

## 知識ベースにおける動詞的概念の表現法

佐々木 泰, 岡田 和久, 高田 正之, 小谷 善行

東京農工大学工学部数理情報工学科

自然言語の動詞は各々明確な意味をもつ。こうした動詞の概念を知識ベース中の概念として表現し、さらにその記述自体を推論方法の記述とする方式を提起した。

動詞的概念の定義的意味は、第一に、動作の前後の状態変化として記述する。これは動作概念のインスタンスが、前後の状態概念のインスタンスを推論機構により生成することで実現する。第二に、スロット属性値の制限で記述する。

動詞的概念の記述の上でさらに重要なこととして、その行為から自然に予想される前後や周辺の状況の問題がある。これを解決するために、フレーム強度、リンク強度という確からしさを表す概念を導入した。システムは新しい事象の知識を獲得すると、その周辺にある程度確からしい事象を推論し生成する。また、二つ以上の知識から推論される事象は確からしさが高まる。

### Representation of verb concepts in the knowledge base

Yasushi SASAKI , Kazuhisa OKADA , Masayuki TAKATA , Yoshiyuki KOTANI

Department of Information Science

Tokyo University of Agriculture and Technology

2-24-16, Nakamachi, Koganei-shi, Tokyo, 184, Japan

Each verb of natural language has a specific meaning. We proposed a method of describing concepts of verbs in the knowledge base to be that of inference by itself.

We designed representation of definitive meaning of verb concepts, firstly, as state changing by acting the action. This is implemented as an inference mechanism by which an instance concept of action produces instance concepts of state around it. Secondly, it is represented by constraint of slot values.

There is another type of meaning of verb concepts, which does not define it but supports natural reasoning by the action of the verb. We propose two certainty factors, 'the strength of frame' and 'the strength of link' on our knowledge base Fairy. If the system acquires some information of an event, it produces related frames with the strength stronger than some threshold. Also it strengthens the link of the frame inferred again by another path.

## 1. はじめに

知識は、一般的な概念を表すクラス概念と個々の事実を表すインスタンスに分けられる。本研究は自然言語理解の立場から、動詞的概念を一般的な知識のなかで表現する方法を論じている。

自然言語理解において動詞が重要な意味をもつ。この動詞の表現方法について多くの研究がなされている。それらは動作を構成要素の組合せで表現しようとするもの、動作を機能により分類するものなどがある。しかし、動詞的概念表現を、推論知識として用いるためには多くの問題がある。その一つは、不確かな知識の扱い方への対処である。本研究では、「確からしさ」をフレーム型知識ベースの中に組み込むことにより、これを解決しようとしている。

## 2. 動詞的概念

動作の意味を考える場合、その動作単独では表現しきれず、他の動作と何らかの関連付ける必要がある。例えば、

行く・・・ある所に移動する<sup>(1)</sup>

投げる・・・腕を振って、持っていたものを離れた所へ飛ばす<sup>(2)</sup>

ようになる。前者は動作の特殊化であり、後者は他の動作を用いた言い替えである。本報告では、動作の前後の変化に着目することで動作の表現方法を提案する。

本報告で対象とする「動詞的概念」は次に示すものである。

- ・「行く」「いる」などの文法的には動詞に相当するもの
- ・「レストランで食事をする」といった対応する動詞は存在しないが、特定の動作を表しているもの

後者を動作の階層構造の中で、サブクラスとして表現する。例えば、「食べる」のサブクラスとして、その下位概念に「レストランで食事をする」を表現する。これにより、動詞

的概念を知識ベース中で一貫した捕らえ方をする。

## 3. インスタンスの抽象化

本報告で対象としている「動詞的概念」は知識ベース中ではクラス概念として扱う。一方、インスタンスは個々の事実を表現するものである。われわれが設計している知識ベース Fairy (Frame system for Acquisition and Inference of Representation in a story) では、物語の内容を表現し知識ベースの表現能力を検討している[1][2]。

