

# 状態依存プラン解析

小 暮 潔<sup>†,☆</sup> 島 津 明<sup>†,☆☆</sup>

本論文では、行為の効果と前提条件を計算し、先行行為の効果により生じる状態変化に依存してプランを解析する手法を提案する。本手法は状態に基づく行為記述を許し、行為間の可能化関係をとらえ、行為列の状態依存解釈のために必要な情報を得る。本手法は統語解析技術、特にチャート法に基づく。状態変化を取り扱うために、チャート法で使用する弧構造を拡張し、状態情報を局所的に管理し、状態変化の計算に時間マップを使用する。本手法は Prolog で実現し、対話発話により生じる対話参加者の心的状態の変化に依存する発話解釈に適用できることを確認した。

## State-Dependent Plan Parsing

KIYOSHI KOGURE<sup>†,☆</sup> and AKIRA SHIMAZU<sup>†,☆☆</sup>

This paper proposes a plan parsing method that can calculate the effects and preconditions of actions and that parses plans in a manner dependent on state changes caused by preceding actions. This method, in particular, allows state-based action descriptions, captures enabling relationships between actions, and obtains information allowing state-dependent interpretations of action sequences. This method is based on syntactic parsing techniques, especially on chart parsing. To deal with state changes, this method augments edge structures to keep state information locally and uses time map management. This method has been implemented in Prolog. This method is ascertained to be useful for understanding dialogue utterances in a manner dependent on mental state changes of dialogue participants caused by utterances.

### 1. はじめに

対話において発話を理解し、適切に応答するために発話の背後にある話者のプランを理解する必要があることから、プラン認識に関する種々のモデルが提案されている。間接言語行為などを取り扱うためのモデル<sup>2)</sup>、複雑な対話構造を取り扱うために行為の複数の階層を使用するモデル<sup>3),8),12),13),16)</sup>、対話参加者間の信念の不一致を取り扱うためのモデル<sup>15)</sup>、文より小さい言語表現の発話を取り扱うためのモデル<sup>11)</sup>などである。

観察された行為—被観察行為—の列からプランを推論する具体的な計算方法としてはプラン解析が提案されている<sup>19)</sup>。行為に関する知識の典型的な記述には行為を構成する下位行為の列—分解—が含まれる。こ

のことから、プラン認識は、被観察行為が入力単語に対応し、行為とその分解の関係が文法規則に対応する解析過程と見なせる<sup>☆☆</sup>。このアイデアはサーカムスクリプション<sup>14)</sup>に基づくプラン認識の形式的枠組み<sup>9)</sup>の一部を文脈自由解析と対応させることにより形式化されている<sup>21)</sup>。この対応から、Earley 法<sup>6)</sup>、チャート法<sup>10)</sup>、一般化 LR 法<sup>20)</sup>などの効率的な解析手法をプラン認識に適用できる。

しかし、従来のプラン解析手法<sup>21)</sup>では、行為の効果<sup>3)</sup>や前提条件が取り扱われていない。これらを取り扱えることは協調的な対話を行うシステムを構築するうえで重要である。なぜならば、このようなシステムは発話の理解と生成にそれぞれプラン認識とプランニングを必要とするとともに、それらが密接に情報を交換することを要求するからである<sup>17)</sup>。STRIPS<sup>7)</sup>に代表されるプランニング手法が効果と前提条件を使用し

† NTT 基礎研究所

NTT Basic Research Laboratories

☆ 現在、NTT コミュニケーション科学研究所

Presently with NTT Communication Science Laboratories

☆☆ 現在、北陸先端科学技術大学院大学

Presently with Japan Advanced Institute of Science and Technology, Hokuriku

☆☆☆ たとえば、パスタ料理を作る行為がパスタを茹でる行為とパスタにソースを絡める行為から構成されるとする。入力単語に対応するパスタを茹でる行為とパスタにソースを絡める行為が観察されたとき、文法規則に対応する分解関係「パスタ料理を作る行為がパスタを茹でる行為とパスタにソースを絡める行為から構成される」を使用して、パスタ料理を作る行為を認識できる。

ているから、プラン認識でこれらを取り扱えないと、プラン認識とプランニングで行為に関する異なる知識記述を使用することになり、それらの間の密接な情報交換が困難になる。

また、対話において話者が発話で達成を意図していることは発話の効果に含まれる。この発話の効果は主に対話参加者の心的状態の変化であり、対話システムは直接的に観察できないから、発話の効果の推論が話者の意図の認識に必要である。発話の前提条件は話者が成立していると信じていることであるから、この前提条件の情報から話者の信念を推論できる。したがって、効果や前提条件を取り扱えることが望まれる。

以上の課題は次の具体的問題と結び付く。すなわち、効果と前提条件を取り扱わないプラン認識手法は(1) 既伝達の情報を繰り返さないという言語運用制約<sup>5)</sup>の実現などで有用な、達成する状態による行為の記述<sup>5),13)</sup>、(2) 行為に関する我々の直観と適合するための、行為  $\alpha$  と、 $\alpha$  の効果により前提条件が充足される行為  $\beta$  の間の関係の把握、(3) 間接言語行為などの曖昧性解消で有用な、状態に依存した行為の解釈<sup>1)</sup>が不可能である。

本論文では、対話システムのためのプラン認識に関する上記の課題を解決するために、行為の効果と前提条件を計算し、行為の効果による状態変化に依存してプランを解析するとともに、行為の効果と前提条件を取り扱うプランニング手法と知識記述を共有できる手法を提案する。本手法は基盤とする解析手法としてチャート法を使用する。これはアジェンダ機構により選好情報などにに基づく多様な解析戦略の制御が可能になるからである。また、行為の効果による状態変化をとらえるために時間マップ<sup>4),18)</sup>を使用する。本手法はPrologで実現し、対話発話により生じる対話参加者の心的状態の変化に依存する発話解釈に適用できることを確認した。

本論文の以下では、2章で行為の効果と前提条件に関するモデルを説明する。3章で効果や前提条件を取り扱うことの利点を説明する。4章で効果と前提条件を取り扱うプラン解析手法を提案する。

## 2. 行為の効果と前提条件

### 2.1 行為に関する知識の記述

プラン認識やプランニングでは、応用に適した精密さで行為に関する知識を記述するためにオペレータ<sup>7)</sup>、プランスキーマ<sup>8),13)</sup>、レシピ<sup>12),15)</sup>などと呼ばれる構造を使用する。本論文でも図1に示す recipe 構造—レシピと呼ぶ—を使用する。各フィールドは次の役割

Recipe 構造	
action	〈行為記述〉
decomposition	〈行為記述の列〉
preconditions	〈命題の集合〉
effects	〈命題の集合〉
constraint	〈制約〉

図1 Recipe 構造

Fig. 1 Recipe structure.

を持つ。

- action 値はレシピが規定する行為—レシピ行為と呼ぶ—を示す行為記述<sup>13)</sup>である。
- decomposition 値はレシピ行為の分解、すなわち、レシピ行為を構成する行為の列を指定する。この値が示す行為列の要素を部分行為と呼ぶ。各行為は部分行為を持たない原子行為か、あるいは、部分行為を持つ複合行為のいずれかである。
- preconditions 値と effects 値はそれぞれレシピ行為の前提条件の集合と効果の集合を指定する。前提条件は行為の遂行に成功するために遂行開始時間に成立している必要がある命題であり、ゆえに、行為の遂行に成功するためには遂行開始時間に preconditions 値中の任意の命題の連言が成立している必要がある。効果は行為の遂行に成功したときに遂行終了時間に成立する命題であり、ゆえに、行為の遂行に成功したときに遂行終了時間に effects 値中の任意の命題の連言が成立する。レシピの preconditions 値と effects 値は部分行為列に関するものは含まない。
- constraint 値はレシピ中の変数の具体化に関する制約である。

レシピは次のように表記する。

#### レシピ 1

```

action:      requestref(S, H, T, do(A1)),
preconditions: { ¬knowref(S, T, do(A1)),
                know(S, knowref(H, T, do(A1))),
                want(S, knowref(S, T, do(A1))),
                know(H, want(S, do(A2))) },
decomposition: [s.requestif(S, H,
                           knowref(H, T, do(A1))),
                ],
effects:     { know(H, want(S,
                           knowref(S, T, do(A1)))) },
constraint:  step(A1, A2) ∧ parameter(T, A1).

