

対話状況依存型プラン解析

小暮 潔 島津 明 中野 幹生

NTT 基礎研究所

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話者のプランを理解する必要があることから、プラン推論に関する多くのモデルが提案されている。観察された行為(観察行為)の列からプランを推論する戦略としてプラン解析(plan parsing)が Sidner⁴⁾により提案され、Vilain⁵⁾により形式化されている。プランは典型的にそれを行うための行為列として記述されるから、プラン認識は、観察行為が入力単語に、プランレシピ中の行為と分解(decomposition)の関係が文法規則に対応する解析過程と見なせる。

従来のプラン解析手法が主に取り扱っているのは行為の分解と抽象化であり、対話状態に関係する効果や前提条件は取り扱われていない。しかし、後者は次の点で重要である。(1) 言語的行為の効果は主に直接的に観察不能な対話従事者の心的状態の変化で、話者が言語的行為で意図していることの認識には効果の計算が必要である。(2) 言語的行為の前提条件も心的状態に関係し、これの計算により話者の背景信念を推論できる。また、従来のプラン解析手法には具体的に以下の限界がある。(a) 効果による行為の記述が不可能である。(b) 行為と、その前提条件を充足する先行可能化行為列の間の関連性を捉えられない。(c) 行為の未充足前提条件などに基づく曖昧性解消が不可能である。

本稿では、効果と前提条件を取り扱い、対話状態の変化、特に発話による対話従事者の心的状態の変化に依存するプラン解析手法を提案する。本手法はチャート解析を基盤にし、状態の時間的変化を捉えるために時間マップ¹⁾を使用する。本手法は対話理解システム³⁾の一部として Prolog で実現されている。

2 効果と前提条件を取り扱う必要性

対話理解において行為の効果と前提条件を取り扱う必要があるいくつかの典型的な状況を吟味する。

2.1 効果による行為の記述

プランレシピを記述する際、例えば、次のように分解中の行為を効果だけで指定できると便利である¹⁾。

- (1) 行為: `informref(S, H, Term, Proposition)`
 分解: `achieve(bel(H, Proposition))`
 効果: `belref(H, Term, Proposition)`
 制約: `parameter(Term, Proposition)`

形式 `achieve(P)` の記述は効果 P を持つ行為を示し、このレシピは効果 `bel(H, Proposition)` を持つ行為の遂行が `informref(S, H, Term, Proposition)` の遂行になることを表す。達成すべき効果を持つ多数の行為が存在するとき、この形式が許されなければ、それらすべてを列挙す

る必要が生じる。この形式には別の性質もある。形式 `achieve(P)` で示される構成要素行為を行為者が実行しようとする際、 P がその時点で成立しているならば、行為者が何も遂行しなくてもよいことを示す。対話で聴者が既に信じていると信じていることを話者が伝達しない可能性を考えると、この記述形式は重要である。この形式を可能にするには行為の効果の計算が必要である。

2.2 行為可能化行為

プランニング手続きは一般にゴールが与えられると、これを達成する行為(主行為)を探索する。この手続きは前提条件を持つ主行為を発見すると、前提条件を充足する行為を探索し、主行為の前提条件を充足する行為(主行為の可能化行為)と主行為から構成される行為列を返す。この形式の行為列が与えられたら、プランニングの逆過程であるプラン認識は、行為列全体を主行為のある効果(ゴール)を達成するための一連のものと思なすべきである。そのため、行為の効果と前提条件を取り扱う必要がある。

2.3 状態依存解釈

行為の効果と前提条件を取り扱えないと発話の解釈が困難なことがある。例えば、次の対話断片を考える。

- (2) 話者 1 厚木研究センタへの行き方を教えて下さい。
 話者 2 まずバスで吉祥寺に出て下さい。

一般に「下さい」で終る文の発話(表層言語行為型 `surface_request`) は依頼と解釈されるけれども、話者 2 の発話は厚木研究センタへ行くためのプランのステップの記述と解釈されるべきである。これは話者 1 の発話が話者 1 が厚木研究センタに行くためのプランを持つことを望んでいることを話者 2 に信じさせる効果を持ち、このような信念を話者 2 が持つときに、上の話者 2 の発話がプランのステップの記述と考えられるからである。この話者 2 の発話の解釈は次のレシピを用いて行える。

- (3) 行為: `describe_plan_step(S, H, Action, Plan)`
 前提条件: `bel(S, want(H, have_plan(H, Plan)))`
 分解: `surface_request(S, H, Action)`
 効果: `bel(H, step(Action, Plan))`
 制約: `step(Action, Plan)`

