぶ。

3.1.2 抽象的状况

構成素列(constituent sequence)

r が n 項関係で、 $x_1, \dots, x_n$ が object であるとき、 $y = \langle r, x_1, \dots, x_n \rangle$ は構成素列である。

状況型(situation type)

構成素列と値 0, 1 との関係を状況型と呼ぶ。

つまり、状況型 s は、 $s = \{ \langle y, i \rangle \mid y \text{ は構成素列、} i \text{ は } 0, 1 \}$ であるような対の集合である。
 $s = \{ \langle \langle r, a, b \rangle, 1 \rangle, \langle \langle r_1, a, c \rangle, 0 \rangle \}$ のとき、これを以下のように記述することがある。

$s := r, a, b; \text{yes}$

$r_1, b, c; \text{no}$

更に s を部分集合として含むような s_1 は以下のように記述される。

$\text{in } s_1 : r, a, b; \text{yes}$

$r_1, b, c; \text{no}$

整合的(coherent)

状況型 s が以下の条件を満たすとき、整合的と呼ばれる。

① s は等しい構成素列に異なる値を与えていない。

つまり、以下のようなものは整合的でない。

$\text{in } s : r, x_1, \dots, x_n; \text{yes}$

$\text{in } s : r, x_1, \dots, x_n; \text{no}$

② s は異なる object が等しいという構成素列に yes を与えていない。

つまり、以下のようなものは整合的でない。

$\text{in } s : \text{same}, a, b; \text{yes}$

③ s は同一の object が等しいという構成素列に no を与えていない。

つまり、以下のようなものは整合的でない。

$\text{in } s : \text{same}, a, a; \text{no}$

ここで、①③は、論理的無矛盾性のためである。また、②によって以下のような場合に生じる矛盾が防がれる。

$\text{in } s : r, a; \text{yes}$

$\text{in } s : r, b; \text{no}$

$\text{in } s : \text{same}, a, b; \text{yes}$

事態(state of affair)

l が領域、 s_0 が状況型であるとき、その対 $s=\langle l, s_0 \rangle$ は事態と呼ばれる。

事態とは静的な状況の記述であり、ある領域で何が生じているかについて述べている。

以下の条件を満たすとき事態 s はfactual と呼ばれる。

$\text{in } s : r, a_1, \dots, a_n; \text{yes}$ ならば領域 l において $a_1, \dots, a_n$ が関係 r にある。

$\text{in } s : r, a_1, \dots, a_n; \text{no}$ ならば領域 l において $a_1, \dots, a_n$ が関係 r がない。

以下の条件を満たすとき事態 s はactualと呼ばれる。

領域 l において $a_1, \dots, a_n$ が関係 r にあるならば $\text{in } s : r, a_1, \dots, a_n; \text{yes}$ 。

領域 l において $a_1, \dots, a_n$ が関係 r がないならば $\text{in } s : r, a_1, \dots, a_n; \text{no}$ 。

出来事の経過(course of event) 以後coe と略称

l が領域、 y が構成素列、 i が $1, 0$ のとき、その3組 $\langle l, y, i \rangle$ の集合 e はcoe と呼ばれる。事態に比較して、coe は動的な状況の記述である。

coe は以下のように略記される。

```
e := at l : hungry Molly; yes
      at l1 : eating, Molly; yes
           sleeping, Jackie; yes
      at l2 : hungry, Molly; no
      (l < l1 < l2)
```

また、上のような e を部分として持つようなcoe e_0 は以下のように記述される。

```
in e0 at l : hungry Molly; yes
      at l1 : eating, Molly; yes
           sleeping, Jackie; yes
      at l2 : hungry, Molly; no
      (l < l1 < l2)
```

coe e から領域 l における状況型 $\langle y, i \rangle$ への関数を $e^*(l)$ とする。

つまり、 $e^*(l) = \{ \langle y, i \rangle \mid \langle l, y, i \rangle \in e \}$

coe e に含まれる全ての領域 l において、 $e^*(l)$ が整合的であるとき、

coe e は整合的であると呼ばれる。

3.1.3 現実と状況

状況構造(structure of situation)

状況構造は直観的にはある世界を定義したものであると言える。

以下の条件を満たすような空でない部分クラス(sub collection) M_0 を持つcoeのクラス(collection) M からなるものを状況構造と呼ぶ。

1. すべての $e \in M_0$ は整合的
2. もし、 $e \in M_0$ で、かつ $e_0 \subset e$ であれば、 $e_0 \in M$
3. もし、 X が M の部分集合(subset)であれば、 X のすべてのcoe がその部分であるようなcoe e が M_0 の中に存在する。
4. もし、 C が M における制約(constraint)であれば、 M は C を満たす(respect)。
4. については 3.3節で述べるが、およそその意味はその世界が満たすべき制約を満たしているという意味である。

M_0 に含まれるcoe はその構造においてactualなcoe であると呼ばれる。

M に含まれるcoe はその構造においてfactual なcoe であると呼ばれる。

強固(persistence)

coe の集合 P は以下の条件を満たすとき、強固であると呼ばれる。

$e \in P$ かつ $e \subset e_1$ ならば $e_1 \in P$

一般に情報は強固なものである。なぜなら、"I am sitting." という発話の解釈 P は、発話の領域を l 、発話者を a とすると、 $P = \{e \mid \text{in } e \text{ at } l; \text{sitting } a; \text{yes}\}$ という強固なcoeの集合となる。しかし、定冠詞theを含むような発話の解釈は強固とはならない。ある発話の解釈が強固であるか否かはその真理値を決定する際に問題となってくる。これについては 3.5節で述べる。

情報伝達の

M_0 をcoeの集合 M における2項関係とする。つまり、 M_0 をcoeの順序対であるとする。

以下の条件を満たすとき、 M_0 をある構造において情報伝達の(informational)と呼ぶ。

すべてのfactualなcoe $e \in M \cap M_0$ において、

$e M_0 e_1$ であるようなfactualなcoe $e_1 \in M$ が存在する。

また、このとき、 M_0 を意味がある(meaningful)と呼び、 $e M_0 e_1$ を e_1 は e からの意味のある付加(meaningful option) であると呼ぶ。

$[e] = \{e_1 \mid e M_0 e_1\}$ と記述するものとする。単に $[e]$ と書くことがある。

M_0 は、 $e \in M \cap M_0$ ならば、 $[e] \cap M$ が空でないときに情報伝達のである。

$[e]$ が強固であるとき、情報が強固であると呼ぶ。

この定義は、前章で述べた意味の関係理論と関連する。ある出来事 e が情報を伝達する（意味がある）ということは、ある H_0 があり、 eH_0e_1 であるようなある e_1 が、 e が factual l であるときに factual であるということに他ならない。

例えば、宣言的な発話の集合を HF とし、 H_0 を正しい宣言と状況の間関係とする。領域 l での“A dog is biting Molly.”という発話 u は領域 l における actual な出来事 e_1 に関する情報を伝達する。ここで、関係 H_0 によって、 $[u]$ は以下のような coe e_2 から成ることがわかる。

ある a があって、

in e_2 : at l : biting, a , Molly; yes

dog, a ; yes

actual な出来事 e_1 はこのうちのひとつである。また、 $[u]$ は強固である。

3.2 イベント型

未定項(basic indeterminate)

特定されていない素材を表現するものとして未定項を導入する。

$\dot{a}, \dot{b}, \dots$ 個体未定項(individual indeterminate)

$\dot{r}, \dot{s}, \dots$ 関係未定項(relation indeterminate)

$\dot{l}, \dot{l}_1, \dots$ 領域未定項(location indeterminate)

イベント型(event type)

オブジェクトとして、未定項を含んだ coe をイベント型と呼ぶ。

イベント型を表現するには、 $E, E_1, \dots$ を用い、

必要があれば、未定項をその後に付ける。

例えば、 $\dot{a}, \dot{r}, \dot{l}$ が未定項であるイベント型は $E(\dot{a}, \dot{r}, \dot{l})$ と表現される。

イベント型は coe のクラスと考えることができる。

アンカー(anchor)

あるイベント型 $E(E = E(\dot{a}, \dots, \dot{r}, \dots, \dot{l}, \dots))$ に対して、関数 f が、 e の中に幾つかの未定項に個体、関係、領域を与えると、 f は E のアンカーと呼ばれる。

アンカー f が与えられたとき、 f の定義域である x を値 $f(x)$ で置換したような新しいイベント型が得られるが、これを $E[f]$ と記述する。

f が l のすべての未定項について定義されていたとき、つまり、 $E[f]$ が coe の時、 f を l のtotal anchorと呼ぶ。

あるcoe e はイベント型 E に対して $E(f)$ が e の部分となるようなアンカー f が存在するとき、タイプ E であると呼ばれる。

ルール

未定項に以下のように定義されるルールを加える。

$\hat{r} = \langle \hat{x}, E \rangle$ $\hat{x}$ は未定項、 E はイベント型

ルールへのアンカー

f が E のアンカーであり、 $f(\hat{r}) = f(y)$ であるとき、 f はルール $\hat{r} = \langle \hat{y}, E \rangle$ のアンカーであると呼ばれる。

文脈(context)

イベント型 $E(\hat{r}_1, \dots, \hat{r}_n)$ 、ここで $\hat{r}_1, \dots, \hat{r}_n$ はルールを含む未定項に対して、 e のすべての未定項を一意にアンカーできるcoe e を E の文脈と呼ぶ。

スキーマ(schema)

スキーマはイベント型の集合である。 $S = \{E_1, E_2, \dots\}$

f が S の要素であるイベント型のアンカーであるとき、 f は S のアンカーであると呼ばれる。このとき、 $S[f]$ は $E \in S$ である $E[f]$ の集合となる。

e があるタイプ E ($E \in S$) であるとき、 e はタイプ S であると呼ばれる。このことからわかるようにスキーマはイベント型の選言的な集りである。

3.3 制約(constraint)