```

ここで、記法  $[X_1, X_2, \dots, X_n]$  は  $X_1, X_2, \dots, X_n$  の列を示し、 $\neg P$  は  $P$  の否定を示し、 $P \wedge Q$  は  $P$  と  $Q$  の連言を示す。レシピの表記で、大文字で始まる記号は変数である。表記に明示的に指定されていないとき、

表1 行為記述の例

Table 1 Examples of action descriptions.

行為記述	意味
go( $A, P_1, P_2, M$ )	行為者 $A$ が場所 $P_1$ から場所 $P_2$ へ方法 $M$ で行く行為。
inform( $S, H, P$ )	話者 $S$ が聴者 $H$ に命題 $P$ を伝達する言語行為。
requestref( $S, H, T, P$ )	話者 $S$ が聴者 $H$ に命題 $P$ 中の項 $T$ が何であるのかを伝達することを依頼する言語行為。
s.inform( $S, H, P$ )	話者 $S$ が聴者 $H$ に命題 $P$ を表現する平叙文を発話する表層言語行為。
s.requestif( $S, H, P$ )	話者 $S$ が聴者 $H$ に命題 $P$ の真偽に関する肯否疑問文を発話する表層言語行為。

表2 命題の例

Table 2 Examples of propositions.

命題	意味
do( $A$ )	行為 $A$ が行われること。
know( $A, P$ )	行為者 $A$ が命題 $P$ を信じていること。
knowref( $A, T, P$ )	行為者 $A$ が命題 $P$ 中の項 $T$ が何であるのかを知っていること。
want( $A, P$ )	行為者 $A$ が命題 $P$ の成立を望んでいること。

preconditions 値と effects 値は空集合, constraint 値は恒真命題 true とする。また, decomposition 値が未指定のとき, その行為が原子行為であることを示す。

レシピ 1 中の行為記述と命題は表 1 と表 2 の意味を持つ。制約  $\text{step}(A_1, A_2)$  は行為記述  $A_1$  と  $A_2$  が等しいか, あるいは, 行為記述  $A_3$  で,  $A_3$  が示す行為の部分行為を  $A_1$  が示し, かつ,  $\text{step}(A_3, A_2)$  であるものが存在するとき, かつ, そのときにかぎり, 真であり,  $\text{parameter}(P, A)$  は  $P$  が行為記述  $A$  の構成要素であるとき, かつ, そのときにかぎり, 真である。ゆえに, レシピ 1 はある行為が行われるという命題の構成要素が何であるのかの質問が構成要素を知っているかどうかの質問から構成されることを示す。また, preconditions 値の第 4 要素は質問で言及している行為, あるいは, その上位行為が行われることを話者が望んでいることを聴者が知っていること, すなわち, 質問理由が提示されていることを示す。本論文の言語的行為のレシピでは, 正常入出力条件に関する前提条件がつねに充足されていることを仮定し, その記述を省略する。また, 「知っている」と「信じている」を特に区別しない。

## 2.2 行為の効果

効果は行為の遂行に成功したときに遂行終了時間に成立する命題である。レシピに基づき, 行為の効果の規定する。そのために, 行為に関する時間を次のようにとらえる。

- 複合行為の開始時間と終了時間はそれぞれ部分行為列の開始時間と終了時間と同一である。
  - 任意の行為列  $[\alpha_1, \dots, \alpha_n]$  に関して, その開始時間と終了時間はそれぞれ  $\alpha_1$  の開始時間と  $\alpha_n$  の終了時間と同一である。また,  $\alpha_i$  ( $2 \leq i \leq n$ ) の開始時間は  $\alpha_{i-1}$  の終了時間と同一である。
- また, 状態変化に関して, 次を仮定する。

- 状態変化を起こす, すなわち, 命題を成立させるのは行為だけである。
- 成立した命題は無限に継続するか, あるいは, 矛盾する命題が成立する最初の時間まで継続する。

行為のレシピの effects 値は行為の終了時間に成立する命題を指定する。ゆえに, 部分行為を持たない原子行為の場合, このレシピが指定する命題が効果である。複合行為の場合, レシピが指定する命題以外に, これを構成する部分行為列の効果—すなわち, この列の遂行により行為終了時間に成立する命題—も考慮する必要がある。

行為列の効果は次のようにとらえる。最初に行列  $\Gamma_1 = [\gamma_1]$  の場合を考える。この列の終了時間に  $\gamma_1$  の効果がすべて成立し, 他に命題を成立させるものが存在しないから,  $\Gamma_1$  の効果は  $\gamma_1$  のものと同一である。次に行列  $\Gamma_i$  が列  $\Gamma_{i-1} = [\gamma_1, \dots, \gamma_{i-1}]$  と行為  $\gamma_i$  から構成される場合を考える。行為  $\gamma_i$  の効果は  $\gamma_i$  の終了時間と同一である  $\Gamma_i$  の終了時間に成立するから,  $\Gamma_i$  の効果でもある。列  $\Gamma_{i-1}$  の効果, すなわち,  $\Gamma_{i-1}$  の遂行により  $\Gamma_{i-1}$  の終了時間に成立している命題  $\varphi$  は矛盾する命題  $\varphi'$ —contradicts( $\varphi, \varphi'$ ) と書く\*—が  $\gamma_i$  遂行中のどこかで成立する場合,  $\Gamma_i$  の終了時間まで継続せず,  $\Gamma_i$  の効果とならない (図 2)。そうでない場合,  $\varphi$  は  $\Gamma_i$  の終了時間にも成立し, ゆえに,  $\Gamma_i$  の効果となる。

部分行為列  $\Gamma_n = [\gamma_1, \dots, \gamma_n]$  から構成される複合行為  $\alpha$  はレシピが指定する効果と行為列  $\Gamma_n$  の効果を持つものとする。

ただし, このような効果の中に同時に成立開始する 2 つの矛盾する命題が存在する場合, それらは同時に成立できないから, 効果が確定しない。このような場

\* たとえば, 任意の行為者  $A$  と命題  $\varphi$  に関して,  $A$  が  $\varphi$  と  $\neg\varphi$  を同時に知ることがないとするれば, contradicts(know( $A, \varphi$ ), know( $A, \neg\varphi$ )) である。

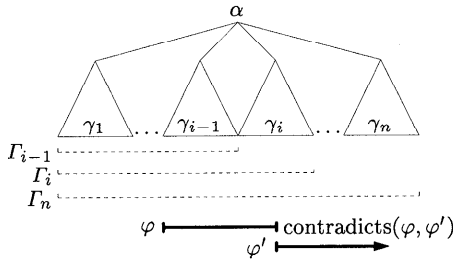


図2 行為の効果  
Fig.2 Effects of action.

合を効果衝突,あるいは,E-衝突と呼び,効果が定義されないとし,議論の対象外とする.

2.3 行為の前提条件

前提条件は行為の遂行に成功するために遂行開始時間に成立している必要がある命題である.ゆえに,行為を認識する際,行為者が合理的であるとされれば,前提条件が行為開始時間に充足されていると行為者が信じていることと矛盾する事実が発見されてはならない.

行為の効果と同様に,前提条件を規定する.行為のレシピの preconditions 値は遂行に成功するために遂行開始時間に成立している必要がある命題を指定する.ゆえに,部分行為を持たない原子行為の場合,このレシピが指定する命題が前提条件である.複合行為の場合,レシピが指定する命題以外に,これを構成する部分行為列の前提条件—すなわち,この列の遂行に成功するために遂行開始時間に成立している必要がある命題—も考慮する必要がある.

最初に行列  $\Gamma_1 = [\gamma_1]$  の場合を考える.この列の開始時間に成立する必要があるのは  $\gamma_1$  の前提条件であるから,  $\Gamma_1$  の前提条件は  $\gamma_1$  のものと同一である.次に行列  $\Gamma_i$  が列  $\Gamma_{i-1} = [\gamma_1, \dots, \gamma_{i-1}]$  と行為  $\gamma_i$  から構成される場合を考える.行列  $\Gamma_{i-1}$  の前提条件は  $\Gamma_{i-1}$  の開始時間と同一である  $\Gamma_i$  の開始時間に成立する必要があるから,  $\Gamma_i$  の前提条件でもある.行為  $\gamma_i$  の前提条件  $\psi$  に関して,次のように考える.