このレシピを用いるには、対話の初期状態と先行行為の効果から決定される対話状態と前提条件を取り扱えなければならない。

3 行為の効果と前提条件

3.1 行為の効果

行為の効果をもとにそのレシピが指定する効果と、存在すれば、構成要素行為の効果の和であると考えられる。例え

¹⁾このレシピは文献 2) のものの変形である。

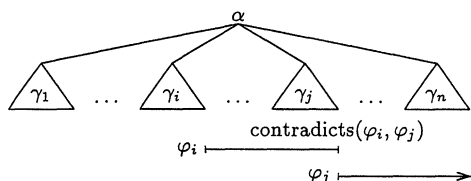


図 1: 行為の効果

ば, surface_request による describe_plan_step の遂行の効果は, レシピ3が指定する効果と, surface_request のレシピが指定する効果, surface_request の構成要素行為のレシピが指定する効果などなどの和である. 行為が時間区間を持つとすると, 行為の効果は行為終了時点には成立し, 無限に継続するか, あるいは, 矛盾する事実が成立する最初の時間まで継続するとモデル化できる. このとき, 構成要素行為 $(\gamma_1, \dots, \gamma_n)$ を持つ行為 α に関して, γ_i の効果 φ_i は, 矛盾する効果 φ_j — $\text{contradicts}(\varphi_i, \varphi_j)$ と書く—を持つ γ_i 以後の構成要素行為 γ_j が存在するとき, α 終了時点で成立せず (図 1), α 全体の効果とならないとし, そうでないとき, α 全体の効果であるとする.

3.2 行為の前提条件

行為の前提条件も, 効果と同様に, そのレシピが指定するものと, 構成要素行為の前提条件の和と捉える. ただし, 構成要素行為の前提条件は先行構成要素行為の効果により充足されることがある. 構成要素行為 $(\gamma_1, \dots, \gamma_n)$ を持つ行為 α を考える (図 2). 行為 γ_k の前提条件 φ_k は, 効果 φ_k を持つ先行構成要素行為 γ_i が存在し, φ_k が γ_k 開始時点まで継続するとき, α 開始時点で成立している必要がなく, α 全体の前提条件とならないとする (図 2 場合 (a)). 逆に, φ_k と矛盾する効果 φ_i を持つ γ_i が存在するとき, α 開始時点での φ_k の成立は, γ_k の前提条件の充足に貢献しない. ここで, γ_i と γ_k の間の γ_j が効果 φ_k を持ち, φ_k が γ_k 開始時点まで継続するとき (場合 (b)), 場合 (a) と同様に, φ_k は α 全体の前提条件とならない. しかし, そのような γ_j が存在しないとき (場合 (c)), φ_k は充足されず, 行為者が正しい行為を遂行すると仮定できるとき, そのような α は認識されるべきではない. また, γ_k の先行行為 γ_j の前提条件, あるいは, α のレシピが指定する前提条件を充足させると, γ_k の前提条件が充足できなくなるときも (場合 (d)), α は認識されるべきではない. 場合 (c) のように行為の効果と前提条件が干渉する場合を効果-前提条件衝突, 場合 (d) のように前提条件同士が干渉する場合を前提条件-前提条件衝突と呼ぶことにする.

4 プラン解析アルゴリズム

4.1 分解文法

プランレシピ中の行為とその分解の関係は文法規則と見なすことができる. 例えば, レシピ 3 の分解関係は, describe_plan_step(S, H, Action, Plan)

→ surface_request(S, H, Action)

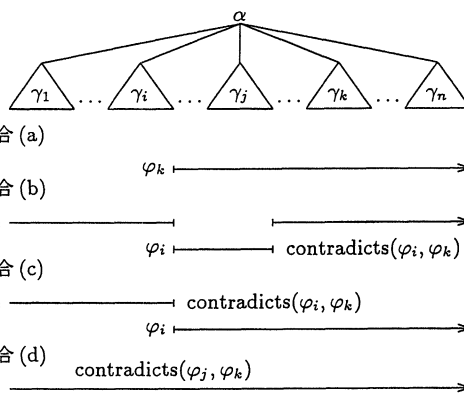


図 2: 行為の前提条件

で捉えられる. 分解関係のこのような解釈は分解関係を表す分解文法を与え, プラン認識への統語解析技術の適用を可能にする. この考え方を基盤に, 効果や前提条件も取り扱う手法を構成する. 以下の議論は他の解析手法にも適用できるが, 具体的にするために, 上昇型チャート解析に焦点を当てる.