構成は、各フレームの性格に応じて動作、具象物、場所、時間などのフレームによって表現されている。例えば、「太郎が川から山に行った」は図1のような表現となる。

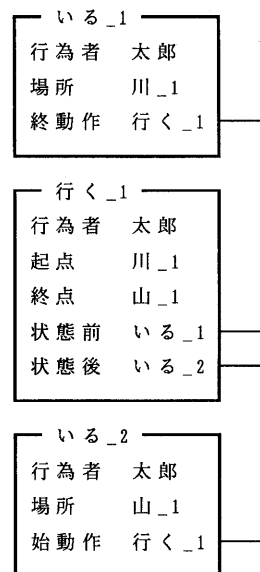


図1 「太郎が川から山に行った」

ここで、図1には文章中に陽に書かれていない「いる」が表現されている。これは「行く」という意味には暗黙的にその前後の「いる」も含まれているからである。

(1) IPAL-辞書編-[3] p.399 ゆく 1【行く, 往く】《意》〈意記〉

(2) IPAL-辞書編-[3] p.282 投げる 1【投げる】《意》〈意記〉

このようにインスタンス設計の際に、  
「文章中に陽に書かれていないことも推論することによって表現する」

という立場をとった。動詞的概念を表現する際にも同様の立場で行い、動作の前後を推論する機構を用いて実現する。

クラス概念の設計を行う際に、既に多量に存在しているインスタンスを抽象化することを考える。例えば、「行く」に関しては図1を抽象化することによって表現する。インスタンスとして多くのデータが既にあり、その構成はほぼ確定している。インスタンスで用いるスロット名、スロット値の属性が決定しているのでクラス概念としては、これらを抽象化することを考える。

「行く」に関して抽象化した結果を図2に示す。

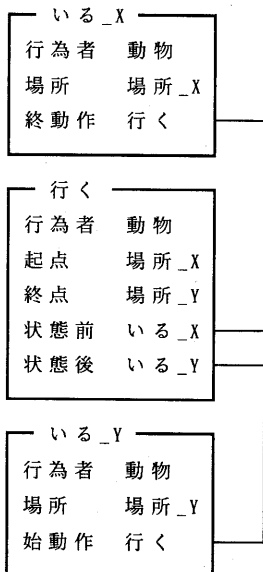


図2 抽象化した「行く」のモデル

図2の「行く」に関してみるとその「行為者」「終点」などのスロット名が決定し、同時にスロット値の属性が決定する。またこれらのスロットと同様に「状態前」「状態後」といった動作の前後も決定する。

「行く」に限らず動詞的概念を表すクラス概念として各フレームがもつべき情報は、

- ・インスタンスに必要なスロット名

・インスタンスのスロット値の属性がある。そこで、これらの情報をフレームとして表現したものを図3に示す。

行く		
行為者	require	動物
起点	require	場所
終点	require	場所
状態前	require	いる
状態後	require	いる

図3 「行く」のクラス表現

ここで require ファセットはスロット値に制限を加えるものである。図3の例は、行為者スロットの値が動物の下位概念であることを限定している。また、状態前、状態後スロットの値がそれぞれ「いるフレーム」の下位概念であると限定している。

図3により図2の「行くフレーム」に関する表現を可能とした。次にその前後の「いるフレーム」に関する情報を記述する必要がある。図3は各フレームとのリンクを表現したにすぎず、そのリンクしたフレームが表現する内容に関する記述は行っていない。そこで各フレームの内容を表現するための手続きとしてデモンを用いる。例えば図2の「状態後」を表すフレームに関しては、

- ・「行く」の行為者スロットの値が「いる\_Y」の行為者スロットの値である
- ・「行く」の終点スロットの値が「いる\_Y」の場所スロットの値である

といえる。これらをデモンを用いて表現する。

先に述べたように知識ベース Fairy は prolog 表記によって実現している。そこで、デモンは prolog の述語名を記述しておき、それを実行することにより表現を可能とした。