3.3.1 その意味と分類

制約(constraint)とは、直観的には世界の構造やそこで成り立つ状況間の関係を表わしている。前章でも述べたが、この制約に適切に反応することで、生物は行きてゆけるわけである。

制約は以下のように分類される。

無条件制約(unconditional constraint, ubiquitous constraint)

最大の領域 lu で成立する制約

条件制約(conditional constraint)

特定の状況で成立する制約

別の観点からは以下のように分類される。

集合論からの制約

状況意味論が基づいている集合論KPU からくる制約であり、集合はその要素として自分自身を含まない、などがそれである。

形而上学的制約

無矛盾の原則からくる制約で、すべてのactualなcoe は整合的、任意の2つのfactualなcoe はあるactualなcoe の部分となっているなど。状況構造の定義の中に含まれている。

現実の構造からくる制約

必然的制約(necessary constraint)

性質や関係の中に含まれている必然的な制約

女性是人間である、キスは接触である、など。

nomical制約(nomical constraint)

自然法則などからくる犯しがたい関係としての制約

投られたボールは落ちるなど

規約的制約(conventional constraint)

生物の共同体の中で約束されることによって生じる制約

ベルが鳴ると授業の終わりであるなど。

当然のことながら、条件制約はすべて現実の構造からくる制約である。

3.3.2 定義

包含(involve)

イベントのタイプの間の関係involveを導入する。

タイプS1であるすべてのactualなcoe c がタイプS2のactualなcoe の部分であるとき、S1がS2を包含(involve) されると呼ばれる。

無条件制約

以下のような事態C0が単純な制約(simple constraint) である。

$C0 := \lambda u : \text{involves}(F, S; \text{yes})$

ここで、F はイベント型、S はスキーマ

単純な制約を部分として含む任意のcoe は制約と呼ばれる。

例えば、

$E := \text{at } \dot{I} : \text{kisses}, \dot{a}, \dot{b}; \text{yes}$

$E1 := \text{at } \dot{I} : \text{touches}, \dot{a}, \dot{b}; \text{yes}$

のとき、 $C0 := \text{at } lu : \text{involves}, E, E1; \text{yes}$ は制約である。

C を単純制約とする、つまり、

$C := \text{at } lu : \text{involves}, E, S; \text{yes}$

このとき、

- ① $\text{coe } e0$ がタイプ E であれば、 $e0$ は C に対して (with respect to C) 意味がある (meaningful)。これを $e0 \in \text{MF}_C$ と記述する。
- ② $e0$ が C に対して意味があり、 E に対するすべての total anchor f に対して、 $e0$ がタイプ $E[f]$ であれば、 $e1$ がタイプ $S[f]$ であるとき、 $e1$ は C に対して $e0$ からの意味ある付加であると呼ばれる。これを $e0 \text{ H}_C e1$ と記述する。
- ③ $e0$ が C に対して意味があるとする。 $e1$ と $e0$ の和が整合的でないとき、 $e0$ は C に対して $e1$ を排除 (preclude) すると呼ぶ。これを $e0 \text{ preclude}_C e1$ と記述する。

状況構造が単純制約 C を満たす (respect) とは以下の条件を満足することである。

その構造におけるすべての actual な $e0$ について、 $e0$ が C に対して意味があれば、ある factual な $e1$ が C に対して $e0$ からの意味ある付加になる。

もし、ある状況構造が制約 C の部分であるすべての単純制約を満たせば、その構造は制約 C を満たす。

条件制約

以下に示すような単純制約 C とイベント型 $E1$ があったとき

$C := \text{at } lu : \text{involves}, E, S; \text{yes}$

次のものは条件制約である。

$\text{at } lu : \text{involves}, E + E1, S; \text{yes}$

より、一般的には任意の制約 C とスキーマ S を考え、 $C + S$ を $C0 + E0$ と定義する。ここで、 $C0$ と $E0$ はそれぞれ C と S の部分である。このとき、対 $\langle C, S \rangle$ からなる CIS が条件制約である。

幾つか制約の例を挙げる。

①妻を持っている人は結婚している。

C1:= at *lu* : involves, *E*, *E1*;yes

E:= at *l* : wife-of, *a*, *b*;yes

E1:= at *l* : man, *b*;yes
 married, *b*;yes

②コインを投げたら表か裏のどちらかが出て、両方出ることはない。

C2:= at *lu* : involves *E*, *S*;yes

E:= at *l* : tossing, *a*, *b*;yes
 coin, *b*;yes

S:=({*Eh*, *Et*})

Eh:= at *l* : heads, *b*;yes
 tails, *b*;no

Et:= at *l* : tails, *b*;yes
 heads, *b*;no

3.4 文の意味

宣言的な文φの意味は、φが発話された状況*u*とその発話によって描かれた状況*e*の間
の関係*u* [φ] *e*としてとらえられる。この関係は、*u*と*e*の両方を制約する。

この関係の*e*に対する制約が従来、意味としてとらえられていたものである。

*u*に対する制約は従来、speech actやspeaker intentionなどとされていたものである。

発話の意味を記述するためには、発話の状況と描かれた状況に存在する不変物と、それ
らの間にある制約を描かなければならない。これらをそれぞれ、発話状況(discourse sit
uation)、関連(speaker connection)と呼ぶ。

発話状況は以下のようなイベント型DUで表現される。これは、いつどこで誰が誰に向か
って何を発話しているかを表わすものである。

DU:= at *l* : speaking, *a*;yes
 addressing, *a*, *b*;yes
 saying, *a*, *U*;yes

ここから次のルールが導かれる。

speaker=< $\dot{a}$, DU>	発話者	adで略記
addressee=< $\dot{b}$, DU>	聞き手	bdで略記
disc-loc=< $\dot{l}$, DU>	発話のあった領域	ldで略記
expression=< $\dot{U}$, DU>	発話内容	Udで略記

関連(speaker connection)

名前や指示代名詞は、発話者が何かを状況の中の個体を参照(refer)するのに用いられる。このようなものを表わすためにロールrefを考える。

```
ref=< $\dot{a1}$ , REF>
REF:= at  $\dot{l}$ :speaking, $\dot{a}$ ;yes
      addressing, $\dot{a}$ , $\dot{b}$ ;yes
      saying, $\dot{a}$ , $\dot{U}$ ;yes
      refering-to, $\dot{a}$ , $\dot{U}$ , $\dot{a1}$ ;yes
```

また、時制を表わすもの、例えば、is biting のisなどもある領域を参照すると考えることができる。これらを時標識(tense marker)と呼ぶ。例えば、isの場合

```
in u : at ld : refering-to, $\dot{a}$ ,is, $\dot{l1}$ ;yes
      (where ld o  $\dot{l1}$ )
```

このように、発話は言葉U からそれによって参照されているものc(U)への部分関数を与えていることがわかる。この関数を関連(speaker connection)と呼び、c で表わす。

この関数は、発話u と描かれた状況s とのリンクをとるものである。例えば、

```
in s : at l1 : biting,Jackie,Holly;yes
      (l1 o ld)
      where l1=c(is),Jackie=c(jackie),Holly=c(molly)
      ただし、jackieは言葉"jackie"であり、Jackieは個体Jackieである。
```

以後、発話状況u として、上に定義した発話状況d と関連 cを考え、

u [ϕ] eをd, c [ϕ] eと表現することとする。

compositionalityとsetting

文の意味は、その部分である文の成分の意味をシステムティックに合成してゆくことで得られる。文の成分はそれ自体で、状況を描くわけではなく、文が描く状況e を構成する成分となるような状況の構成要素を描くことになる。従って文の成分 α の意味は、発話状況、関連、状況の要素、描かれる状況の4項関係となる。つまり、

d, c, [α] σ , e

この σ をsetting と呼ぶ。

例えば、is biting Molly という動詞句の意味は以下のように表わされる。

$$\begin{aligned} & d, c[\text{is biting molly}] \dot{a}, e \\ & \text{iff} \\ & \text{in } e: \text{at } l : \text{biting}, \dot{a}, \text{Molly}; \text{yes} \\ & \text{ここで、Molly} = c(\text{molly}), l (=c(\text{is})) \text{ o ld} \end{aligned}$$

このように動詞句の場合には個体 a が setting となる。この a は文の他の部分によって、決定されるわけである。

名前や代名詞の意味は同様に以下のように表わせる。

$$\begin{aligned} & d, c[\text{name}] \dot{a}, e \quad \text{iff } c(\text{name}) = \dot{a} \\ & d, c[I] \dot{a}, e \quad \text{iff } \dot{a} = \text{ad} \\ & d, c[\text{you}] \dot{a}, e \quad \text{iff } \dot{a} = \text{bd} \\ & d, c[\text{it}] \dot{a}, e \quad \text{iff } c(\text{it}) = \dot{a} \end{aligned}$$

その他の意味については次章で述べる。

重文の意味は以下のように表わされる。 ϕ と ψ を文として、

$$\begin{aligned} & d, c[\phi \text{ and } \psi] e \\ & \text{iff} \\ & \text{there is an extension } c1 \text{ of } c \text{ such that} \\ & d, c1[\phi] e \text{ and } d, c1[\psi] e \end{aligned}$$
$$\begin{aligned} & d, c[\phi \text{ or } \psi] e \\ & \text{iff} \\ & \text{there is an extension } c1 \text{ of } c \text{ such that} \\ & d, c1[\phi] e \text{ or } d, c1[\psi] e \end{aligned}$$

文の選言的な結合の意味は、2つの文の意味の単純な選言ではない。共通の関連 $c1$ を持っていることで、例えば、

Jackie, a dog, is barking or she is sneezing.