4.2 効果と前提条件の計算

時間マップ

行為の効果による状態の時間的変化を捉えるために時間マップを用いる. 時間マップは (潜在) 事実トークンの集合と時間に関する順序から構成される. 事実トークンは 3 つ組 (t_1, t_2, φ) で, t_1 と t_2 は時点で, φ は事実記述 (ここでは項) である. 事実トークン (t_1, t_2, φ) は, φ が t_1 で成立し, t_2 か, あるいは, 矛盾する事実が成立する時点まで継続し続けることを表し, 後者の場合, 自然終了の前にクリップされるという. ここでは, 時点として整数を使用する. また, 無制限の継続を表すのに時点 \top を使用する. 例えば, 時間 t に終了する行為の効果 φ は (t, \top, φ) で表される. 時点として整数と \top を使用することから, 時間に関する順序として整数と最大要素 \top に関する通常的全順序を使用する. しかし, 事実トークンは終了時間が未確定な行為の効果を表すために用いられることがあり, このような場合には事実トークンは開始時点として未束縛変数を持つ. このような場合, 効果の開始時点がその行為の既知の構成要素行為の効果の開始時点以前でないことと, \top より前のことだけが判明しているので, 未束縛時点変数をこれらの条件を満たすように取り扱う.

事実トークン集合 \mathcal{F} の時間マップは, 次の条件が満たされるとき, 事実 φ が $[t_1, t_2]$ に成立することを保証すると言い, $\text{tm_holds}([t_1, t_2], \varphi, \mathcal{F})$ と書かれる. すなわち, (1) φ が潜在的に成立し, $[t_1, t_2]$ を取り囲む区間 $[t'_1, t'_2]$ が存在し, かつ, (2) このデフォルト継続性が t_2 以前にクリップされないときである.

行為の前提条件に関しても事実トークンと同様の 3 つ

(edge) 構造	
start	〈節点 (整数)〉
end	〈節点 (整数)〉
action	〈行為記述〉
rsubactions	〈行為記述列〉
constraints	〈制約の集合〉
effects	〈事実トークンの集合〉
preconditions	〈事実トークンの集合〉
aend	〈節点に束縛される変数〉
tree	〈木構造を表す列〉

図 3: edge 構造

組で表す。行為の開始時点が t であるとき、その前提条件 ψ は (t, t, ψ) で表す。

データ構造

チャート解析では、行為を弧で表す。弧で表される行為の効果と前提条件に関する情報を弧ごとに管理する必要があるから、図 3 の弧構造を用いる。弧 e の $start$ と end の値は節点で、弧で表される行為 (の部分) の開始時点と終了時点を示す整数である。 $action$ 値は行為記述で、ここでは項であることを仮定する。 $rsubactions$ 値は行為を認識するために同定する必要がある行為記述の列である。弧はこの値が空列のときに不活性弧、そうでないときに活性弧と呼ばれる。 $constraints$ 値は変数具体化に関する制約の集合である。 $effects$ 値と $preconditions$ 値はそれぞれ効果と前提条件を表す事実トークンの集合である。 $aend$ (action end) フィールドは行為の終了時点を示す、節点に束縛される変数を格納する。活性弧で表される行為の終了時点が未確定であるので、その効果の開始時点も未確定である。このような効果の情報を管理するために、開始時点が $aend$ 値の事実トークンを用いる。 $tree$ 値は解析木を表す列である。以下では、弧 e の $field$ 値を $field(e)$ で示す。

チャート手続き

上述の取り扱いにより、基本的に以下の手続きにより行為の効果と前提条件を計算できる。

手続き 1 (観察行為) α_j を j 番目の観察行為の記述とする。行為 β_r を持つレシピで、 α_j と β_r がレシピが指定する制約の集合 C_r を充足する最汎単一化子 θ を持つものそれぞれに関して、 $j-1$ から j への次のような不活性弧を作成する。すなわち、 $action$ 値が $\alpha_j\theta$ で、 $constraints$ 値が $C_r\theta$ 、 $effects$ 値が $\{(j, j, \varphi_r\theta) \mid \varphi_r \in E_r\}$ 、 $preconditions$ 値が $\{(j-1, j-1, \psi_r\theta) \mid \psi_r \in P_r\}$ 、 $tree$ 値が $\langle \alpha_j\theta \rangle$ のものである。ここで、 E_r と P_r はレシピが指定する効果と前提条件の集合である。

手続き 2 (上昇型予測) e_i を不活性弧とする。分解 $\langle \gamma_1, \dots, \gamma_n \rangle$ を持つ各レシピに関して、 $action(e_i)$ と γ_1 の単一化子 θ で、 E_r と P_r をレシピが指定する効果と前提条件の集合、 v_n を新規変数とすると、 $E = (effects(e_i))\theta \cup \{(v_n, \top, \varphi_r\theta) \mid \varphi_r \in E_r\}$ 、 $P = \{(t, t, \psi) \mid (t, t, \psi) \in (preconditions(e_i))\theta \wedge \neg tm_holds((t, t, \psi), E)\} \cup \{(start(e_i), start(e_i), \psi_r\theta) \mid \psi_r \in P_r\}$ で、効果-前