「行く」のフレーム表現は以上の議論を踏まえたとえ図4の形として決定する。また実際の prolog 表記のフレーム形式を図5に示す。ここで、フレーム中の第4引数については、次節で述べる。

行く		
行為者	require	動物
起点	require	場所
終点	require	場所
状態前	require	いる
状態後	require	いる
起点	if_added	デモン1
起点	if_added	デモン2
終点	if_added	デモン3
終点	if_added	デモン4

- デモン1：行為者スロットの値を  
状態前の「いるフレーム」の  
行為者スロットに埋めよ
- デモン2：起点スロットの値を  
状態前の「いるフレーム」の  
場所スロットに埋めよ
- デモン3：行為者スロットの値を  
状態後の「いるフレーム」の  
行為者スロットに埋めよ
- デモン4：終点スロットの値を  
状態後の「いるフレーム」の  
場所スロットに埋めよ

図4 「行く」のクラス表現

```

行く(行為者, require, 動物, 100).
行く(起点, require, 場所, 100).
行く(終点, require, 場所, 100).
行く(状態前, require, いる, 100).
行く(状態後, require, いる, 100).
行く(起点, if_added,
    make_ins(行為者, 状態前, 行為者), _).
行く(起点, if_added,
    make_ins(起点, 状態前, 場所), _).
行く(終点, if_added,
    make_ins(行為者, 状態後, 行為者), _).
行く(終点, if_added,
    make_ins(終点, 状態後, 場所), _).

```

図5 「行く」の prolog 形式

以上のように require ファセットとデモンを用いることにより、図2の抽象化したフレーム構成を実際に知識ベース上に実現した。

インスタンスとしてのフレームは、図1のように意味ネットワーク的に各フレームがリンクしている。一方、クラス概念としてのフレームは、モデルとしては図2のようにリンクしているが、実際の表現方法としては各フレームごとに独立している。「行く」に関しては「いる」と関連すると表現するが、「いる」はそれ自体でまた一つのフレームとして存在する。「いる」の行為者スロットの属性などは、「いるフレーム」の中で require ファセットを用いて表現する。

#### 4. 動作の前後

前節では「行く」を例に示した。多量のインスタンスのうち「行く」に関しては比較的概念が明確であり、前後の状態がほぼ一意に決定する。しかし、動作の中には前後が一意に決定しないものも多く存在する。

例えば、「殴る」について考える。インスタンスとしての「殴る」の表現は図6、図7のようになる。

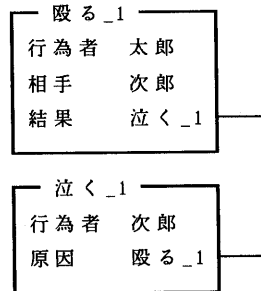


図6 「太郎が次郎を殴った」

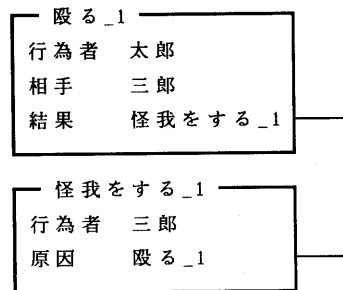


図7 「太郎が三郎を殴った」

図6、図7とも構成はほぼ同じであり‘行為者’‘相手’などのスロット名は決定し、そのスロット値の属性も確定している。これは「行く」の例と同様であるが、異なる点は「行く」のように前後が一意には決定しないことである。図6の例では、「殴った結果、次郎が泣いた」と表現している。図7では、「殴った結果、三郎が怪我をした」と表現している。このように‘原因-結果’といった動作間の因果関係は特定できるものではない。それぞれ、各動作が行われた際の状況によって異なることが多い。図6では結果として「次郎が泣く」となるが、これは次郎が「泣き虫」であるという知識も関係してくる。つまり、動作単独の意味ではなく、具象物等を含めたより多くの知識に裏付けされたものである。

「殴る」についても「行く」と同様にインスタンスを抽象化することを考える。動作の前後に関しては一意には決定しないが候補として記述する。図8に「殴る」のクラス表現を示す。