- 列  $\Gamma_{i-1}$  が効果  $\psi$  を持つとき (図3 場合 (a)),  $\Gamma_i$  の開始時間に  $\psi$  が成立していなくても  $\gamma_i$  の前提条件  $\psi$  は充足するから,  $\Gamma_i$  の遂行を成功させるためにその開始時間に成立している必要がない.すなわち,  $\Gamma_i$  の前提条件ではない.
- 列  $\Gamma_{i-1}$  が効果  $\psi$  を持たないとき,  $\Gamma_{i-1}$  の遂行中のどこかで  $\psi$  と矛盾する命題  $\psi'$  が成立するとき (場合 (b)),  $\Gamma_i$  の開始時間に  $\psi$  が成立していても,  $\gamma_i$  の前提条件  $\psi$  は充足されない.すなわ

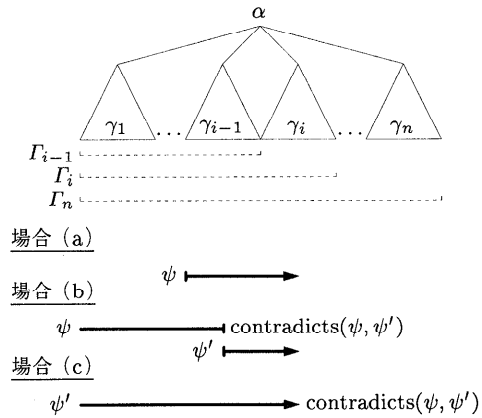


図3 行為の前提条件  
Fig.3 Preconditions of action.

ち,この前提条件の充足は不可能である.このような場合を効果-前提条件衝突,あるいは,EP-衝突と呼ぶ.このような衝突が生じる行為列は不適切であり,前提条件が定義されないとし,議論の対象外とする.

- 行列  $\Gamma_{i-1}$ ,あるいは,行為  $\gamma_i$  が  $\psi$  と矛盾する命題  $\psi'$  を前提条件として持ち,かつ,  $\psi$  と  $\psi'$  がいずれも場合 (a) でも場合 (b) でもないとき,  $\Gamma_i$  の遂行に成功するために  $\psi$  と  $\psi'$  がともに充足されるには  $\Gamma_i$  の開始時間に両方が成立している必要があるが,これは不可能である (場合 (c)).このような場合を前提条件衝突,あるいは,P-衝突と呼ぶ.この場合も,EP-衝突の場合と同様に,前提条件が定義されないとし,議論の対象外とする.
- 以上のいずれでもないとき,  $\gamma_i$  の前提条件  $\psi$  は  $\Gamma_i$  の開始時間に成立している必要があるから,  $\Gamma_i$  の前提条件である.

次にレシピが指定する効果と前提条件と部分行為列の効果と前提条件の間の相互作用を考える.レシピが指定する効果は行為終了時間に成立するから,それより前の時間に注目する部分行為列の前提条件の充足に影響を与えない.また,レシピが指定する前提条件は行為開始時間に注目するから,それより後の時間に成立する部分行為列の効果の影響を受けない.しかし,部分行為列の前提条件の命題がレシピが前提条件として指定する命題と矛盾するP-衝突が生じる場合がある.このような場合も前提条件が定義されないとし,議論の対象外とする.

部分行為列  $\Gamma_n = [\gamma_1, \dots, \gamma_n]$  から構成される複合行為  $\alpha$  はレシピが指定する前提条件と行列  $\Gamma_n$  の

前提条件を持つものとする。

以下では、E-衝突、EP-衝突、P-衝突を一括して衝突と呼ぶ。このような衝突は、レシピの記述に大局的整合性がとれていない場合や、後述の、相互に無関係であると認識される行為の列を取り扱う場合に生じる。

### 3. 効果と前提条件を取り扱う利点

プラン認識で行為の分解に加えて、効果と前提条件を取り扱うことには、それらを取り扱うプランニングと同一レシピを使用できることを含め、次の利点がある。

#### 3.1 状態に基づく行為記述

レシピを記述する際、達成する状態を行為を指定できると便利であり、プランニングやプラン認識でそのようなレシピが使用されている<sup>5),13)</sup>。以下では、命題  $\varphi$  が成立する状態を達成する行為を示すのに記述  $\text{achieve}(\varphi)$  を使用する。たとえば、次の説明用のレシピは S が H に  $P \wedge Q$  を伝達する行為が H が P を知っている状態を達成する行為と H が Q を知っている状態を達成する行為から構成されることを示す。

#### レシピ 2

action:           inform(S, H,  $P \wedge Q$ ),  
preconditions: { know(S,  $P \wedge Q$ ),  
                  know(S,  $\neg\text{know}(H, P \wedge Q)$ ),  
decomposition: [ achieve(know(H, P)),  
                  achieve(know(H, Q)) ],  
effects:           { know(H,  $P \wedge Q$ ),  
                  know(S, know(H,  $P \wedge Q$ )) }.

この記述  $\text{achieve}(\varphi)$  が示す行為は効果  $\varphi$  を持つ行為になる。効果  $\varphi$  を持つ多数の行為が存在するとき、この形式の記述が許されなければ、それらを列挙する必要が生じる<sup>\*</sup>。すなわち、この形式の記述にはレシピ記述を容易にするという利点がある。

この形式の行為記述の重要な性質はこの記述が示す行為が遂行直前の状態に依存することである。極端な場合、成立させようとする命題がすでに成立しているならば、何も行為を遂行する必要がない。たとえば、 $\text{achieve}(\text{know}(H, P))$  は H が P を知っているときに H に P を知らせるために特に何かを行う必要がないことを示す。この性質を利用し、既伝達の情報は繰り返さないという言語運用制約が漸次的発話生成のためのプランニングで実現されている<sup>5)</sup>。この行為記述形

式をプラン認識で許し、このような言語運用制約に従う発話列を理解するためには、行為の効果とそれによる状態変化を計算する必要がある。

#### 3.2 行為間の可能化関係

プラン認識の逆過程のプランニングを考える。プランニング手続きはゴールが与えられると、ゴールを効果として持つ行為—主行為と呼ぶ—を探索する。発見した主行為が未充足前提条件をとともなうとき、これを充足し、主行為の遂行を可能にする行為—可能化行為と呼ぶ—を探索し、発見すると、可能化行為に主行為が続く行為列を返す。この可能化行為と主行為の間の関係を可能化関係と呼ぶ。可能化行為に主行為が続く形式の行為列が与えられたとき、プラン認識は可能化関係を把握し、行為列全体を主行為の特定の効果（ゴール）を達成するためのものであると認識するべきである。たとえば、質問に関するレシピ 1 が与えられているとき、「開発センタから研究所へ行きたいのですが」と発話する行為と「開発センタから研究所へ行く方法を知っていますか」と発話する行為から<sup>\*\*</sup>、前者により質問理由が提示され、後者による質問の前提条件が充足されることを認識するべきである。このような可能化関係の認識には行為の効果と前提条件を計算する必要がある。

行為の前提条件を充足する行為を部分行為とし、前提条件を不要にすることが考えられる。しかし、部分行為は行為の一部であるが、前提条件の充足はそうではなく、この区別の重要性が指摘されている<sup>3)</sup>。たとえば、言語的行為一般の前提条件として正常入出力条件があり、その具体化として話者と聴者が話ができることがあるが、話者がこの前提条件を充足するために聴者に近づき、質問しても、近づいたことを質問の一部であるとは普通は考えない。前提条件を取り扱うことには、このような我々の直観に適合するようにレシピを記述できるという利点がある。

#### 3.3 行為の状態依存解釈

行為の解釈の妥当性が行為の直前の状態と前提条件に依存することがある。たとえば、(a)「開発センタから研究所へ行く方法を知っていますか」のような命題の構成要素の「方法」を知っているかどうかに関する疑問文を発話する行為により、話者は(1)聴者が行く方法を知っているかどうかの質問、(2)方法の質問、(3)方法を伝達することの提案のいずれかを行っている可能性があり、各行為は異なる前提条件を持ち、前

<sup>\*</sup> さらに、新しい行為  $\alpha$  のレシピを追加する際、その効果の達成を部分行為として持つ可能性のある行為を調べ、そのような行為それぞれに関して、 $\alpha$  を decomposition 値中に含むレシピを追加するなどの知識管理上の問題を生じる。

<sup>\*\*</sup> ここでは、方法がどこかに行くことに関する命題の構成要素であることを仮定する。

提条件の充足程度に依存して行為の解釈の妥当性が異なる<sup>1)</sup>。異なる前提条件としては、行為(1)は聴者が方法を知っているかどうかを話者が知らないことを持ち、行為(2)は、レシピ1が指定するように、聴者が方法を知っていることを話者が知っていることを持ち、行為(3)は聴者が知らないことを話者が知っていることを持つ。ゆえに、聴者が方法を知っていることを話者が知っているとき、発話(a)の妥当な解釈は前提条件が充足される行為(2)である。このような解釈の妥当性を考慮するには、行為の直前の状態と行為の前提条件を知る必要がある。前提条件の命題は初期状態で成立していたり、以前の行為の効果により成立していることがあるから、行為直前の状態を知るには初期状態と先行行為の効果から状態を推論する必要がある。