という文において、jackieが吠えていないときでも、jackieが犬であるということが言える。その意味では文の解釈においては厳密な意味での compositionality は成立しない。

文の解釈は以下のような状況のクラスとなる。

$$[\Phi] = \{e \mid d, c[\Phi]e\}$$

$[\Phi]$ を陳述(statement)と呼ぶ。

つまり、陳述は発話内容と発話状況と関連の3つによって決定される。

陳述の中にactualなcoeが含まれていればその発話は真である。

3.4 普通名詞の意味

普通名詞や限定表現は、ある状況の中でそれが有する性質を記述することで、あるobjectを同定するためのものである。つまり、objectを同定するためある状況で何が生じているかを利用するわけである。この利用される状況は資源状況(resource situation)と呼ばれる。例えば、a dog という表現は、資源状況において、犬であるという特徴を有した個体を示している。

定冠詞による表現の意味は、以下のように表わすことができる。

$$d, c[\text{the } \pi] \dot{a}, e$$

iff

$$d, c[\pi] \dot{a}, e_1 \text{ and there is at most one } \dot{b} \text{ such that } d, c[\pi] \dot{b}, e_1$$

ここで e_1 は資源状況である。上からもわかるようにthe π の場合、その特徴を持った $\dot{b}$ は資源状況の中にひとつしか存在してはならない。このことから、定冠詞を含んだ文の解釈は強固とはならない。

$d, c[\text{the } \pi] \dot{a}, e_1$ は、 e_1 から個体 $\dot{a}$ への部分関数を表現していると考えられる。資源状況 e_1 において、Dimpleが私を噛んだのであれば、

$$d, c[\text{the dog that bit me}](e_1) = \text{Dimple} \quad \text{である。}$$

これらの資源状況は関連によって指定される。

名詞句の用法

資源状況として何をとるかによって名詞の用法を幾つかに分類することができる。

指示的用法(referential use)

the man in a red vest is a fool

という発話における名詞句の用法は指示的である。このことは、既に発話者や聞き手にとって利用できる資源状況があるということである。

属性的用法(attributive use)

The player that won the race is Fred.

この発話における資源名詞句の用法は、属性的である。名詞句の資源状況が既に存在したとした場合、この発話の解釈はFredはFredであると言っているにすぎないことになる。従って、この場合の資源状況は描かれる状況に等しい。

同格用法(appositive use)

Jim was chasing Molly, the dog, that bit me

このような用法は同格的と呼ばれる。この場合も資源状況は描かれる状況に等しい。

普通名詞を含んだ文において、資源状況が決定された陳述を完全(complete)と呼ぶ。

完全な陳述はその中にactualなものを含めば真である。

従って、2つ場合で陳述は偽となる。

資源状況がactualでないとき。

描かれた状況がactualでないとき。

4. alias とインプリメント

4.1 aliasとは、

alias(Artificial Language for Illustrating Situation Semantics)は、その名の通り、状況意味論を説明するための人工言語である。これは、冠詞を扱っていないdeterminer-free aliasと単数名詞つまりは冠詞までをも含むsingular alias のふたつが文献(1)の付録として提案されている。determiner-free aliasはsingular alias のサブセットであるから、ここでは後者についてのみ説明する。

aliasの入力となるのは、多少修正された英文 α とそれに伴う意味的な曖昧さを除くための意味特徴 W の対 $\langle \alpha, W \rangle$ である。出力は、 α の解釈であり、これは d, c, e の関係を表現する一種の論理式として得られる。

以下に入力となる α の例(文法のところで述べるがこれは W の情報を含めた記法である)とそれに対応する英文を挙げる。

```
{jackie}1 bite w molly and she(1) excite w  
Jackie was biting Molly and she was exciting.
```

```
[{(the dog n)0}1 love n (the boy n who own n her(1))1]1  
The dog loves the boy who owns her.
```

```
[{jackie <(a dog n)1>}1 bark n or she(1) sneeze n]1  
Jackie, a dog, is barking or she is sneezing.
```

意味特徴としては、以下のものがある。

Scat(α)

α の意味カテゴリ 未定項の集合であり、

setting σ は α のScatにアンカーされるものとする。

未定項には以下のものがある。

個体未定項 a, b 領域未定項 l 真偽値未定項 tv 、集合未定項 x イベント未定項 e_0, e_1

個体未定項 a, b はそれぞれロールを持っており、 a は主格、 b は直接目的格を表わす。

イベント未定項 e_0, e_1 の内、 e_0 は入力全体の資源状況として最終的には $RS(\alpha)$ となる。

e_1 は入力全体によって描かれる状況 e によってアンカリングされる。

つまり、 e_0 は指示的用法の時、 e_1 は属性的用法、同格的用法の時の資源状況となる。

例えば、冠詞を含む名詞句のScatは $\{a, e_i\}$ であり、その用法によって $i=0, 1$ が決まる。

また、それ以外の名詞(名前、代名詞)のScatは $\{a\}$ となる。

Var(α)

α の中に現れる変項(variable)の集合

aliasにおいてでは代名詞を変項と呼ぶ。

Tense(α)

α の中に現れる時標識(tense marker)の集合

AF(α)

α を先行詞(antecedant)として持つ変項の集合

$x \in \text{AF}(\alpha)$ は α が x の先行詞であることを示している。

例えば、先の文においてはsheが何を指すかという曖昧性がある。

この曖昧性は次のようなAFの差異として表現できる。

$\text{AF}(\text{jackie})=\{\text{she1}\}, \text{AF}(\text{molly})=0$

$\text{AF}(\text{jackie})=0, \text{AF}(\text{molly})=\{x1\}$

$\text{AF}(\text{jackie})=\text{AF}(\text{molly})=0$

換言すれば、入力にこの意味特徴を付加することで意味の曖昧性をなくすわけである。

RS(α)

α の中に現れる資源状況(resource situation)の集合

関連cの定義域は、これらの意味特徴によって以下のように定義される。

$\text{domain}(c) \subseteq \text{Var}(\alpha) \cup \text{Tense}(\alpha) \cup \text{RS}(\alpha)$

settingの定義域はScatに一致する。

レキシコン

aliasにおける単語のカテゴリとそれぞれの意味は以下のとおりである。

RS relational symbol 動詞と普通名詞に対応する。

一項関係 自動詞に対応する。Scat= $\{\dot{a}, \dot{i}, \dot{tv}\}$

シンボルR1に対応する一項関係を r としてその意味は

$d, c[R1]\sigma, e \text{ iff } \langle r, a_\sigma, t_{v_\sigma} \rangle \in e(l_\sigma)$

二項関係 他動詞に対応する。Scat= $\{\dot{a}, \dot{b}, \dot{i}, \dot{tv}\}$

シンボルR2に対応する二項関係を r としてその意味は

$d, c[R2]\sigma, e \text{ iff } \langle r, a_\sigma, b_\sigma, t_{v_\sigma} \rangle \in e(l_\sigma)$

IT individual terms 代名詞や固有名詞に対応する。

3種類に分類できる。すべて、 $Scat=\{\dot{a}\}$

一、二人称代名詞

$d, c[I] \sigma, e \text{ iff } a_{\sigma} = ad \text{ speaker of } d$

$d, c[you] \sigma, e \text{ iff } a_{\sigma} = bd \text{ addressee of } d$

三人称代名詞 インデックスを付加した変数と考える。

$d, c[Xn] \sigma, e \text{ iff } a_{\sigma} = c(Xn)$

固有名詞

$d, c[J] \sigma, e \text{ iff } a_{\sigma} = j$

ここで、 j はシンボル J に対応する個体

より厳密に言った場合、名前は直接、個体に対応するものではないが、ここでは簡単のため、固有名詞と個体との対応は一意であるとする。

TM tense marker 時制を示すものに対応する。

現在を表わすもの n と過去を表わすもの w を考える。 $Scat=\{\dot{I}\}$

$d, c[nj] \sigma, e \text{ iff } I_{\sigma} = c(nj) \text{ and } I_{\sigma} \geq Id$

$d, c[wj] \sigma, e \text{ iff } I_{\sigma} = c(wj) \text{ and } I_{\sigma} < Id$

DET determiner 冠詞に対応する。ここでは、 a と the を扱う。

ともに、 $Scat=\{\dot{X}, \dot{a}\}$

$d, c[a] \sigma, e \text{ iff } a_{\sigma} \text{ is in } X_{\sigma}$

$d, c[the] \sigma, e \text{ iff } a_{\sigma} \text{ is the unique thing in } X_{\sigma}$

構文カテゴリ

Aliasは以下の構文カテゴリを設定している。

LRP (Located Relation Phrase) 時制と真埋値が定まった動詞に対応する

PrPh (Property Phrase) 動詞句、普通名詞に対応する。

NP (Noun Phrase) 名詞句に対応する。

s (Sentence) 文に対応する。

文法

Aliasの文法は、構文カテゴリの生成とそれに対応する意味と意味特徴によって記述される。意味に対するルールはすべてif and only ifの形式で述べられている。

Aliasの文法を以下に示す。

なお、すべてのルールで、RSとTense はそのコンポーネントのユニオンとなる。

S Rules

S1 α がNPで π がPrPhであるとき、 $(\alpha \pi)$ はS である。

$$\text{Scat} = (\text{Scat}(\alpha) \cup \text{Scat}(\pi)) - \{\dot{a}\}$$

$$\text{Var} = \text{Var}(\alpha) \cup (\text{Var}(\pi) - \text{AF}(\alpha))$$

$$\text{AF} = \text{AF}(\alpha) \cup \text{AF}(\pi)$$

$$d, c[(\alpha \pi)]\sigma, e \text{ iff}$$

以下のような c を拡張(extend)した c_1 と σ を拡張した σ_1 がある。

$$d, c_1[\alpha]\sigma_1, e \text{ かつ } d, c_1[\pi]\sigma_1, e$$

S2 ϕ と ψ がS であるとき、 $(\phi \text{ and } \psi)$ はS である。

$$\text{Scat} = \text{Scat}(\phi) \cup \text{Scat}(\psi)$$

$$\text{Var} = \text{Var}(\phi) \cup (\text{Var}(\psi) - \text{AF}(\phi))$$

$$\text{AF} = \text{AF}(\phi) \cup \text{AF}(\psi)$$

$$d, c[(\phi \text{ and } \psi)]\sigma, e \text{ iff}$$

以下のような c を拡張した c_1 と σ を拡張した σ_1 がある。

$$d, c_1[\phi]\sigma_1, e \text{ かつ } d, c_1[\psi]\sigma_1, e$$

s3 ϕ と ψ がS であるとき、 $(\phi \text{ or } \psi)$ はS である。

意味特徴についてはS2に同じ

$$d, c[(\phi \text{ or } \psi)]\sigma, e \text{ iff}$$

以下のような c を拡張した c_1 と σ を拡張した σ_1 がある。

$$d, c_1[\phi]\sigma_1, e \text{ または } d, c_1[\psi]\sigma_1, e$$

S4 ϕ がScatに $\dot{e}i$ を含むs であれば、Scatにある $\dot{e}i$ をAFに移したのもS である。

その他の意味特徴は変化しない。このような ϕ を $[\phi]1$ と略記すると、

$$d, c[[\phi]1]\sigma, e \text{ iff}$$

$$\sigma \text{ を拡張した } \sigma_1 \text{ があって、} e_{1_{\sigma}} = e \text{ かつ } d, c[\phi]\sigma_1, e$$