殴る		
行為者	require	動物
相手	require	動物
結果	require	泣く
結果	require	怪我をする
相手	if_added	デモン1
相手	if_added	デモン2

- デモン1： 相手スロットの値を  
結果の「泣くフレーム」の  
行為者スロットに埋めよ
- デモン2： 相手スロットの値を  
結果の「怪我をするフレーム」の  
行為者スロットに埋めよ

図8 「殴る」のクラス表現

図8に示したように動作の前後に着目した際、結果として「泣く」「怪我をする」という2種類の候補を記述した。これは、「行く」と「いる」の関係と同様 require ファセットを用いることにより表現可能とした。

「殴る」の結果として「泣く」「怪我をす

る」といったように一意には決定するわけではない。「殴る」の前後の動作との関連は、

- ・ 怒ったから殴ったのであろう
  - ・ 殴られた相手は泣くだろう
  - ・ 殴られた相手は怪我をするかもしれない
- のように不確かな要素を含んでいる。このような概念を知識ベース上を実現するため、「～だろう」「～かもしれない」といった表現が必要となってくる。

これを実現するため、「確からしさ」という概念を導入した。知識ベース Fairy は prolog 表記で実現しており、個々のフレームは述語として定義する。各節がそれぞれ一つの知識に対応しており、その節に「確からしさ」を加えたものとして表現する。

図9に「殴る」の prolog 形式を示す。

```

殴る(行為者, require, 動物, 100).
殴る(相手, require, 動物, 100).
殴る(原因, require, 怒る, 80).
殴る(結果, require, 泣く, 80).
殴る(結果, require, 怪我をする, 50).
. . . . .

```

注) デモンは省略

図9 「殴る」の prolog 形式

この第4引数の 80、50 が「確からしさ」に相当する。これにより前述の「～だろう」「～かもしれない」といった不確かな要素を表現する。例えば、

殴る(結果, require, 怪我をする, 50).  
という表現により、

「殴った結果、怪我をするかもしれない」を意味する。これにより動作の前後の関係を記述する。その際に、一意に決定しないものについても候補として記述しておく。そこに可能性として 80%、50% といった「確からしさ」を導入する。この「確からしさ」は動作間の関係に限らず‘行為者’‘相手’などの各スロットにも適応できる。ここでは動作の前後の着目しているため‘結果’‘状態後’のスロットが注目される。しかし、本質的なフレーム構造の中では、単なるスロット名で

あり表現方法が特殊なわけではない。従って、

殴る(結果, require, 怪我をする, 50).

殴る(行為者, require, 動物, 100).

のいずれも同様に扱う。それぞれの意味は、

- ・「殴る」の結果スロットの値として「怪我をする」の下位概念に限定する。また、その可能性は 50% である。
- ・「殴る」の行為者スロットの値として動物の下位概念に限定する。その可能性は 100% である。

となる。このように「確からしさ」の導入にあたっては知識ベース全体として一貫した表現方法を可能とした。

動作の中には、「殴る」のように動作の前後が明確に決定しないものと、「行く」のように概念が明確であり前後の状態が決定するものがある。本報告はこれらの動作について、いずれにも対応できる表現方法を提案した。すなわち、

require ファセット

確からしさ

デモン

を用いて動作を表現する。require ファセットと「確からしさ」を併用して使用することにより、スロット名とスロット値を表現し、フレームの構成を決定する。デモンを使用することにより、動作の前後を表現するフレームの内容を表現する。

## 5. 確からしさ

動詞的概念を表現するうえで動作の前後に着目した。実現方法としてデモンを利用し、動作の前後を推論する推論機構を用いた。そのため、不確かな内容を表現する必要が生じ、「確からしさ」という概念を導入した。ここでは、この「確からしさ」について更に考える。