## 4. プラン解析手法

### 4.1 分解文法によるプラン解析

行為に関するレシピの分解(decomposition)値中の行為が時間的に全順序を持ち、連続しているとき、行為とその分解の間の関係は文法規則と見なせる。たとえば、レシピ1の分解関係は次の規則で表現される。

$\text{requestref}(S, H, T, \text{do}(A1))$

$\rightarrow \text{s\_requestif}(S, H, \text{knowref}(H, T, \text{do}(A1)))$

分解関係のこのような解釈は分解関係の文法(分解文法)を与える。被観察行為列中の行為が時間的に全順序を持ち、連続していると仮定すると、効率的な統語解析手法をプラン認識に利用できる。

この分解文法による解析のアイデアを基盤に、行為の効果と前提条件を取り扱う手法を構成する。以下では、議論を具体化するために上昇型チャート法に焦点を当てる。すなわち、チャートの弧で行為を表現し、節点で時間を表現する。

### 4.2 効果と前提条件の表現

#### 4.2.1 効果の表現

行為の効果をとらえるために時間マップと呼ばれる時間推論のための一般的な手法を使用する<sup>\*</sup>。時間マップは以下から構成される。

**事実トークンの集合** 事実トークンは3つ組  $\langle t_1, t_2, \varphi \rangle$  である。ここで、 $t_1$  と  $t_2$  は時間であり、 $\varphi$  はレシピの effects 値の要素と同様の命題である。事実トークン  $\langle t_1, t_2, \varphi \rangle$  は  $\varphi$  が  $t_1$  で成立し、 $t_2$  まで継続するか、あるいは、矛盾する命題が成立する最初の時間まで継続することを表現する。

**時間の順序** 前述のようにチャートの節点で時間を表現する。単純化のために節点が整数であることを仮定する。また、特殊な時間として無制限の継続を表現するための  $T$  を導入する。レシピが指定する効果は行為終了時間に成立するが、活性弧が示す行為は終了時間が未確定であるから、そのレシピが指定する効果の成立開始時間も未確定であり、未束縛変数で示される。ただし、この行為の終了時間は整数で示される既知の部分行為の終了時間より後であり、時間  $T$  より前である。そこで、整数の間では通常的全順序関係と同様であり、かつ、 $T$  が任意の未束縛変数と整数より大きく、任意の未束縛変数が任意の整数より大きい順序  $\preceq$  を使用する。以下では、時間に関する順序はこれに固定する。

事実トークン集合  $\mathcal{F}$  を持つ時間マップでは、事実  $\varphi$  は次の条件が充足される時、かつ、そのときにかぎり、 $t_1$  から  $t_2$  まで成立することが保証されるといい、 $\text{tm\_holds}(\langle t_1, t_2, \varphi \rangle, \mathcal{F})$  と書かれる。

(1)  $t_3 \preceq t_1 \preceq t_2 \preceq t_4$  である  $\langle t_3, t_4, \varphi \rangle \in \mathcal{F}$  が存在し、かつ、

(2)  $\text{contradicts}(\varphi, \varphi')$ 、かつ、 $t_3 \prec t_5 \preceq t_2$  である  $\langle t_5, t_6, \varphi' \rangle \in \mathcal{F}$  が存在しない。

レシピが指定する効果は行為終了時間に成立し、無限に継続するか、あるいは、矛盾する命題が成立する最初の時間まで継続するから、時間  $t$  に終了する行為のそのような効果  $\varphi$  を  $\langle t, T, \varphi \rangle$  で表現する。

#### 4.2.2 前提条件の表現

行為の前提条件も事実トークンと同様の3つ組で表現する。レシピが指定する前提条件は行為の開始時間で成立している必要があるから、時間  $t$  に開始する行為のそのような前提条件  $\psi$  は  $\langle t, t, \psi \rangle$  で表現する。

#### 4.3 データ構造

前述のようにチャートの弧で行為を表現する。弧が表現する行為の効果と前提条件を弧と対応させる必要があるから、弧として図4に示す edge 構造—弧と呼ぶ—を使用する。各フィールドは以下の役割を持つ。

- start 値と end 値は弧が表現する行為(の部分)の開始時間と終了時間を表現する節点である。
- action 値と subactions 値は弧が表現する行為の記述と、これを認識するために同定する必要がある行為の記述の列である。弧は subactions 値が空列であるときに不活性弧、他のときに活性弧と呼ぶ。
- effects 値と preconditions 値はそれぞれ効果と前提条件を表現する3つ組の集合である。
- aend 値は行為の終了時間を示す変数を格納する。

<sup>\*</sup> 本論文では文献(18)の用語を使用する。

Edge 構造	
start	〈節点〉
end	〈節点〉
action	〈行為記述〉
subactions	〈行為記述の列〉
effects	〈3つ組の集合〉
preconditions	〈3つ組の集合〉
aend	〈変数〉   〈節点〉
constraint	〈制約〉
tree	〈列〉

図 4 Edge 構造  
Fig. 4 Edge structure.

活性弧が表現する行為の終了時間は未確定である。この行為のレシピが指定する効果の開始時間（事実トークンの第 1 要素）をこの aend 値とする。この値は不活性弧では end 値に束縛する。

- constraint 値は変数具体化に関する制約である。
- tree 値は解析過程を表現する列である。各行為を開始時間、終了時間、行為記述の 3 つ組で示す。

弧  $e$  のフィールド  $field$  の値を  $e.field$  で示す。同様にレシピ  $r$  のフィールド  $field$  の値を  $r.field$  で示す。

**4.4 効果と前提条件を計算する基本チャート手続き**  
行為記述の列を入力とし、レシピが指定する分解関係によるプラン解析を行う上昇型チャート法は統語解析の場合と同様に、3 つの手続きで解析を進める。すなわち、入力行為記述から弧を作成する入力手続き、不活性弧とレシピの分解関係から弧を作成する上昇型予測手続き、活性弧と不活性弧から弧を作成する結合手続きである。これらの手続きを拡張し、プラン解析中に効果と前提条件を計算する。これは、4.2 節で述べた効果と前提条件の表現方法から、上述の手続きに対応する以下の手続きで行える。以下では、入力が長さ  $n$  の行為記述列  $[\alpha_1, \alpha_2, \dots, \alpha_n]$  であり、 $j \neq k$  に関して、行為記述  $\alpha_j$  と  $\alpha_k$  が共通の変数を含まないことを仮定する。

**手続き 1 (入力)**  $\alpha_j$  を  $j$  番目の行為記述とし、 $r$  を原子行為のレシピ<sup>☆</sup>とする。このとき、 $r.action$  と  $\alpha_j$  の単一化子  $\theta$  で、いずれの衝突もなく、かつ、 $r.constraint$  を充足する最汎なものそれぞれに関して、変数名変更で同一になる弧がすでに存在しないかぎり<sup>☆☆</sup>、次の  $j-1$  から  $j$  までの不活性弧 (start 値が

$j-1$ , end 値が<sup>§</sup>  $j$ , subactions 値が<sup>§</sup>空列  $[]$  である弧)  $e$  を作成する。

$$\begin{aligned}
 e.action &= \alpha_j\theta, \\
 e.effects &= \{ \langle j, \top, \varphi\theta \rangle \mid \varphi \in r.effects \}, \\
 e.preconditions &= \{ \langle j-1, j-1, \psi\theta \rangle \mid \\
 &\quad \psi \in r.preconditions \}, \\
 e.aend &= j, \\
 e.constraint &= (r.constraint)\theta, \\
 e.tree &= [\langle j-1, j, \alpha_j\theta \rangle].
 \end{aligned}$$