RS Rule

α が Scat に $\dot{e}0$ を含むあるカテゴリの表現であれば、 Scat にある $\dot{e}0$ を RS に移したのも同じカテゴリである。このような α を α' と略記すると、

$d, c[\alpha']\sigma, e \text{ iff }$

σ を拡張した $\sigma 1$ があって、 $e0_{\sigma} = c(e0)$ かつ $d, c[\alpha]\sigma 1, e$

PrPh Rules

PrPh1 π が Scat に $\dot{b}$ を含まない LRP であるとき、それは PrPh である。

意味特徴、意味は変化しない。

PrPh2 π が Scat に $\dot{b}$ を含む LRP であり、 β が NP であるとき、 $(\pi\beta)$ は PrPh である。

$\text{Scat} = \text{Scat}(\beta)$

$\text{Var} = \text{Var}(\beta)$

$\text{AF} = \text{AF}(\beta)$

$d, c[(\pi\beta)]\sigma, e \text{ iff }$

σ を拡張した以下のような $\sigma 1$ がある。

$d, c[\pi]\sigma 1, e$ かつ $d, c[\beta]\sigma 2, e$ かつ $a_{\sigma 2} = b_{\sigma 1}$

PrPh3 π と $\pi 1$ がともに PrPh であるとき、 $(\pi \text{ who } \pi 1)$ は PrPh である。

$\text{Scat} = \text{Scat}(\pi) \cup \text{Scat}(\pi 1)$

$\text{Var} = \text{Var}(\pi) \cup (\text{Var}(\pi 1) - \text{AF}(\pi))$

$\text{AF} = \text{AF}(\pi) \cup \text{AF}(\pi 1)$

$d, c[(\pi \text{ who } \pi 1)]\sigma, e \text{ iff }$

c を拡張した以下のような $c 1$ がある。

$d, c 1[\pi]\sigma, e$ かつ $d, c 1[\pi 1]\sigma, e$

NP Rules

NP1 すべての IT は NP である。意味と意味特徴は変化しない。

NP2 α が NP であり、 Xi が変項であるとき、 Xi を $\text{AF}(\alpha)$ に加えることができる。

このような α を $\{\alpha\}1$ とすると、

それ以外の意味特徴は変化なし

$d, c[\{\alpha\}1]\sigma, e \text{ iff } c$ を拡張した $c 1$ があって $d, c 1[\alpha]\sigma, e$

$c 1(Xi)$ が定義されていれば $c 1(Xi) = a_{\sigma}$

NP3 δ が DET であり、 π が PrPh であれば、 $(\delta\pi)$ は NP であり、 $\text{Scat}=\{\dot{a}, \dot{e}i\}$

i に応じて、これを $(\delta\pi)i$ と記述すると、

$d, c[(\delta\pi)i]\sigma, e \text{ iff}$

σ を拡張した以下のような $\sigma 1$ がある。

$X = \text{Ext}(\pi, e, d, c, \sigma) \text{ かつ } d, c[\delta]\sigma 1, e$

ここで、 $\text{Ext}(\pi, e, d, c, \sigma)$ は以下のように定義される π の外延である。

$\text{Ext}(\pi, e, d, c, \sigma) = \{a_\sigma \mid d, c[\pi]\sigma 1, e \text{ } a \text{ 以外で、} \sigma \text{ と } \sigma 1 \text{ は一致する}\}$

NP4 α と β が NP であり、 β の Scat が $\{\dot{a}, \dot{e}i\}$ のとき、 $(\alpha < \beta >)$ は NP である。

意味特徴はすべて α と β の意味特徴のユニオンとなる。

$d, c[(\alpha < \beta >)]\sigma, e \text{ iff}$

$d, c[\alpha]\sigma, e \text{ かつ } d, c[\beta]\sigma, e$

LRP Rule

α が RS であり、 t が TH であれば、 αt と $\sim \alpha t$ は LRP である。

意味特徴は共に、

$\text{Scat} = \text{Scat}(\alpha) - \{\dot{i}, \dot{t}v\}$

$\text{Var} = 0$

$\text{lense} = \{t\}$

$\text{AF} = 0$

意味はそれぞれ

$d, c[\alpha t]\sigma, e \text{ iff}$

以下のような σ を拡張した $\sigma 1$ がある。

$t v_\sigma = 1 \text{ かつ } d, c[\alpha]\sigma 1, e \text{ かつ } d, c[t]\sigma 1, e$

$d, c[\sim \alpha t]\sigma, e \text{ iff}$

以下のような σ を拡張した $\sigma 1$ がある。

$t v_\sigma = 0 \text{ かつ } d, c[\alpha]\sigma 1, e \text{ かつ } d, c[t]\sigma 1, e$

4.2 Aliass のインプリメント

4.2.1 方針

状況意味論全体はどちらかと言えば「発話の意味がなぜこうなるか」を論じる理論であって、「どう解析すれば発話の意味が得られるか」を論じているものではない。このため、Aliassにおいても、発話状況における関連はgiven として扱われているし、AFなどの意味特徴を付加することによって曖昧さが生じないよう配慮されている。しかし、今回のインプリメントは、次章で述べるQAシステムのベースとなるものを目標としているため、発話解析の立場で行なう。つまり、関連は既に決定しているものではなく、発話は意味の曖昧性を含んだままのものを対象とする。

ただし、Prologの特徴を生かすことで、関連についてはそれが与えられていなくてもプログラム的には入差のないものにできる。つまり、関連が与えられていないときには、その枠組み(Prolog の言葉で言えば、unbindの変数を内容とする構造)を作りだすようにプログラムを設計することで、後の処理も極めて容易なものとなる。

具体的なインプリメントの方針を以下に示す。

- Aliassは意味合成規則がiff の形式で記述されており、基本的には、これをそのままプログラム化することで、システムを実現することが可能である。この特徴を充分生かすために、未定項の表現として、prologの論理変数をあてる。
- 関連は未知であるものとして扱う。先にも述べたようにaliassの出力レベルでは関連はその枠組みだけが与えられ、次章で述べる部分によってこれを同定する。また、発話状況については、これを同定すること（つまり、ある発話がいつどこで誰によってなされたかを決定すること）は本システムで扱う問題ではないととらえこれについては簡単な扱いにとどめる。
- 発話解析を目的とするため、意味特徴の内、Var, Tense, AF, RS は除く。Scatはレキシコンから得られる情報であるため利用する。ただし、未定項にPrologの論理変数を利用する関係からScatは未定項の集合ではなく、setting の定義域を表わすものとして、a, b, l, ... など定数の集合とした。
- 入力としては、インデックスなどを含まない一般的な英文を考え、動詞のtvやINについては、形態素解析によって抽出する手法をとる。

4.2.2 文法とレキシコンの変更

以下の目的で文法とレキシコンに若干の変項を加えた。

- ① 意味標識を利用しないことによる変更、特に関連の扱いに影響する。
- ② 受理可能な英文の範囲拡大のための変更、文法規則の付加となる。

意味標識を利用しないことによる変更

関連の扱い

関連はリストで表現し、その要素は $c(X, Y)$ という形式とした。これで、 $Y=c(X)$ を表現する。 X はそれがIT, TM, RSの内のどの変項であるかに応じて、 $var(A)$, $t-var(A)$, $rs(A)$ の形式をとる。更に X がITの時は、 Y は $B:E$ というロールの形式をとる。

ITの場合を例に関連の扱いを述べる。名詞句の内、変項があらわれた時は、それにインデックス n を割当て、 $c(var(n), X)$ を関連に加える。変項の先行詞となる可能性がある名詞句（つまり、変項と一二人称代名詞を除く名詞句）が現れた場合は関連に $c(var(N), a:e)$ を加える。ここで、 $a:e$ はその名詞句の意味とも言えるロールである。このような処理の必要上、NPのカテゴリを2つに分類した。

このような処理の結果以下のような文では、

Jackie bites molly and she excites

$[c(var(N), jackie:..), c(var(M), molly:..), c(var(1), A:E)]$

という関連が生成される。これを適当にunify することで、正しい関連が得られるわけである。

このような処理の性質上、文法規則中で、行なわれていた関連 c に関するチェックは一切行わず、そのような場面では、適当な関連の要素を関連に加えるという形になる。

この処理を容易にするため、レキシコンも、ITを $d-pro$ (一二人称代名詞)、 $i-pro$ (変項)、 $name$ (固有名詞)の3つに分類した。なお、この分類は文献(2)による

普通名詞の扱い

資源状況についてもITと同様なことが言える。普通名詞が指示的用法で用いられているか、属性的用法で用いられているかは、入力からは判断できないという問題である。(同格的用法は構文構造から判定ができる。)このため、そのような名詞句が現れた場合には、関連に $c(rs(N), E)$ を加え、文が完成した時点で、 $Scat$ に $e1$ を含んでいるものについては、関連に $c(rs(0), E)$, $c(rs(1), e)$ を加えるという処理を行なう。ここで、 e はその発話によって描かれる状況である。その後の処理は先に述べたものと等しい。この必要性から s のカテゴリをふたつに分類した。