自然言語理解においてインスタンスとして表現する内容には、

- ・文章中に陽に書かれているもの
- ・文章中に陽に書かれていないもの

の大きく 2 種類がある。前者の表現についてはさほど問題ではなく、後者が重要な意味をもって来る。さらに、陽に書かれていないこ

ととは、

- ① 明らかに事実であるため省略しているもの
- ② 一意には決定しないが通常は推論によって理解できるもの
- ③ 明らかに不確実であり、わからないもの

がある。このうち ③ は知識ベースに表現しないことが当然である。① については基準が明らかであるため表現は比較的容易である。② の表現が最も困難である。本報告で述べる「確からしさ」によってこの ② を表現可能とした。

### 5. 1 確からしさの種類

知識ベース Fairy はインスタンスではフレーム同士の意味ネットワーク的な表現となっている。インスタンスとしての「確からしさ」に関して図 10 のような最も単純なモデルを考える。

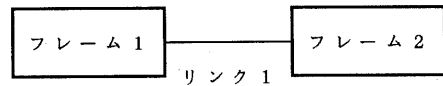


図 10 最も単純なモデル

ここではフレームを 2 つに限り、その間のリンクも 1 つに限定しこれで十分閉じているものとする。このモデルで考えられる「確からしさ」とは、

- ・フレーム 1 の存在の確からしさ
- ・フレーム 2 の存在の確からしさ
- ・リンク 1 の信頼性

がある。「フレームの存在の確からしさ」とは、そのフレームがどの程度確かなものであるということである。通常、フレームは存在するかしないかでしかない。ここでは、動作、状態フレームに関して、実際に行われたことかどうかを表現する意味で用いる。

次に「リンクの信頼性」とは、個々の概念であるフレーム間にどの程度の関係があるかということである。これも通常は、リンクするかしないかでしかない。しかし、動作の前後の関係に着目した場合意味あい異なっ

くる。前例の「殴る」に関して‘原因-結果’  
といったリンクにはどの程度のあるか  
ということを表示する必要がある。

以上の2種類を知識ベースFairy中で  
曖昧性を表現する「確からしさ」として扱う。  
前者のフレームの存在の確からしさを‘フレ  
ーム強度’、後者のリンクの信頼性を‘リン  
ク強度’とそれぞれ名付ける。

## 5. 2 フレーム強度

個々のフレームはそれぞれある概念を表現  
している。フレーム強度はそのフレーム自体  
がどの程度確かなものであるかを意味する。  
通常のフレームでは存在するかないかのど  
ちらかでしかない。例えば、「太郎」「富士  
山」などといったフレームは確かに存在す  
る。これはそれ自体で一つ概念である。ま  
た、「ある山」についてもフレームとしての  
存在は明らかである。

しかし、動作、状態に関しては一概にはい  
えず、「太郎が山に行った」を「行く」とい  
うフレームによって表現する。これが明ら  
かな事実であればそのフレームは存在する。  
そのような事実がなければフレームは存在し  
ない。その中間的な表現として「行ったか  
もしれない」「行ったろう」があり、これ  
を表現するためフレーム強度を用いる。

実際には推論過程において生成された不  
確かなフレームの表現法として使用する。具  
体例として、「太郎が次郎を殴った」から推  
論した「次郎が怪我をした」を図11に示す。

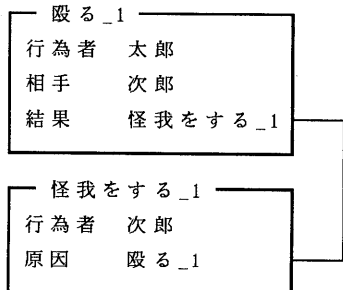


図11 「太郎が次郎を殴った」

ここで、「殴る\_1」は明らかな事実であり  
フレーム強度は100となる。しかし、「怪我  
をする\_1」は推論によって生成された曖昧な  
ものであり、フレーム強度は例えば50とな  
る。この決定については後で詳しく述べる。

## 5. 3 リンク強度

フレーム表現ではフレーム自体が大きな意  
味をもつが、フレーム間のリンクもまた重要  
な意味をもつ。このリンクにどの程度深い関  
係があるかを示したものがリンク強度である。