ここで、記法  $X\theta$  は  $X$  に代入  $\theta$  を適用した結果を示す。

**手続き 2 (上昇型予測)**  $e_i$  を不活性弧、 $r$  を decomposition 値  $[\gamma_1, \gamma_2, \dots, \gamma_m]$  のレシピとする。このとき、 $e_i.action$  と  $\gamma_1$  の単一化子  $\theta$  で、いずれの衝突もなく、かつ、 $e_i.constraint \wedge r.constraint$  を充足する最汎なものそれぞれに関して、 $e_i.start$  から  $e_i.end$  への次の弧  $e$  を作成する。

$$\begin{aligned}
 e.action &= (r.action)\theta, \\
 e.subactions &= [\gamma_2, \dots, \gamma_m]\theta, \\
 e.effects &= (e_i.effects)\theta \\
 &\quad \cup \{ \langle v, \top, \varphi\theta \rangle \mid \\
 &\quad \quad \varphi \in r.effects \}, \\
 e.preconditions &= (e_i.preconditions)\theta \\
 &\quad \cup \{ \langle e_i.start, e_i.start, \psi\theta \rangle \mid \\
 &\quad \quad \psi \in r.preconditions \}, \\
 e.aend &= v, \\
 e.constraint &= (e_i.constraint \\
 &\quad \wedge r.constraint)\theta, \\
 e.tree &= [\langle e_i.start, v, r.action \rangle, \\
 &\quad e_i.tree]\theta.
 \end{aligned}$$

ここで、 $v$  は新規変数であり、 $e.subactions$  が空列であるときに  $e.end$  に束縛する。

**手続き 3 (結合)**  $e_a$  と  $e_i$  を隣接する活性弧と不活性弧とし、 $e_a.subactions$  を  $[\gamma_1, \gamma_2, \dots, \gamma_m]$  とする。このとき、各  $\mathcal{P} \subseteq e_i.preconditions$  に関して、次を行う。すなわち、 $\gamma_1$  と  $e_i.action$  の単一化子  $\theta$  で、各  $\langle j, j, \psi \rangle \in \mathcal{P}$  に関して、

$$\text{tm\_holds}(\langle j, j, \psi\theta \rangle, (e_a.effects \cup e_i.effects)\theta)$$

であり、各  $\langle j, j, \psi \rangle \in e_i.preconditions - \mathcal{P} = \mathcal{P}'$  に関して、

$$\neg \text{tm\_holds}(\langle j, j, \psi\theta \rangle, (e_a.effects \cup e_i.effects)\theta)$$

であり、いずれの衝突もなく、かつ、 $e_a.constraint \wedge e_i.constraint$  を充足する最汎なものそれぞれに関して、 $e_a.start$  から  $e_i.end$  への次の弧  $e$  を作成する。

$$\begin{aligned}
 e.action &= (e_a.action)\theta, \\
 e.subactions &= [\gamma_2, \dots, \gamma_m]\theta,
 \end{aligned}$$

☆ 正確には、それまでに出現した変数を含まないように変数名変更されたレシピである。手続きの記述中のレシピはこのような構造を示す。

☆☆ このような弧が存在しないという条件は以下のすべての手続きに適用されるものとし、以下の手続きの記述では省略する。

$$\begin{aligned}
e.\text{effects} &= (e_a.\text{effects} \cup e_i.\text{effects})\theta, \\
e.\text{preconditions} &= (e_a.\text{preconditions} \cup \mathcal{P}')\theta, \\
e.\text{aend} &= e_a.\text{aend}, \\
e.\text{constraint} &= (e_a.\text{constraint} \\
&\quad \wedge e_i.\text{constraint})\theta \wedge C, \\
e.\text{tree} &= T\theta.
\end{aligned}$$

ここで、 $C$  はさらなる代入だけで前提条件の充足状態が変化することを禁止する次の制約である。

$$\begin{aligned}
C &= \bigwedge_{(j,j,\psi) \in \mathcal{P}\theta} \text{tm.holds}(\langle j, j, \psi \rangle, e.\text{effects}) \\
&\quad \wedge \bigwedge_{(j,j,\psi) \in \mathcal{P}'\theta} \neg \text{tm.holds}(\langle j, j, \psi \rangle, e.\text{effects}).
\end{aligned}$$

また、 $T$  は列  $e_a.\text{tree}$  の最後に要素  $e_i.\text{tree}$  を追加した列であり、 $e.\text{subactions}$  が空列であるときに  $e.\text{aend}$  を  $e.\text{end}$  に束縛する。

上述の手続き中の衝突に関する条件を説明する。E-衝突は同時に成立開始する 2 つの矛盾する命題が存在することであるから、弧  $e$  で E-衝突が生じるというのは、 $\text{contradicts}(\varphi_1, \varphi_2)$  である  $\langle j, T, \varphi_1 \rangle$ ,  $\langle j, T, \varphi_2 \rangle \in e.\text{effects}$  が存在することである。EP-衝突は前提条件の命題が行為開始時間で成立していたとしても、充足しないということであるから、弧  $e$  で EP-衝突が生じるというのは、

$\neg \text{tm.holds}(\langle j, j, \psi \rangle, \{ \langle e.\text{start}, T, \psi \rangle \} \cup e.\text{effects})$  となる  $\langle j, j, \psi \rangle \in e.\text{preconditions}$  が存在することである。また、P-衝突は矛盾する命題を前提条件が含むことであるから、弧  $e$  で P-衝突が生じるというのは、 $\text{contradicts}(\psi_1, \psi_2)$  である  $\langle j_1, j_1, \psi_1 \rangle$ ,  $\langle j_2, j_2, \psi_2 \rangle \in e.\text{preconditions}$  が存在することである。

上述の手続きで作成される弧  $e$  が表現する行為の終了時間で成立している効果は、

$$\{ \varphi \mid \text{tm.holds}(\langle e.\text{end}, e.\text{end}, \varphi \rangle, e.\text{effects}) \}$$

であり、この行為遂行の成功のために開始時間で成立している必要がある前提条件は、

$$\{ \psi \mid \langle j, j, \psi \rangle \in e.\text{preconditions} \}$$

である。このように弧が表現する行為の効果と前提条件を計算するための基本手段を獲得した。以下では、3 章で吟味した状況を取り扱うために、さらにプラン解析手続きを拡張する。

#### 4.5 状態に基づく行為記述

3.1 節で言及した形式  $\text{achieve}(\varphi)$  の記述をレシピの  $\text{decomposition}$  値中に許すようにする。まず、この形式の記述が効果  $\varphi$  を持つ行為を示すという性質を満たすようにする。そのために、 $\text{decomposition}$  値と関係する手続き 2 と手続き 3 を次のように拡張する。

手続き 4 (上昇型予測)  $e_i$  を不活性弧、 $r$  を

$\text{decomposition}$  値  $[\gamma_1, \gamma_2, \dots, \gamma_m]$  のレシピとする。このとき、

(1)  $\gamma_1 = \text{achieve}(\varphi)$  の場合、

$$\text{tm.holds}(\langle e_i.\text{end}, e_i.\text{end}, \varphi \theta \rangle, (e_i.\text{effects})\theta)$$

である代入  $\theta$  で、いずれの衝突もなく、かつ、 $e_i.\text{constraint} \wedge r.\text{constraint}$  を充足する最汎なものそれぞれに関して、手続き 2 と同様に指定される弧  $e$  を作成する。ただし、 $e$  の  $\text{constraint}$  値には、 $\varphi$  が不成立となることを禁止する次の制約が追加されている。

$$\text{tm.holds}(\langle e_i.\text{end}, e_i.\text{end}, \varphi \theta \rangle, (e_i.\text{effects})\theta).$$

(2) 他の場合、手続き 2 と同様に弧  $e$  を作成する。

手続き 5 (結合)  $e_a$  と  $e_i$  を隣接する活性弧と不活性弧とし、 $e_a.\text{subactions}$  を  $[\gamma_1, \gamma_2, \dots, \gamma_m]$  とする。このとき、

(1)  $\gamma_1 = \text{achieve}(\varphi)$  の場合、各  $\mathcal{P} \subseteq e_i.\text{preconditions}$  に関して、次を行う。すなわち、

$$\text{tm.holds}(\langle e_i.\text{end}, e_i.\text{end}, \varphi \theta \rangle, (e_i.\text{effects})\theta)$$

である代入  $\theta$  で、各  $\langle j, j, \psi \rangle \in \mathcal{P}$  に関して、

$$\text{tm.holds}(\langle j, j, \psi \theta \rangle, (e_a.\text{effects} \cup e_i.\text{effects})\theta)$$

であり、各  $\langle j, j, \psi \rangle \in e_i.\text{preconditions} - \mathcal{P}$  に関して、

$$\neg \text{tm.holds}(\langle j, j, \psi \theta \rangle, (e_a.\text{effects} \cup e_i.\text{effects})\theta)$$

であり、いずれの衝突もなく、かつ、 $e_a.\text{constraint} \wedge e_i.\text{constraint}$  を充足する最汎なものそれぞれに関して、手続き 3 と同様に指定される弧  $e$  を作成する。ただし、 $e$  の  $\text{constraint}$  値には、 $\varphi$  が不成立となることを禁止する次の制約が追加されている。

$$\text{tm.holds}(\langle e_i.\text{end}, e_i.\text{end}, \varphi \theta \rangle, (e_i.\text{effects})\theta).$$