普通名詞の扱いにおいては、もうひとつの問題がある。普通名詞に付与される時標識である。Aliassにおけるa dog の解釈は以下になる。

setof(B, in E at I : dog B ;yes, X) ,
is-member-of(A, X)

これは、ある状況E において領域I に犬であるような個体の集合があつて、その集合のある要素がa dog の意味であると読める。ここで、E は資源状況である。I は普通名詞に付与される時標識によって決定される領域である。直感的にはそれがいつ犬であつたかを示している。この時標識は入力となる文には明示されない。犬の場合は、ある領域で犬であれば、Iuで犬であるという制約を考えることで問題を解消できるが、以下に示す例では極めて複雑な問題となる。

(1) 彼は妻を殴った。(それが原因で離婚した。)

(2) 彼は公園で妻にプロポーズした。

(1) の例ではその女性が彼の妻であつたのは、Idにおいてではなく、発話の中の動詞が持つ時標識に関連する領域を含む領域においてであろう。(2) の場合では逆にIdにおいての妻であつて、プロポーズの時点ではそうではないのである。

この曖昧性はaliassの範囲で解消できるものではない。これを曖昧なまま残すため、時変数(t-var)をおき、普通名詞に付与される時標識はすべてこのような変数であるとする、これに関連に加えて、適当なunify を行なうことで、その領域を同定する手法である。

受理範囲拡大のための変更

関係代名詞節の拡張

aliassが扱える関係代名詞節は、以下に示すような主語を関係代名詞で置換したもののみである。

a girl who owns him

これに、以下のような直接目的語を関係代名詞で置換したものを加えた。

a girl who john loves

構文カテゴリとして、主語のない文に相当するPrPh1 (適切な命名ではないかもしれない) 以下の2つの文法規則を付加した。

PrPh1 Rule

β がNPであり、 π がScatに $\dot{b}$ を含むLRP であるとき、 $(\beta \pi)$ はPrph1 である。

$\text{Scat} - \text{Scat}(\beta) - \text{Scat}(\pi)$

$d, c [(\beta \pi)] \sigma, e \quad \text{iff}$

σ を拡張した以下のような $\sigma 1$ がある。

$d, c [\beta] \sigma 1, e \quad \text{かつ} \quad d, c [\pi] \sigma 1, e$

PrPh4

π が PrPh であり、 π_1 が PrPh1 であるとき、 $(\pi \text{ who } \pi_1)$ は PrPh である。

$\text{Scat} = (\text{Scat}(\pi_1) - \{\dot{b}\}) \cup \{\dot{a}\}$

$d, c[(\pi \text{ who } \pi_1)] \sigma, e \text{ iff}$

σ を拡張した以下のような σ_1 がある。

$d, c[\pi] \sigma_1, e \text{ かつ } d, c[\pi] \sigma_2, e \text{ かつ } a_{\sigma_2} = b_{\sigma_1}$

所有代名詞の付加

his dog や his father など所有代名詞を含む名詞句を加えた。これらについては形態素解析的な変型によって扱った。

一項関係で表わされる名詞(dog, cat) は以下のように変型する。

his dog $\rightarrow$ a dog t-var who he own t-var

二項関係で表わされる名詞(father, friend) は以下のように変型する。

his father $\rightarrow$ a father t-var him

father は $\text{Scat} = \{\dot{a}, \dot{b}, \dot{i}, \dot{iv}\}$ の RS であるから、father t-var him は $\text{Scat} = \{a\}$ の PrPh となる。

このような処理と動詞の形態素解析を簡単にするため、レキシコンに普通名詞のカテゴリ cn を設けた。

4.2.3 データ構造など

文法とレキシコンの記述は DCG で行なった。各構文カテゴリは下に示すような arty3 の述語である。

P 構文木生成のための構造

sf 以下の5要素からなるリスト

Sem $d, c, [\alpha] \sigma, e$ を表現する論理式、以下で詳説する。

D 発話状況を表わす4要素のリスト [Ld, Ad, Bd, U]

C 関連 前述

s setting σ を表わす変項のリスト

E 描かれた状況を表わす変項

Scat Scat を表現する以下のサブリスト [a, b, i, tv, x, e0, e1]

これは S の定義域を決めるもので、s と Scat の要素が対応する。

Sem はD, S, E の関係を表すもので、Prologの項によって表現される。基本的な述語を以下に示す。

is-in([L, R, A, B, T], E)

in E : at L : R, A, B;T を表現する。

setof(A, Event, X)

Event を満すA の集合がX {A|Event} の外延がX

is-member-of(A, X)

集合X の要素がA

is-the-unique-member-of(A, X)

集合X は要素A のみからなる

これらを',' と';' で結合したものがSem となる。

レキシコンの例を以下に示す。

rs(rs(love), [is-in([L, love, A, B, Tv], E), D, [], [A, B, L, Tv], E], [a, b, l, tv])-->[love].

cn(rs(dog), [is-in([L, dog, A, Tv], E), D, [], [A, L, Tv], E], [a, l, tv])-->[dog].

i-pro(it(i), [Ad=A, [ld, Ad, Bd, U], [], [A], L], [a])-->[i].

d-pro(it(he(N)), [(S, is-in([Ld, male, A, 1], E)),

[Ld, Ad, Bd, U], [c(var(N), A:S)], [A], E], [a])-->[he], {next-num(N)}.

name(it(jackie), [(A=jackie, is-in([Ld, female, A, 1], E)),

[Ld, Ad, Bd, U], [], [A], E], [a])-->[jackie].

det(det(a), [is-member-of(A, X), D, [], [A, X], E], [a, x])-->[a].

ここで、next-num(N) は呼ばれるたびに新しい自然数を生成する述語である。

文法規則の例として、s1に対応するDCG 表現を下に示す。

s(s(P1, P2), [(Sem1, Sem2), D, C, S, E], Scat)-->

np(P1, [Sem1, D, C1, S1, E], Scat1),

prph(P2, [Sem2, D, C2, S2, E], Scat2),

{union(Scat1, Scat2, Scat3), difference(Scat3, [a], Scat)},

{equal(C1, C2, C)},

{equal(S1, Scat1, S2, Scat2, S3, Scat3), extend(S, Scat, S3, Scat3)}.

ここで、

union/3,difference/3は、リストを集合と見做して、各々、第 3項が第 1,2項の和集合、差集合となるものである。

equal/3 は前に述べた理由から一般的なappend/3である。

extend/4は定義域ScatであるS を定義域Scat3 に拡張したものがS3であるとき成功する述語である。

equal/6 は、S1とS2がScat1 とScat2 の重複した部分でunify 可能であるとき成功する述語で、unify された結果のS1,S2 をScat3 に拡張したものがS3となる。これは、aliasの規則におけるS1=S2 を表している。

このように、Aliasの文法規則をそのままプログラム化したイメージとなる。

左再帰的な規則については、DCG で直接表現することが不可能であるが、dummy(X,Y,X,Y)という述語を用意し、プログラム上で、入力文字列を取り出して、これに操作を加えることで、無限ループに陥ることを防いでいる。

形態素解析も、dummy/4 をもちいて入力文字列を取り出して、その変形を行なっている。例えば、

```
likes --> like,present  
didnt like --> not,like,past
```

変形の手法は文献(7) のものを採用した。

インプリメントしたAliasの入出力例をFig.1 に示す。

この出力は関連を適当に評価した後に得られるものであり、本当の意味での出力はもう少し、複雑である。

特に、ex.2で、jackieが寝ていないときも犬であることに注意してもらいたい。これは、関連というものを発話の意味に組み込んだことによる状況意味論独特の特徴である。

5. QAシステムへの応用

5.1 システム概要

本システムは、平叙文、疑問文を入力として許し、平叙文の場合はそれによって描かれる状況を作りだし、疑問文の場合は、状況に照してその質問に答えるというものである。

基本的な考え方として、発話の理解を発話そのものによって描かれる状況とそれ以前に述べられた状況からfactualな状況、つまり、状況構造によって得られる意味ある付加をふくめて整合的な状況を生成してゆくこととらえている。この様子をFig.2に示す。

システムのブロックダイアグラムをFig.3に示す。

平叙文の場合、前章で述べたAliassの出力として、発話の意味を表現している論理式と評価されていない関連を得る。この関連を適当に評価し、論理式から状況を生成する。これと状況構造から意味ある付加を生成し、これらがそれまで述べられてきた状況と整合的であるかのチェックを行ない、整合的であれば、その状況を過去の状況に加えてこのループを繰り返す。整合的でない場合は、バックトラックにより、別の方法で関連を評価し、整合的な状況が得られるまで、generation and testを反復する。

疑問文の場合、疑問文のためのalias (以下q-aliasと略記)の出力として、論理式と評価されていない関連を得る。この関連を適当に評価し、論理式から状況を生成する。これと状況構造から意味ある付加を生成する。yes-no型の疑問文の場合、この状況がこれまで述べられてきたものと整合的であればyesを非整合的であればnoを返す。wh型の疑問文の場合には、この時点で得られるのは、一種のロールであり、これをそれまでに述べられた状況を文脈としてアンカリングを行ない、得られた結果を解答とする。

状況は事態に対応する以下の要素からなるリストで表現している。

$[l, r, a, b, t]$ もしくは $[l, r, a, t]$

それぞれの意味は、

$at\ l : r, a, b; t$ $at\ l : r, a; t$

である。状況意味論からの当然の帰結として、リストの要素となっていないものについては、それが含まれるか否かについては何も述べていない。

本システムでは、状況に加えて、そこに登場している個体のリスト(id-list)を持っている。これは、代名詞同定の差異に利用されるもので、A:[で表現されるロールからなるリストである。ただし、一般のロールとは異なり、aは既にアンカリングが済んでおり、未定項ではない。また、a dog など名前を持たない個体については、aはシステムによって用意されたid-no.が付与されている。

例として、次の文が最初に入力されたときの状況とid-listを示す。

```
入力      jackie owns a dog
状況      [[present, own, jackie, id(1), 1],
           [present, female, jackie, 1],
           [present, dog, id(1), 1]]
id-list   [[id(1):...],
           [jackie:is-in([presnt, female, jackie, 1], E)]]
```

5.2 QAシステムにおけるAliass

5.2.1 Aliassの制限

平叙文を解析するAliassは前章で述べたAliassに以下の制約を加えたものである。これらは、主に代名詞同定などその後の解析から来る制約である。

選言による文の連結は対象から除く。

jackie barked or she sneezed.