図11のように「太郎が次郎を殴った」か  
ら「次郎が怪我をした」を推論した場合につ  
いて考える。「殴る\_1」と「怪我をする\_1」  
の間のリンクは非常に重要な意味をもつ。こ  
れは、推論の拠点となるのが‘原因-結果’  
という関係であるからである。

図11で個々のフレームは前述のフレーム  
強度によって、

- ・太郎が次郎を殴ったことは事実である
- ・次郎は怪我をしたかもしれない

ということを表示する。しかし、

- ・殴ったから怪我をした

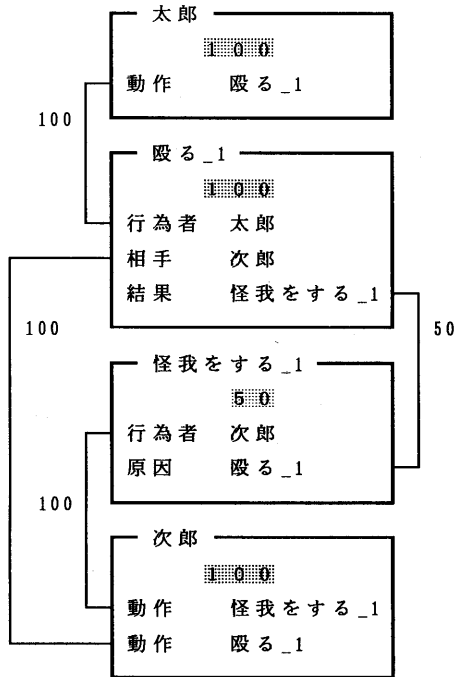
という動作間の関係についてはフレーム強度  
とは異なるものである。‘原因-結果’とい  
うリンクによって因果関係の有無は表現して  
いるが、その信頼性については表現してい  
ない。「殴ったから怪我をした」という関係  
は推論によって導かれたものであり、不確  
かなものである。これをリンク強度によ  
って表現する。すなわち‘原因-結果’  
というリンクに‘確からしさ’を付加する  
ことによって動作間の因果関係を表現す  
る。

このリンク強度もまたフレーム強度と同様  
明らかな事実との間に問題はない。例え  
ば、文章中に「殴ったから怪我をした」と  
明記されていれば、これは明らかな事実  
でありリンク強度を100として表現す  
る。また動作間の関係ではなく、「殴る\_1」  
と「太郎」のような‘行為者-動作’  
というリンクは明らかな事実であるため、  
リンク強度を100として表現する。

このように、新たに‘確からしさ’  
という概念を導入し不確かな表現に  
対して有効な表

現とした。

図 1 2 に「太郎が次郎を殴った」から推論し、「確からしさ」を付加した表現を示す。



注) ■■■■ がフレーム強度を表す

図 1 2 「確からしさ」の表現例

## 6. 確からしさの決定

前節で示したように知識ベース Fairy では「確からしさ」を導入し不確かな要素に関しても表現した。これはフレーム強度、リンク強度の 2 種類を提案した。またこの「確からしさ」はクラス概念、インスタンスの両方がそれぞれもつものである。

クラスとしての「確からしさ」は図 9 に示したように require ファセットと併用することでリンク強度を表現する。

インスタンスとしてのリンク強度はクラスが概念から継承する。これは、インスタンスとしてのスロット値を決定する際にクラス概念を参照することで決定する。require ファセットによってスロット値の属性が決定し、同時にそのリンク強度を継承しインスタンスとし保有する。

一方フレーム強度に関してはインスタンス独自のものである。これはあることがらどの程度確かなものであるかを示すものである。クラス概念として表現するフレームはすべて確かなものであり、個々のフレームが表すことがらは事実である。従ってクラス概念としてのフレーム強度は常に 100 であり、「確からしさ」を考慮する必要がない。インスタンスは個々のフレームについて「確からしさ」を考慮する必要があり、重要な意味をもってくる。インスタンスの「確からしさ」については次のようになる。