(2) 他の場合、手続き 3 と同様に弧  $e$  を作成する。

次に形式  $\text{achieve}(\varphi)$  の記述が  $\varphi$  が成立している状態では何も行為を遂行する必要がないことを示すという性質を満たすようにする。命題  $\varphi$  がこの形式で指定される行為を遂行開始しようとする時間に成立しているのは、この行為を部分行為として持つ行為  $\alpha$  のこの時間より前に遂行された部分行為により成立し、この時間まで継続している場合か、あるいは、 $\alpha$  の遂行開始時間には成立していて、継続している場合のいずれかである。前者の場合は、形式  $\text{achieve}(\varphi)$  の記述を  $\text{subactions}$  値の第 1 要素として持つ活性弧の終端で  $\varphi$  が成立している場合である。後者の場合には、 $\varphi$  はこの形式で指定される行為なしで  $\alpha$  を遂行する場合の前提条件ととらえられる。これらの場合を取り扱うために次の手続きを導入する。

手続き 6 (前提条件導入)  $e_a$  を活性弧とし、 $e_a.\text{subactions}$  を  $[\gamma_1, \gamma_2, \dots, \gamma_m]$  とし、 $\gamma_1 = \text{achieve}(\varphi)$  であるとする。このとき、代入  $\theta$  で、

$$\text{tm.holds}(\langle e_a.\text{end}, e_a.\text{end}, \varphi \theta \rangle, (e_a.\text{effects})\theta)$$



が (a) 成立する場合と (b) 成立しない場合のそれぞれに関して、いずれの衝突もなく、かつ、 $e_a.constraint$  を充足する最汎なものそれぞれに関して、 $e_a.start$  から  $e_a.end$  への次の弧  $e$  を作成する。

$$\begin{aligned} e.action &= (e_a.action)\theta, \\ e.subactions &= [\gamma_2, \dots, \gamma_m]\theta, \\ e.effects &= (e_a.effects)\theta, \\ e.preconditions &= \begin{cases} (e_a.preconditions)\theta, & \text{場合 (a) のとき,} \\ (e_a.preconditions)\theta \\ \cup \{ \langle e_a.end, e_a.end, \varphi \theta \rangle \}, & \text{場合 (b) のとき,} \end{cases} \\ e.aend &= e_a.aend, \\ e.constraint &= (e_a.constraint)\theta \wedge C, \\ e.tree &= (e_a.tree)\theta. \end{aligned}$$

ここで、 $C$  はさらなる代入だけで前提条件の充足状態が変化することを禁止する制約であり、場合 (a) では、

$$C = \text{tm\_holds}(\langle e_a.end, e_a.end, \varphi \theta \rangle, e.effects)$$

であり、場合 (b) では、

$$C = \neg \text{tm\_holds}(\langle e_a.end, e_a.end, \varphi \theta \rangle, e.effects)$$

である。また、 $e.subactions$  が空列であるときに  $e.aend$  を  $e.end$  に束縛する。

上述の手続き 4~6 により形式  $\text{achieve}(\varphi)$  の行為記述を取り扱える。

#### 4.6 行為間の可能化関係

3.2 節で説明したように、行為列  $[\alpha_1, \alpha_2]$  で、 $\alpha_1$  と  $\alpha_2$  の間に (部分的な) 可能化関係が存在する場合、すなわち、 $\alpha_1$  の効果により  $\alpha_2$  の前提条件 (の一部) が充足される場合、この 2 つの行為は独立したものではなく、 $\alpha_2$  の特定の効果を達成するためのものであることがある。このような場合を認識するために、次の手続きを導入する。

**手続き 7 (前提条件削除)**  $e_1$  と  $e_2$  を隣接する不活性弧とする。このとき、 $e_2.preconditions$  の空ではない部分集合  $\mathcal{P}$  それぞれに関して、以下を行う。すなわち、各  $\langle j, j, \psi \rangle \in \mathcal{P}$  に関して、

$$\text{tm\_holds}(\langle j, j, \psi \theta \rangle, (e_1.effects \cup e_2.effects)\theta)$$

であり、各  $\langle j, j, \psi \rangle \in e_2.preconditions - \mathcal{P} = \mathcal{P}'$  に関して、

$$\neg \text{tm\_holds}(\langle j, j, \psi \theta \rangle, (e_1.effects \cup e_2.effects)\theta)$$

である代入  $\theta$  で、いずれの衝突もなく、かつ、 $e_1.constraint \wedge e_2.constraint$  を充足する最汎なものそれぞれに関して、 $e_1.start$  から  $e_2.end$  への次の不活性弧  $e$  を作成する\*。

$$\begin{aligned} e.action &= (e_2.action)\theta, \\ e.effects &= (e_1.effects \cup e_2.effects)\theta, \end{aligned}$$

$$\begin{aligned} e.preconditions &= (e_1.preconditions \cup \mathcal{P}')\theta, \\ e.aend &= e_2.end, \\ e.constraint &= (e_1.constraint \\ &\quad \wedge e_2.constraint)\theta \wedge C, \\ e.tree &= [ \langle e_1.start, e_2.end, \\ &\quad e_2.action \rangle, \\ &\quad e_1.tree, \\ &\quad e_2.tree ]\theta. \end{aligned}$$

ここで、 $C$  はさらなる代入だけで前提条件の充足状態が変化することを禁止する次の制約である。

$$\begin{aligned} C &= \bigwedge_{\langle j, j, \psi \rangle \in \mathcal{P}\theta} \text{tm\_holds}(\langle j, j, \psi \rangle, e.effects) \\ &\quad \wedge \bigwedge_{\langle j, j, \psi \rangle \in \mathcal{P}'\theta} \neg \text{tm\_holds}(\langle j, j, \psi \rangle, e.effects). \end{aligned}$$

#### 4.7 行為の状態依存解釈

3.3 節で言及したような状態依存解釈を行うためには、行為遂行直前の状態を知る必要がある。この状態は初期状態に先行行為の効果作用した結果であるから、これを計算しなければならない。初期状態は初期状態で成立する命題を効果として持つ特殊な行為を表現する不活性弧を導入することで取り扱える。そのような弧は、たとえば、次の手続きで導入できる。

**手続き 8 (初期状態)**  $\Phi$  を初期状態を表現する命題の集合とする。このとき、 $-1$  から  $0$  への次の不活性弧を作成する。

$$\begin{aligned} e.action &= \text{initialize}, \\ e.effects &= \{ \langle 0, T, \varphi \rangle \mid \varphi \in \Phi \}, \\ e.preconditions &= \emptyset, \\ e.aend &= 0, \\ e.constraint &= \text{true}, \\ e.tree &= [ \langle -1, 0, \text{initialize} \rangle ]. \end{aligned}$$

行為直前の状態を知るためには、先行する、相互に無関係であると認識される行為の列—無関係行為列と呼ぶ—を考慮しなければならない。また、行為の前提条件が充足されるかどうか、充足される可能性があるかどうかは、先行行為列と組み合わせて、充足されていない前提条件が残ったり、衝突が生じるかどうかを

\* この手続きで作成される弧  $e$  の action 値を  $e_2$  の action 値の具体化にしているのは、結果的に前提条件の充足をこの記述で示される行為の一部として取り扱っていることになり、前提条件の充足が行為の一部ではないという主張と矛盾して見えるかもしれない。しかし、これは上昇型予測や結合の手続きでの行為記述の照合の説明を単純化するための便宜上のものである。この問題は、 $e$  の action 値に別の行為記述 (たとえば、 $e_1$  と  $e_2$  の action 値を  $\alpha_1$  と  $\alpha_2$  とするとき、 $\text{enabling}(\alpha_1\theta, \alpha_2\theta)$ ) を使用し、行為記述の照合を複雑にすることにより解決される。実現したプログラムでもそのようにしている。

検査することにより調べられる。そこで、無関係行為列を構成することを考える。これはレシピの追加により行える。最初に、無関係行為列の要素となることを許す行為 $\alpha$ の記述  $\alpha$  に関して、次の形式のレシピを導入する。

### レシピ 3

action: seq\_element( $\alpha$ ),  
decomposition: [ $\alpha$ ].

次にこのような行為から行為列を作成する次のレシピを導入する。

### レシピ 4

action: sequence([Act]),  
decomposition: [seq\_element(Act)].

### レシピ 5

action: sequence([Act | Acts]),  
decomposition: [seq\_element(Act), sequence(Acts)].