という入力はdeterministic な解釈ができない。これによって、生成される状況はスキーマとなり、そのような状況に対して、

did jackie sneeze?

という質問が来た場合、その解答が困難となる。

領域の限定

領域としては、Idを含むpresent とそれに先行するpastという2つのみを考えることとする。また、時標識は、そのどちらかを直接指定するものとする。つまり、状況の中に1 o Id などと記述することはやめ、I=present とunify するものとする。

資源状況の限定

名詞が指示的用法で用いられたとき、その資源状況は、それまでに描かれた状況全体とする。属性的用法や同格的用法で用いられたときの資源状況は当然その名詞を含む発話によって描かれる状況である。

the の扱い

定冠詞のthe については、上に述べた資源状況に対する制約からその扱いに変更を加える。発話中にthe dog と言われた場合、その資源状況に犬は一匹しか存在してはならないという解釈となるが、このままでは、それまでに描かれた状況に複数の犬が存在した場合整合的な解釈が存在しないことになる。これを防ぐためにthe の扱いは不定冠詞a の扱いと同じにする。現実には、それまでに述べられた状況の一部が資源状

況となっているのでこのような問題は生じないのであるが、そのためには資源状況を正確に決定するという問題が生じてくることになる。この問題の解決はかなり困難であると考えられる。また、このように the を扱うことによって Aliass によって生成されるすべての状況が強固であるという利点が生まれる。

5.2.2 疑問文解析のための Aliass

疑問文解析のための Aliass(以下 q-alias と呼ぶ)について述べる。

疑問文の意味表現は、状況意味論から得られるものを若干簡単化して以下のように行なう。

yes-no 型疑問文

入力文に対応する平叙文が持つものと同じ。

does jackie own a dog? の意味は

jackie owns a dog. と同じ意味表現を持つ。

実際には、これによって、描かれる状況が整合的か否かを発話者が尋ねているというイベントが加わるべきであろうが、それはプログラムとして実現し、意味表現には含まない。

wh 型疑問文

疑問代名詞を未定項と考え、入力文に対応する平叙文が持つ意味に、request(X)(X は疑問代名詞に対応する未定項)を加えたものを意味表現とする。

who loves jackie? の意味は、

X loves jackie. の意味表現に request(X) をくわえたもの

これは、request(X)によって、発話者がそこにアンカリングされる個体を要求しているというイベントを表現していると考えたものである。

q-alias は以下のレキシコンと文法規則を持つ。

レキシコン

Aliass に以下の単語カテゴリを加えた。

q-pro 疑問代名詞

Scat={ $\dot{a}$ }

$d, c[\alpha] \sigma, e \quad \text{iff} \quad \text{request}(A)$

ただし、whose は、who の所有格として扱った。

q-tm Tense-marker 代動詞

平叙文 Aliass においては、TH は動詞の形態素解析によって得られるものとしたが、q-alias においては、代動詞を TH にあてる。意味は TH に同じ。

文法規則

以下の文法規則からなる。Aliassと重複する部分は説明を省いた。

question Rules

q1 α は q-pro で π が PrPh であるとき、 $(\alpha\pi)$ は question である。

意味特徴、意味ともに S1 に同じ。

q2 α が q-pro で π が PrPhq であるとき、 $(\alpha\pi)$ は question である。

$\text{Scat} = (\text{Scat}(\pi) - \{\dot{b}\}) \cup \{\dot{a}\}$

$d, c[(\alpha\pi)]\sigma, e \quad \text{iff}$

σ を拡張した以下のような σ_1 がある。

$d, c[\alpha]\sigma_1, e$ かつ $d, c[\pi]\sigma_2, e$ かつ $a_{\tau_2} = b_{\sigma_1}$

q3 π が PrPhq で α が NP であるとき、 $(\pi\alpha)$ は question である。

$\text{Scat} = (\text{Scat}(\pi) - \{\dot{b}\}) \cup \{\dot{a}\}$

$d, c[(\alpha\pi)]\sigma, e \quad \text{iff}$

σ を拡張した以下のような σ_1 がある。

$d, c[\alpha]\sigma_1, e$ かつ $d, c[\pi]\sigma_2, e$ かつ $a_{\tau_2} = b_{\sigma_1}$

PrPhq Rules

PrPhq1 t が q-tm であり、 α が NP であり、 π が Scat に b を含む RS であるとき、

$(t\alpha\pi)$ は PrPhq である。

$\text{Scat} = \text{Scat}(\pi) - \{\dot{t}, \dot{v}\}$

$d, c[(t\alpha\pi)]\sigma, e \quad \text{iff}$

σ を拡張した以下のような σ_1 がある。

$d, c[t]\sigma_1, e$ かつ $d, c[\pi]\sigma_1, e$

かつ $d, c[\alpha]\sigma_1, e$ かつ $t \vee_{\sigma} v = 1$

PrPhq2 π が RS であり、 α が NP であるとき、 $(\pi\alpha)$ は PrPhq である。

このルールは be 動詞の疑問文のためのもので、 π は be 動詞でなければならない。

意味その他は PrPhq1 に同じ。

関連の扱いはAliasとほぼ同じである。ただし、疑問文における普通名詞の用法は指示的用法に限定することができるので、資源状況の扱いは多少異なる。

5.3 状況構造

制約は以下に示すようなPrologのfact形式で表現した。

```
involve([[L,child,A,B,1],[L,male,B,1]],[L,father,B,A,1]).
```

ある領域で、A がB の子供であり、B が男であれば、その領域でB はA の父である。

```
involve([[L,girl,A,1]],[L,female,A,1]).
```

ある領域でA が少女であれば、その領域でA は女である。

```
preclude([[L,R,A,B,1],[L,R,A,B,0]],true).
```

```
preclude([[L,'=',A,A,0]],true).
```

状況構造が整合的であることからくる。

```
preclude([[L,male,A,1],[L,female,A,1]],true).
```

ある領域でA が男であり、かつ、女であることはない。

```
preclude([[L,father,A,B,1],[L,father,C,B,1]],not(A=C)).
```

ある領域で、B の父はひとりである。

involve は第1項の事態がすべて状況の中にあれば、第2項の事態もその状況に含まれるという意味である。precludeは第1項の事態のすべてが状況の中であって、かつ第2項の条件を満たしているとき、その状況が整合的でないことを示している。第2項の条件はPrologによるインプリメントの関係で事態の形式で記述できなかったもので、trueは無条件を表わす。制約におけるスキーマは実現しなかった。

システムにおいて、involve は意味ある付加を求める一種の推論規則として用いられ、precludeは、得られた状況が整合的であるかのチェックに利用される。

状況意味論に忠実に従うとすれば、precludeとすべきものは上例の内の最初の2つのみであり、それ以外はinvolve で記述すべきものである。例えば3番目の例は、

```
involve([[L,male,A,1]],[L,female,A,0]).
```

と記述できる。しかし、このようにした場合、例えばjackieが犬であるという状況から、彼女は猫でなく、少女でなく、少年でもないというような莫大な状況が意味ある付加として導かれてしまう。これは、探索空間を無闇に広げることになって好ましくない。このため、このような一部の制約は、チェックにのみ利用し、状況に加えないようにしている。

5.4 関連の評価と整合性チェック

5.1で述べたように、関連の評価と整合性チェックはgeneration, test の関係となっている。つまり、関連の評価は整合的な状況を生成するまで、バックトラックによって反復される。関連の評価は以下の順序でおこなわれる。

資源状況

名詞の用法は、まず、指示的用法であるとする。

それによって、整合的な状況が生成されないときは、属性的用法とする。

時標識

普通名詞に付与されている時標識は、まず、それを含むPrPh内の時標識と一致させる。

代名詞（変項）

代名詞は、文内を前方に調べ、まず最も近いものと一致させる。

文内に整合的な状況を生成するような個体があれば、id-list を最近のものから順に調べる。

バックトラックの際は、この逆順にalternative が捜される。この順序、特に、名詞を属性的用法と見做すのを最後にすることは重要である。a dog が属性的用法で用いられるということは、描かれる状況に新しい犬という性質を持った個体が増えることに他ならない。この新しい個体については、何が述べられていてもその状況は整合的になってしまう。以下の例で考えてみる。

jackie owns a dog.
the dog is her pet.

すべてのものは自分のペットになれないという制約があれば代名詞her が指しているのは、a dog であるという解釈がでてくる。この解釈でも、a dog とthe dog が異なる個体であれば、つまり、the dog が属性的に用いられていれば整合的である（犬はペットを持たないという制約があれば別であるが）。しかし、すなおな解釈は、her=jackie, the dog=a dog であろう。前者のような解釈を防ぐために、このような順序が要求されるのである。

この様にして行なわれた関連の評価に続いて、整合性のチェックが行なわれる。これは、先のinvolve を用いて、これまでの状況と得られた発話の解釈から帰結される意味ある付加を加え、それら全体がprecludeに示されている制約を満たすかをチェックするわけである。本システムの入出力例をFig.4 に示す。

5.5 問題と今後の課題

代名詞同定について

現在行なっている代名詞同定処理は、0次近似であって、極めて単純なものである。
例えば以下の例を考えてみる。

jackie loves molly.

she owns a dog.