- ・リンク強度はクラスから継承することによって決定する
- ・フレーム強度はインスタンス独自のものであり、クラスには依存しない

そこで、次にフレーム強度の決定について述べる。

### 6. 1 フレーム強度の初期値

フレーム強度は不確かなことがらを表現するために導入した概念である。これは推論によって生じた問題である。つまり、推論によって得られる新たなフレームは不確かなものである。

例えば「太郎が次郎を殴った」から推論した「次郎が怪我をした」は不確かなものである。さらに「次郎が怪我をした」から推論すると「次郎が死んだ」となるが、これは一層不確かなものである。不確かなものから推論したものは更に不確かであるといえる。この曖昧性がフレーム強度である。このことから、

- (1) フレーム強度はそのフレームの生成のもととなったフレームのフレーム強度に依存する

といえる。

また、推論の基準となるのがクラスに表現している「動作の前後の関係」である。そこで表現している「原因」「結果」「状態前」などによって推論が行われフレームが生成される。その際に、require ファセットとリンク強度を参照することで動作の前後のフレームを決定する。これが新たに生成されるフレームの基準となる。このことから、



(2) フレーム強度はそのフレームの生成のもととなったフレームとのリンク強度に依存する

といえる。

(1)(2)よりフレーム強度の決定は条件付き確率に相当すると考えられる。そこで確率モデルとしてフレーム強度の決定は、「生成のもととなったフレーム強度」と「そのフレームとのリンク強度」の乗算によるものとする。フレーム強度決定の例を図13に示す。

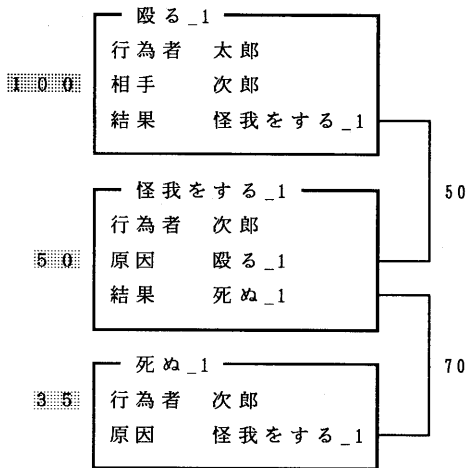


図13 フレーム強度の決定例

ここで各フレーム間のリンク強度は前述のようにクラス概念のデフォルトを継承したものである。

## 6.2 フレーム強度の変更

知識ベース上の「確からしさ」は多くの情報が追加されることにより変更が行われる。リンク強度は文章中に明記されない限りクラスのデフォルトを継承するものである。フレーム強度は新たな事実が追加されることによって不確かであったものが若干確からしいと変化することが考えられる。

不確かなフレームは何らかの推論によって生じたものである。この推論は動作の前後に着目して行っている。その際、多方向から推論が行われ、異なる推論過程によりフレームが生成される。こうして新たに生成されたフ

レームと既に存在するフレームとのマッチングに成功した際には、不確かなことがらであったとしても「確からしさ」は増大する。

例えば「怪我をする」について考える。これを生成する推論経路は一意ではない。「殴る」の結果として推論される場合と、「死ぬ」の原因として推論される場合とが考えられる。これによって違った方向から推論され「確からしさ」が増加する場合を図14に示す。