ここで、記法  $[E | S]$  は列  $S$  の先頭に要素  $E$  を挿入することにより得られる列を示す。

### 行為の状態依存解釈のための選好規則

上述のレシピを使用すると、被観察行為列から無関係行為列を認識することになる。サーカムスクリプションに基づくプラン認識の枠組みでは、プランに基づく行為間の関係だけを考慮し、同一の行為に組み込まれないという意味で相互に無関係な行為の数が少ないものが被観察行為列の良い解釈であるとされている<sup>9)</sup>。これは、たとえば、次の選好規則で取り扱える。

**選好規則 1** 同一区間を覆う 2 つの不活性弧が表現する行為列に関して、短い行為列を選好する $\star\star$ 。

この規則は行為に関する通常のレシピの分解関係や可能化関係によりまとめられた結果の行為を選好する。

以上の準備により、状態依存解釈が可能になる。たとえば、各前提条件の充足が同じ重要性を持つならば、次の前提条件に基づく選好規則を定義できる。

**選好規則 2** 同一区間を覆う 2 つの不活性弧が表現する行為 (列) に関して、preconditions 値の要素数が少ない弧のものを選好する。

このような状態に依存する選好規則は、3.3 節で説

明したような状態依存解釈を可能にする。

## 4.8 まとめ

要約すると、本プラン解析手法はチャート初期化のために手続き 8 を使用し、各入力行為記述に対して手続き 1 を使用し、さらに解析を進めるために手続き 4~7 を使用する。また、本手法は、行為に関する通常のレシピ以外に、無関係行為列の要素となることを許す各行為に対するレシピ 3 とレシピ 4 とレシピ 5 を使用する。

これまでは説明を単純化するために言及しなかったが、本プラン解析手法は多様な解析戦略の制御を可能にするチャートのアジェンダ機構を使用する。すなわち、上述の手続きで作成された弧をいったんアジェンダに格納し、アジェンダから上述の選好規則などに基づく優先順序で弧を選択し、それをチャートに組み込み、組み込まれた弧だけを対象にして上述の手続きを適用する。

### プラン解析例

対話理解におけるプラン解析例を図 5 に示す。この例では、無関係行為列のためのレシピ (requestref に関するレシピ 3 を含む) とレシピ 1 に加えて、2 つの原子行為 s\_inform と s\_requestif が前提条件も効果も持たないことを指定するレシピと、次のレシピを使用する。

### レシピ 6

action: inform(S, H, P),  
preconditions: { know(S, P), know(S,  $\neg$ know(H, P)) },  
decomposition: [s\_inform(S, H, P)],  
effects: { know(H, P), know(S, know(H, P)) }.

また、初期状態で成立する命題の集合として、話者 s が開発センタ c から研究所 l へなんらかの方法 m で行くことを望んでいることを s が知っていること、この s の願望を聴者 h が知らないことを s が知っていること、開発センタから研究所へ行く方法を s が知らないこと、この方法を h が知っていることを s が知っていること、この方法を知ることを s が望んでいることを示す

```
{ know(s, want(s, do(go(s, c, l, m))),
  know(s,  $\neg$ know(h,
    want(s, do(go(s, c, l, m))))),
   $\neg$ knowref(s, m, do(go(s, c, l, m))),
  know(s, knowref(h, m, do(go(s, c, l, m))))
  want(s, knowref(s, m, do(go(s, c, l, m)))) }
```

が与えられ、被観察行為の列の記述として、自分が開発センタから研究所へ行くことを望んでいるという命題を表現する平叙文 (たとえば、「開発センタから研究

$\star$  類似の概念に「部分行為とならない行為」—END 行為<sup>9)</sup> がある。Vilain<sup>21)</sup>は END 行為だけから無関係行為列を構成している。しかし、実時間対話理解のような応用を考えると、つねに END 行為を認識するべきかどうかは疑問である。聴者が END 行為を認識することを話者が意図しているとはかぎらないこと<sup>19)</sup>、話者の発話に追従しなければならないことなどからである。そこで、END 行為以外の行為を無関係行為列が含むことを許すように、この概念を導入した。

$\star\star$  ここで、行為列の長さは、たとえば、次のようにとらえる。弧の action 値が形式 sequence(Acts) の場合には列 Acts の長さとし、他の場合には 1 とする。

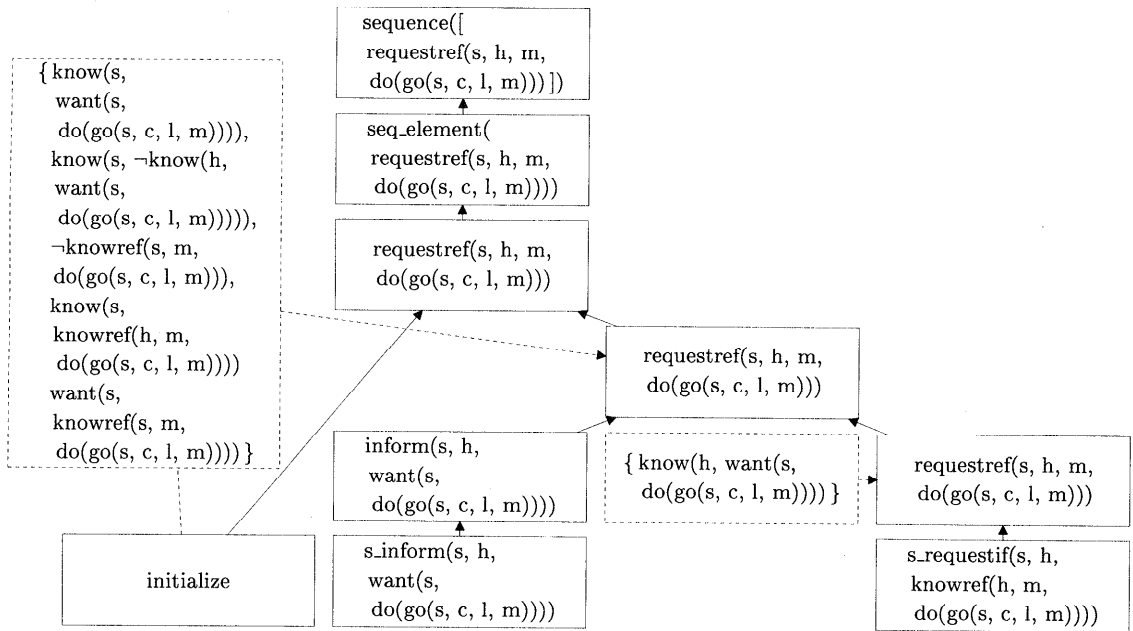


図5 プラン解析例  
Fig. 5 Example plan parsing.

所へ行きたいのですが」)をsがhに発話する行為を示す

s\_inform(s, h, want(s, do(go(s, c, l, m))))

と、開発センタから研究所へ行く方法を知っているかどうかに関する肯否疑問文(たとえば、「開発センタから研究所へ行く方法を知っていますか」)をsがhに発話する行為を示す

s\_requestif(s, h, knowref(h, m, do(go(s, c, l, m))))

が与えられているとする。このとき、図5のように解析できる。この図で実線の箱は行為を示し、実線矢印は出発点の行為から到達点の行為が認識されることを示す。点線の箱が途中にある点線矢印は可能化関係を示し、矢印の出発点の行為が効果に点線の箱の中の命題を含み、これにより矢印の到達点の行為の前提条件が(部分的に)充足されることを意味する。この図は次のことを意味する。すなわち、行為 s\_inform から inform が認識され(レシピ6)、s\_requestif から requestref が認識され(レシピ1)、inform の効果が requestref の前提条件を充足することから両行為が requestref の効果を達成するための行為であることが認識され、この requestref の前提条件が初期状態により充足され、この requestref だけからなる列が認識される(レシピ3とレシピ4)ことである。この場合、この s\_requestif で記述される行為に関する他の解釈(3.3節で述べた聴者の知識に関する質問と

情報提示の提案)に必要なレシピを追加しても、それらの解釈ではEP-衝突が生じることなどから、図の解釈が選択される。

#### 4.9 拡張

説明を単純化するために言及しなかったが、本プラン解析手法は次のように拡張でき、実際に実現されたプログラムはこの拡張に従っている。

##### 4.9.1 意図効果と副次効果

行為の効果は行為者が行為遂行の際に達成することを意図している効果—意図効果—と、付随的に成立する効果—副次効果—に分類できる<sup>\*</sup>。このように分類できる場合、手続き7により可能化関係を認識する際、意図効果だけを対象とする方が適切である。このような可能化関係の認識は、効果を意図効果と副次効果に分類してレシピに記述し、弧の effects 値の要素を事実トークンである3つ組から意図効果と副次効果の区別を示すフラッグを追加した4つ組にし、これに合わせて時間マップが事実の成立を保証する際の条件—すなわち、述語 tm.holds の定義—と各手続きでの effects 値の取扱いを変更するとともに、手続き7が意図効果が前提条件を充足するときにかぎり新しい弧を作成するように変更することにより実現できる。