この場合、現在のシステムで得られる解はshe=molly である。人によって解釈が異なるかもしれないが、やはり、she=jackieの方が自然であろう。なぜなら、現在のトピックはjackieだからである。

また、代名詞同定を含む関連の評価にgeneration and test を用いているのは効率の面からも問題が多い。問題の性質上、基本的にはそのような手法に頼らざるを得ないとしても、早く解に達するためのheuristicsの導入が必須であろう。そして、そのheuristic には、現在のトピックによって探索順序を変更するなどの処理を組み込むことが必要であろう。

be動詞の扱い

現在、be動詞の意味はふたつの個体が等しいという関係にとらえて、一般の動詞と同様に扱っているが、幾つかの問題がある。まず、be動詞の扱いを説明する。

jackie is a dog.

という発話が入力された場合、その解釈は、まず、

[[present,=,jackie,A,1],

[present,dog,A,1]]

となる。jackie loves a dog などの場合には、このA はシステムによってidを与えられるが、be動詞の場合は、A とjackieがunify され、[present,=,jackie,jackie,1] は状況から削除される。

問題が生じるのは否定の場合で、その解釈は、

[[present,=,jackie,A,0],

[present,dog,A,1]]

となってしまう。状況の中に例えばmolly という犬がいた場合は、その解釈は、

[present,=,jackie,molly,0]

となるし、そうでない場合もシステムによってidを与えられ、

[[present,=,jackie,id(1),0],

[present,dog,id(1),1]]

となる。どちらにしろ、望ましい解釈

[present, dog, jackie, 0]

は得られない。be動詞を一般動詞と同じ扱いにしたこと、もしくは、a dog の意味を個体に対応するものととらえたことによる問題であると考えられるが、a dog という普通名詞句の意味がその外延の要素として与えられること（これは、compositionarityから言えば、その前にどのような動詞がきても影響されないはずである。）を考えると重要な問題であると思われる。

6. 考察

以上、状況意味論の説明から始め、それを背景にした簡単なQAシステムの試作について述べた。本章では、談話理解と状況意味論という観点から多少述べる。

状況意味論は、その基本的な部分から「効果のある文」を対象にしている。そして、その効果が最も顕著に現れるのは、言うまでもなく談話という状況においてである。このことは、状況意味論がまさに談話理解のための意味論であることを示唆していると考えられる。例えば、代名詞同定の問題は談話状況と関連の同定という形でとり込むことができるし、focusの問題は資源状況の変化という形で説明できる。更に、speech actやspeaker intensionの問題は談話状況をより詳しく述べることで取り扱えると考えられる。speech actの問題については、発話が状況を変化させるものととらえる形で既に研究がなされている。(3) 先にも述べたように状況意味論は決してこれらの問題の解決を与えてくれるものではないが、それらの処理がひとつの理論の枠組みの中で扱えるということは、統一的な談話理解のための基礎となるに十分なものであると考える。

更に状況構造という形で、世界の構造を与えていることも、談話理解システムに必須である推論規則の扱いに理論的裏付けを与えてくれると考えられる。ただし、これについては、すべてが制約という平板かつ単純なルールの形式ですむかという問題があり、システムのインプリメントの際にはより構造化された表現が必要になってくるのではないと思われる。また、先にふれたspeech actの問題は状況構造の問題としてもとらえることができる。例えば、「ちょっと暑いね」という発話が「窓を開けて下さい」ということを意味するというのは両者が意味ある関係にあるということに他ならない。

また、モンタギュ文法で問題となるreferenceの問題や可能世界にまたがる個体同定についても状況意味論はひとつの解答を与えている。文献(4)では以下のような例で状況意味論の有効性を論じている。

Hob thinks that a witch has blighted Bob's mare and
Nob believes that she(the same witch) killed COB's sow.

状況意味論におけるattitude reportの扱いはその後にくる文に別の資源状況を割りあてるというもので、この資源状況が状況構造にたいしてfactualでなければそれは誤った信念ということになる。これによって、素朴意味論によって生じる問題を逃れている。さらに、factualでない資源状況であってもその素材は現実の世界から取り出されたものであるから、モンタギュ文法で問題となる可能世界にまたがる個体同定の問題も生じないことになるのである。

しかし、状況意味論にも幾つか問題は残されているように思う。そのひとつとして感じるのは、数学的取り扱いが厳密性に欠けるのではないかという点である。状況意味論の数学的基礎はKPU と呼ばれる集合論であるが、ここでは、集合が自分自身の要素となることを許していない。そのことは制約としても明記されてはいるのであるが、所々それが破られているのではないと思われる点がある。もっとも状況意味論はまだ生まれたばかりの理論であると言ってよいだろうから、そのような問題は徐々に解決されていくことであろう。

ともかく、本理論が新しく、未完成な部分が多いとしても、それが非常に大きなポテンシャルを有していることは疑いのないことに思われる。

7. おわりに

状況意味論は新しい意味論でありながら、既に多くの研究者の関心を引いているようで、その簡単なインプリメントも幾つかはなされている。(5),(6) 私もその一人であるが、まだ勉強が足りず、この理論全体を理解しているとは言いがたい状態である。しかし、この理論の一部を垣間見ただけであっても、状況意味論が対象としている範囲の膨大さは充分感じられる。今後、状況意味論が理論の上でどのように発展してゆくかを想像することはむずかしいが、少なくとも、計算機による言語理解の分野におけるこの理論の役割は、今まで独立のものとして扱われていたspeaker intention やspeech actなどの問題を統合するところにあるのではないかと思われる。残念ながら、このような立場での状況意味論のインプリメントはまだ行なわれていないようであるが、そのようなシステムの提案も間近かなのではないだろうか。

謝辞

終始御指導いただいた横井俊夫第3研究室室長に感謝いたします。また、日頃から活発に討論を交していただいた自然言語研究グループの皆様（向井国昭氏、三古秀夫氏、平川秀樹氏、安川秀樹氏、田中裕一氏）をはじめとする第3研究室の皆様にも感謝いたします。

また、本レポートはICOTでの3か月の研修の成果であり、このような貴重な機会を与えて下さいましたICOTならびにNTTの関係者の皆様にも感謝いたします。さらに、私のような若輩を温かく迎えて下さったICOTの皆様にも感謝いたします。

参考文献

- (1) Barwise.J, Perry.J : Situatuon and Attitude MIT Press 1983
- (2) Perry.J : Fragment of Situation Semantics
 Lecture note of COLING 1984
- (3) Evans.D.A. : A Situation Semantics Approach to The Analysis of Speech Act
 19the Annual meeting of the ACL 1981
- (4) Turner.R : HOB, NOB, BOB, COB and Situation Semantics
 Univ of Massachusetts Occasional paper in linguistics vol. 8
- (5) 向井, 他 : 状況意味論における計算モデルとPrologによる実現に向けて
 ICOT TH-0051 1984
- (6) 鈴木 : 日本語文の意味の状況意味論的な記述
 自然言語処理 42-3 1984
- (7) 安川, 他 : Prolog による形態素解析と熟語処理について
 自然言語処理 32-4 1982

```
! : i love you.          <---- input sentence
```

```

s          <---- parsing tree

```



```
sentence = [i,love,you]
```

```
sem = <---- interpretation (logical form)
is_in([_611,love,Ad,Bd,1],_41),
is_in([_611,o,Ld],_41)
```

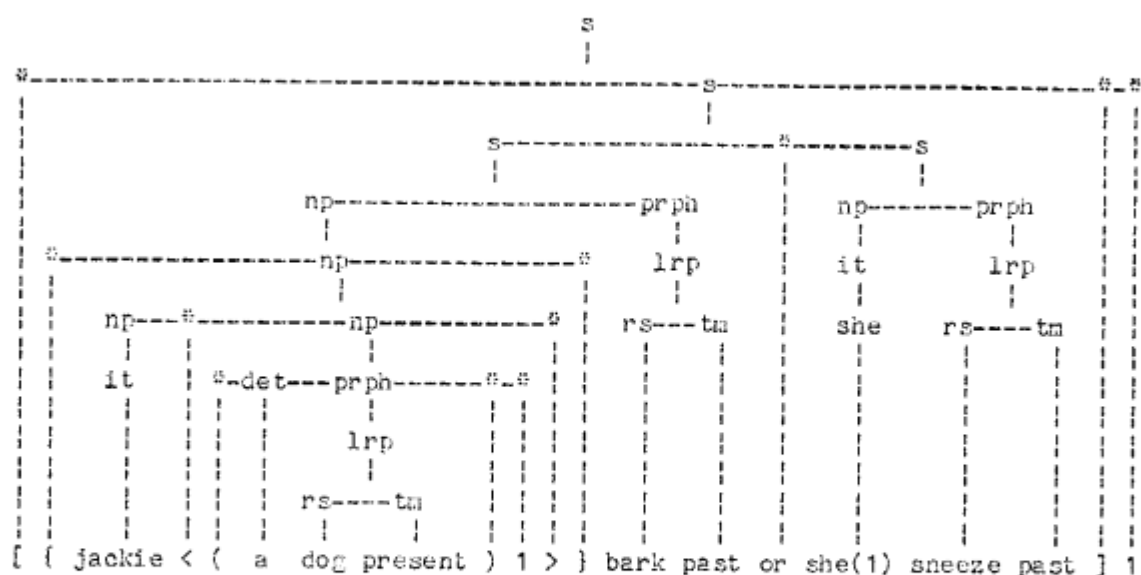
```
d = [Ld,Ad,Bd,U]      <---- discourse situation
```

```
c = [c(present, 511)] <---- connection (evaluated)
```

e = 41 <---- describe situation

Fig.1 Example of execution (part1)

! : jackie, a dog, barked or she sneezed.



sentence = [jackie,,,a,dog,,,barked,or,she,sneezed]

```
sem =
(
  (
    setof(
      _1370,
      is_in([_1495,dog,_1370,1],_41),
      is_in([_1495,o,Ld],_41),
      _1372
    ),
    is_member_of(jackie,_1372)
  ),
  is_in([_2614,bark,jackie,1],_41),
  is_in([_2614,<,Ld],_41)
);
(
  (
    setof(
      _1370,
      is_in([_1495,dog,_1370,1],_41),
      is_in([_1495,o,Ld],_41),
      _1372
    ),
    is_member_of(jackie,_1372)
  ),
  is_in([_2614,sneeze,jackie,1],_41),
  is_in([_2614,<,Ld],_41)
)
```

d = [Ld,Ad,Bd,U]

```
c = [c(var(1),jackie:(setof(_1370,
  (is_in([_1495,dog,_1370,1],_41),
  is_in([_1495,o,Ld],_41)),_1372),
  is_member_of(jackie,_1372))),
  c(present,_1495),
  c(past,_2614)]
```

e = _41

Fig.1 Example of execution (part2)

if the dog loves the boy who owns her.