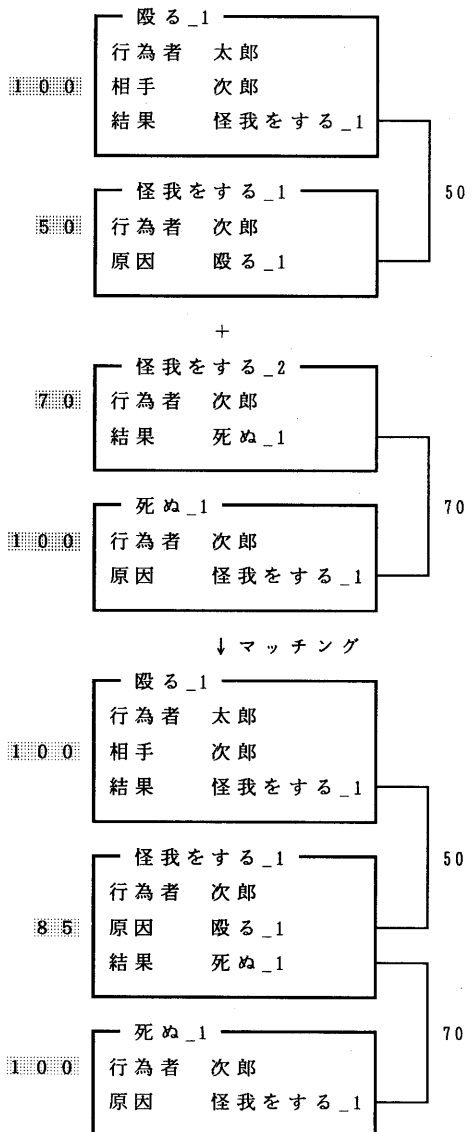


図14 フレーム強度の変更例

図14では「怪我をする」が異なる方向から推論され、そのマッチングによって新たに一つのフレームとして同定されている。その際に、マッチング以前にもっていたそれぞれのフレーム強度から新たなフレーム強度に変更している。この変更は、

$$r = 1 - (1 - p) * (1 - q)$$

r: マッチングによる新たなフレーム強度

p: マッチング以前のフレーム強度

q: マッチング以前のフレーム強度

によるものである。これはそれぞれのフレーム強度の和集合となり、確率モデルとして妥当であると考えられる。

## 7. まとめ

設計したクラス概念を用いて各フレームからのインスタンス生成を行った。また、生成したフレームに対して質問応答を行い知識ベースの妥当性を検討した。

フレーム生成時にデモンが起動し推論が正しく行われた。また、「確からしさ」を付加したことにより知識ベースの表現能力がより豊かなものとなった。

扱った動作は40個であり、使用したデモンは13種類である。動作の前後が明確である「移動」をおもに扱ったが「確からしさ」を導入したことにより、それ以外の動作にも対応した設計とした。

「確からしさ」としてフレーム強度、リンク強度という単純な構造によって知識ベース上に不確かな要素についても表現可能とした。

require ファセットと「確からしさ」を併用し、デモンを用いることによって動詞的概念を知識ベース上に表現した。動作の前後を推論する機構を用いた動詞的概念の表現法を提案した。

## 参考文献

- [1] 佐々木泰、西 明美、高野 啓、  
青島正明、高田正之、小谷善行：  
物語の意味構造を可能な限り記述する実験  
情報処理学会第34回全国大会講演論文集  
6L-5 pp.1351-1652 (1987)
- [2] 佐々木泰、高野 啓、高田正之、  
小谷善行：  
物語知識ベース－概要と適用－  
情報処理学会第36回全国大会講演論文集  
3P-7 pp.1434-1432 (1988)
- [3] 情報処理振興事業協会技術センター：  
計算機用日本語基本動詞辞書 I P A L  
(Basic Verbs)－辞書編－  
情報処理振興事業協会 (1987)
- [4] 村木新次郎：  
述語素による動詞の分類  
情報処理学会研究報告 85-NL-48-5  
(1985)
- [5] 富浦洋一、吉田 将：  
動詞の多義性とその記述について  
情報処理学会研究報告 86-NL-55-2  
(1986)
- [6] 富浦洋一、日高 達、吉田 将：  
語義文からの動詞の階層関係の抽出  
情報処理学会研究報告 87-NL-64-3  
(1987)
- [7] Yorick Wilks：  
A Preferential, Pattern-Seeking,  
Semantics for Natural Language  
Inference  
Artificial Intelligence Vol.6  
pp.53-74 (1975)
- [8] Richard Alterman：  
A Dictionary Based on Concept  
Coherence  
Artificial Intelligence Vol.25  
pp.153-186 (1985)