<sup>\*</sup> たとえば、電車で移動する行為は行為者が移動先の場所に存在すること、電車代だけ所持金が減少することなどの効果を持つが、多くの場合、行為者は後者の効果を意図しない。

#### 4.9.2 前提条件と適用可能性条件

行為の遂行に成功するために遂行開始時間に成立している必要がある命題を一括して前提条件と呼んできたが、これも2種類に分類できる。遂行開始時間に成立していない場合に、成立させるために何かが行われる命題—狭義の前提条件—と、成立させるために何も行わず、行為自身が遂行されない命題—適用可能性条件<sup>☆</sup>—である<sup>3)</sup>。このように分類できる場合、手続き7により可能化関係を認識する際、狭義の前提条件だけを対象とする方が適切である。このような可能化関係の認識は、狭義の前提条件と適用可能性条件を分類してレシピに記述し、弧の preconditions 値の要素をこの区別を示すフラッグを追加した4つ組にし、これに合わせて各手続きでの preconditions 値の取扱いを変更するとともに、手続き7が意図効果が狭義の前提条件を充足するときにかぎり新しい弧を作成するように変更することにより実現できる。

#### 5. おわりに

本論文では、行為の効果と前提条件を取り扱うプラン解析手法を提案した。この手法は、これらを取り扱わない従来のプラン解析手法では不可能であった行為の効果による状態変化に依存するプラン解析と、これらを取り扱うプランニング手法との行為に関する知識記述の共有を可能にする。また、合理的行為者がある行為を遂行するために被観察行為列を遂行したという仮説をいったん設定すると、本手法により計算されたこの行為の効果はこの行為者が達成を意図していたことの候補を提供し、計算された前提条件は行為者が行為遂行前に成立していると信じている命題を提供する。

本論文のプラン解析手法は Sicstus Prolog 上のプログラムとして実現されている。このプログラムでは、各弧がさらに解析順序制御用のためのスコアと選好解釈選択用のスコアを格納するフィールドを持つ。前者はチャート法のアジェンダ機構で使用され、後者は解析候補が得られたときに、最も選好されるものを選択するのに使用される。これらは、すでに述べた選好規則を反映する数値や、弧の長さなどから計算される。

本論文のプラン解析手法は効果と前提条件の計算で否定を使用しているために、弧の effects 値と preconditions 値の要素が変数を含む場合には結果の適格性を示せない。このような場合を適格に取り扱えるようにすることが今後の課題の1つである。

本論文のプラン解析手法は道案内対話の発話解釈に適用している。この応用では、単語を発話する行為を原子行為とし、文内の表現を発話することにより遂行される行為と複数の文を発話することにより遂行される行為を一様に取り扱う対話プランモデル<sup>11)</sup>を使用している。この応用で発話により生じる対話参加者の心的状態の変化に依存する発話解釈に適用できることを確認した。

謝辞 日頃よりご指導いただく石井健一郎情報科学研究部長、有益な示唆をいただく対話理解研究グループの諸氏に感謝の意を表する。

#### 参考文献

- 1) Allen, J.: *Natural Language Understanding*, 2nd edition, Benjamin/Cummings (1995).
- 2) Allen, J.F. and Perrault, C.R.: Analyzing Intention in Utterances, *Artificial Intelligence*, Vol.15, No.3, pp.143-178 (1980).
- 3) Carberry, S.: *Plan Recognition in Natural Language Dialogue*, MIT Press (1990).
- 4) Dean, T.L. and McDermott, D.V.: Temporal Data Base Management, *Artificial Intelligence*, Vol.32, No.1, pp.1-55 (1987).
- 5) 堂坂浩二, 島津 明: タスク指向型対話における漸次的発話生成モデル, 情報処理学会論文誌, Vol.37, No.12, pp.2190-2200 (1996).
- 6) Earley, J.: An Efficient Context-Free Parsing Algorithm, *Comm. ACM*, Vol.13, No.2, pp.94-102 (1970).
- 7) Fikes, R.E. and Nilsson, N.J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artificial Intelligence*, Vol.2, No.3/4, pp.189-208 (1971).
- 8) 飯田 仁, 有田英一: 4階層プラン認識モデルを使った対話の理解, 情報処理学会論文誌, Vol.31, No.6, pp.810-821 (1990).
- 9) Kautz, H.A.: A Circumscriptive Theory of Plan Recognition, *Intentions in Communication*, Cohen, P.R., Morgan, J. and Pollack, M.E. (Eds.), chapter 6, pp.105-134, MIT Press (1990).
- 10) Kay, M.: Algorithm Schemata and Data Structures in Syntactic Processing, Technical Report CSL-80-12, XEROX Palo Alto Research Center, Palo Alto, CA (1980).
- 11) Kogure, K., Shimazu, A. and Nakano, M.: Recognizing Plans in More Natural Dialogue Utterances, *Proc. 1994 ICSLP*, pp.935-938 (1994).
- 12) Lambert, L.: Recognizing Complex Discourse Acts: A Tripartite Plan-Based Model of Dialogue, PhD Thesis, University of Delaware

<sup>☆</sup> たとえば、ある命題の真偽に関する質問を誠実に行う行為には話者が命題の真偽を知らないという条件があるが、話者は知っている場合に忘れるための行為を行わない。

- (1993).
- 13) Litman, D. and Allen, J.F.: A Plan Recognition Model for Subdialogues in Conversations, *Cognitive Science*, Vol.11, No.2, pp.163-200 (1987).
  - 14) McCarthy, J.: Circumscription: A Form of Non-Monotonic Reasoning, *Artificial Intelligence*, Vol.13, No.1/2, pp.27-39 (1980).
  - 15) Pollack, M.E.: Plans as Complex Mental Attitudes, *Intentions in Communication*, Cohen, P.R., Morgan, J. and Pollack, M.E. (Eds.), chapter 5, pp.77-104, MIT Press (1990).
  - 16) Ramshaw, L.A.: A Three-Level Model for Plan Exploration, *Proc. 29th ACL*, pp.39-46 (1991).
  - 17) 島津 明, 小暮 潔, 川森雅仁, 堂坂浩二, 中野幹生: 対話処理システムにおける内的コミュニケーション, 言語処理学会第2回年次大会発表論文集, pp.333-336 (1996).
  - 18) Shoham, Y.: *Artificial Intelligence Techniques in Prolog*, Morgan Kaufmann (1994).
  - 19) Sidner, C.L.: Plan Parsing for Intended Response Recognition in Discourse, *Computational Intelligence*, Vol.1, No.1, pp.1-10 (1985).
  - 20) Tomita, M.: *Efficient Parsing for Natural Language*, Kluwer (1986).
  - 21) Vilain, M.: Getting Serious about Parsing Plans: a Grammatical Analysis of Plan Recognition, *Proc. 8th AAAI*, pp.190-197 (1990).

(平成9年2月28日受付)

(平成9年10月1日採録)



小暮 潔 (正会員)

昭和32年生。昭和54年慶應義塾大学工学部電気工学科卒業。昭和56年同大学院修士課程修了。同年より日本電信電話公社武蔵野電気通信研究所, ATR自動翻訳電話研究所, 日本電信電話株式会社基礎研究所に勤務。平成9年より日本電信電話株式会社コミュニケーション科学研究所に勤務。主幹研究員。自然言語処理などの研究に従事。人工知能学会, 電子情報通信学会, 日本音響学会, 日本認知科学会, 言語処理学会, ACL各会員。



島津 明 (正会員)

昭和23年生。昭和46年九州大学理学部数学科卒業。昭和48年同大学院修士課程修了。同年より日本電信電話公社武蔵野電気通信研究所および日本電信電話株式会社基礎研究所に勤務。平成9年より北陸先端科学技術大学院大学情報科学研究科に勤務。推論手法, 形態素解析, 構文解析, 意味解析, 文生成, 質問応答システム, 機械翻訳システム, 対話システム等の研究に従事。工学博士。言語処理学会, 計量国語学会, 電子情報通信学会, 人工知能学会, ACL, ACM各会員。