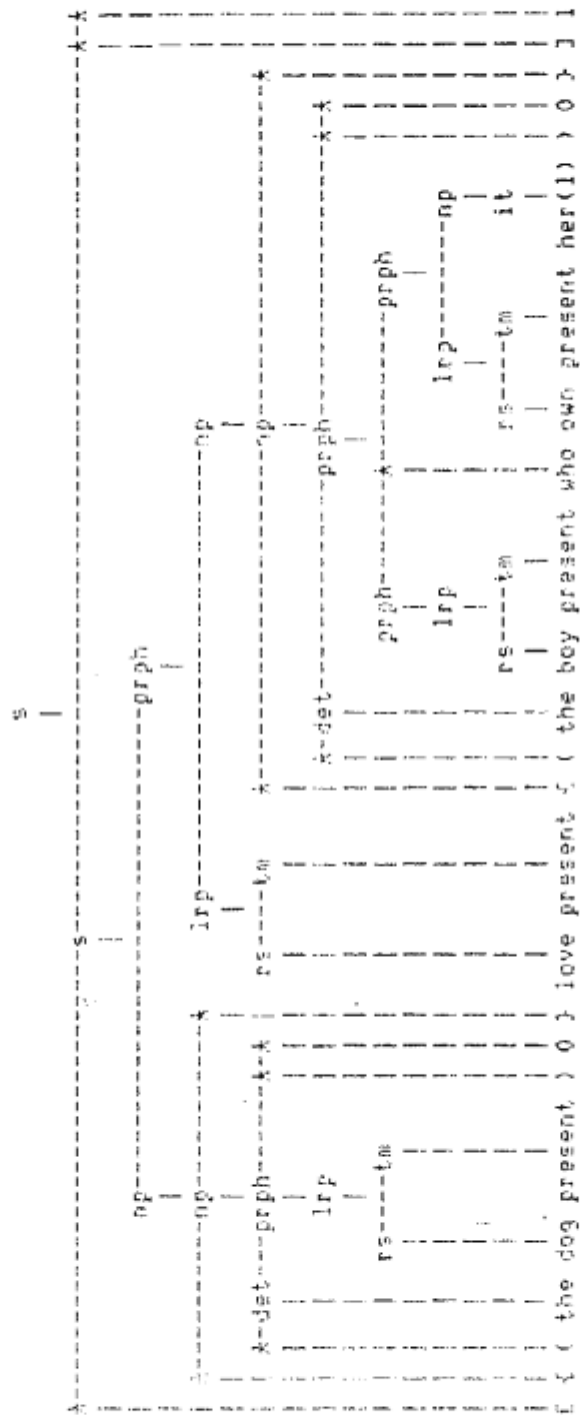


Fig.1 Example of execution (part3)

```

sentence = [the,dog,loves,the,boy,who,owns,her]

sem =
(
  setof(
    _1168,
    is_in([_1293,dog,_1168,1],_1107),
    is_in([_1293,o,Ld],_1107),
    _1170
  ),
  is_the_unique_member_of(_1156,_1170)
),
(
  is_in([_1293,love,_1156,_1939,1],_41),
  is_in([_1293,o,Ld],_41)
),
setof(
  _2818,
  (
    is_in([_1293,boy,_2818,1],_1107),
    is_in([_1293,o,Ld],_1107)
  ),
  (
    is_in([_1293,own,_2818,_1156,1],_1107),
    is_in([_1293,o,Ld],_1107)
  ),
  setof(
    _1168,
    is_in([_1293,dog,_1168,1],_1107),
    is_in([_1293,o,Ld],_1107),
    _1170
  ),
  is_the_unique_member_of(_1156,_1170),
  _2820
),
is_the_unique_member_of(_1939,_2820)

d = [Ld,Ad,Bd,U]

c = [c(var(1),_1156:(setof(_1168,(is_in([_1293,dog,_1168,1],_1107),
                                     is_in([_1293,o,Ld],_1107)),_1170),
                                     is_the_unique_member_of(_1156,_1170))),
  c(present,_1293),
  c(rs(0,_1107)))]

e = _41

```

Fig.1 Example of execution (part4)

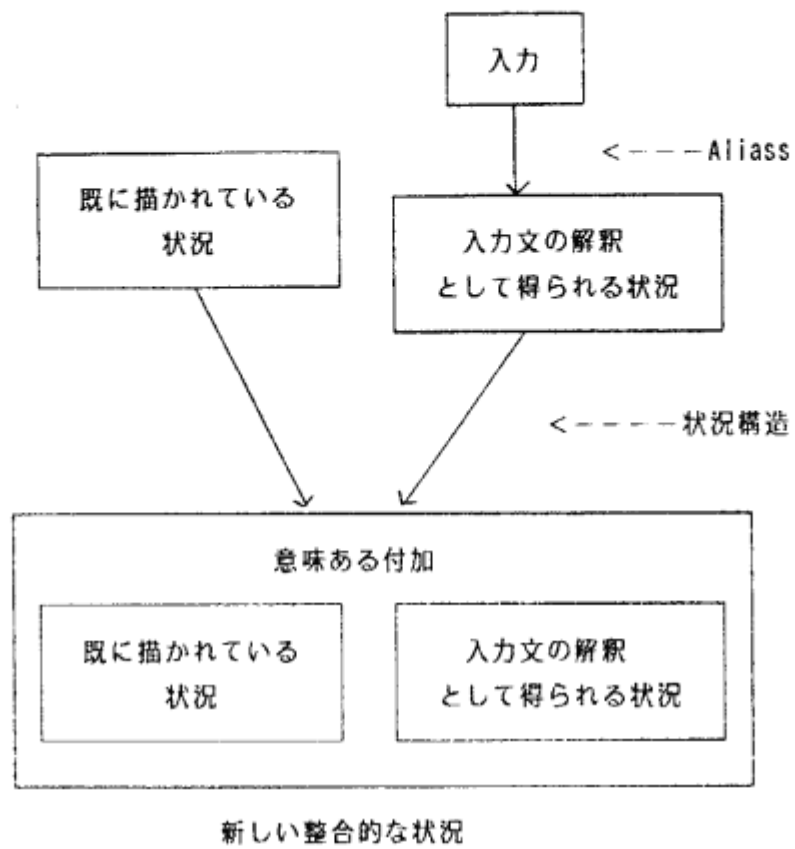


Fig. 2 状況の生成

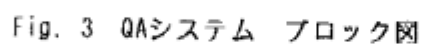


Fig. 3 QAシステム ブロック図

```

! : tom married jackie.          <----- input

[present,female,jackie,1]        <----- situation
[past,marry,tom,jackie,1]
[present,male,tom,1]
[past,marry,jackie,tom,1]
[present,husband,tom,jackie,1]
[present,wife,jackie,tom,1]

jackie:is_in([present,female,jackie,1],new)  <--- individual list
tom:is_in([present,male,tom,1],new)

! : molly,his child,owns a dog.

[present,female,jackie,1]
[past,marry,tom,jackie,1]
[past,marry,jackie,tom,1]
[present,husband,tom,jackie,1]
[present,wife,jackie,tom,1]
[present,dog,id(6),1]
[present,own,molly,id(6),1]
[present,male,tom,1]
[present,child,molly,tom,1]
[present,female,molly,1]
[present,father,tom,molly,1]
[present,mother,jackie,molly,1]
[present,child,molly,jackie,1]

id(6):...
molly:...
tom:...
jackie:...

```

Fig 4 example of question answering (part 1)

! : the dog loves the girl who owns him.

[present, female, jackie, 1]
[past, marry, tom, jackie, 1]
[past, marry, jackie, tom, 1]
[present, husband, tom, jackie, 1]
[present, wife, jackie, tom, 1]
[present, dog, id(6), 1]
[present, own, molly, id(6), 1]
[present, male, tom, 1]
[present, child, molly, tom, 1]
[present, female, molly, 1]
[present, father, tom, molly, 1]
[present, mother, jackie, molly, 1]
[present, child, molly, jackie, 1]
[present, male, id(6), 1]
[present, girl, molly, 1]
[present, love, id(6), molly, 1]

molly: ...
id(6): ...
tom: ...
jackie: ...

! : john likes the dog.

[present, female, jackie, 1]
[past, marry, tom, jackie, 1]
[past, marry, jackie, tom, 1]
[present, husband, tom, jackie, 1]
[present, wife, jackie, tom, 1]
[present, dog, id(6), 1]
[present, own, molly, id(6), 1]
[present, male, tom, 1]
[present, child, molly, tom, 1]
[present, female, molly, 1]
[present, father, tom, molly, 1]
[present, mother, jackie, molly, 1]
[present, child, molly, jackie, 1]
[present, male, id(6), 1]
[present, girl, molly, 1]
[present, love, id(6), molly, 1]
[present, like, john, id(6), 1]
[present, male, john, 1]

john: ...
molly: ...
id(6): ...
tom: ...
jackie: ...

Fig.4 Example of question answering (part2)

```

! : who is mollys mother?

[present,=,_21759,jackie,1]
request(_21759)

Answer = [jackie]

! : whose dog does john like?

[present,like,john,id(6),1]
[present,male,john,1]
request(molly)

Answer = [molly]

! : who loves molly?

[present,female,molly,1]
[present,love,_31465,molly,1]
request(_31465)

Answer = [dog]

! : is jackie tons wife?

[present,female,jackie,1]
[present,=,jackie,jackie,1]

Answer = yes

! : does jackies child own a dog?

[present,own,molly,id(6),1]

Answer = yes

```

Fig.4 Example of question answering (part3)