提条件衝突も前提条件-前提条件衝突も生ぜず、かつ、 $constraints(e_i)$ とレシピが指定する制約の集合 C_r の両方を充足する最汎な各 θ に関して、次のような弧 e を作成する。すなわち、レシピが指定する行為を β_r とすると、 $action$ 値が $\beta_r\theta$ 、 $rsubactions$ 値が $\langle \gamma_2\theta, \dots, \gamma_n\theta \rangle$ 、 $constraints$ 値が $(C_r \cup constraints(e_i))\theta$ 、 $effects$ 値が E 、 $preconditions$ 値が P 、 $aend$ 値が v_n 、 $tree$ 値が $\langle \beta_r, T_i \rangle\theta$ のものである。 e が不活性弧のとき、 v_n を end 値に束縛する。ここで、 T_i は $tree(e_i)$ である。

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、 $rsubactions(e_a)$ を $\langle \gamma_j, \dots, \gamma_n \rangle$ とする。 γ_j と $action(e_i)$ の単一化子 θ で、 $E = (effects(e_a) \cup effects(e_i))\theta$ 、 $P = \{(t, t, \psi) \mid ((t, t, \psi) \in (preconditions(e_a) \cup preconditions(e_i))\theta) \wedge \neg tm_holds((t, t, \psi), E)\}$ で、効果-前提条件衝突も前提条件-前提条件衝突も生ぜず、かつ、 $constraints(e_a) \cup constraints(e_i)$ を充足する最汎な各 θ に関して、次のような弧 e を作成する。すなわち、 $action$ 値が $(action(e_a))\theta$ 、 $rsubactions$ 値が $\langle \gamma_{j+1}\theta, \dots, \gamma_n\theta \rangle$ 、 $constraints$ 値が $(constraints(e_a) \cup constraints(e_i))\theta$ 、 $effects$ 値が E 、 $preconditions$ 値が P 、 $aend$ 値が v_n 、 $tree$ 値が $(T_a \cdot T_i)\theta$ のものである。 e が不活性弧のとき、 $aend$ 値を end 値に束縛する。ここで、 T_a と T_i はそれぞれ $tree(e_a)$ と $tree(e_i)$ であり、 \cdot は列の最後に要素を追加する演算子である。

以上の手続きにより、弧で表される行為の効果と前提条件を計算する基本的な手段を得た。以下では、第 2 節で述べた状況を取り扱うために、解析手続きを拡張する。

4.3 効果による行為の記述

記述形式 $achieve(P)$ の第 1 の側面、すなわち、効果による行為の記述は、手続き 2 と手続き 3 でレシピの分解、あるいは、活性弧の $rsubactions$ 値の第 1 要素がこの形式の記述であるとき、特殊扱いすることで実現できる。手続き 2 に関しては、次のように、この形式を取り扱う部分を追加し、拡張する。

- $\gamma_1 = achieve(P)$ の場合、 $tm_holds((end(e_i), end(e_i), P\theta), (effects(e_i))\theta)$ で、 E_r と P_r をレシピが指定する効果と前提条件の集合、 v_n を新規変数とすると、 $E = (effects(e_i))\theta \cup \{(v_n, \top, \varphi_r\theta) \mid \varphi_r \in E_r\}$ 、 $P = \{(t, t, \psi) \mid (t, t, \psi) \in (preconditions(e_i))\theta \wedge \neg tm_holds((t, t, \psi), E)\} \cup \{(start(e_i), start(e_i), \psi_r\theta) \mid \psi_r \in P_r\}$ で、効果-前提条件衝突も前提条件-前提条件衝突も生ぜず、レシピが指定する行為を β_r とすると、 $\beta_r\theta$ が e_i と同一観察行為を覆う構造が $(tree(e_i))\theta$ 中に存在せず、かつ、 $constraints(e_i)$ とレシピが指定する制約の集合 C_r の両方を充足する各最汎単一化子 θ に関して、手続き 2 と同様の弧を作成する。

同様に、手続き 3 も次の部分を追加し、拡張する。

- $\gamma_j = achieve(P)$ の場合、 $tm_holds((end(e_i), end(e_i), P\theta), (effects(e_i))\theta)$ で、 $E = (effects(e_a) \cup effects(e_i))\theta$ 、 $P = \{(t, t, \psi) \mid ((t, t, \psi) \in (preconditions(e_a) \cup$

$preconditions(e_i)\theta \wedge \neg tm_holds((t, t, \psi), E)$ であり、効果-前提条件衝突も前提条件-前提条件衝突も生ぜず、かつ、 $constraints(e_a) \cup constraints(e_i)$ を充足する各最汎単一化子 θ に関して、手続き3と同様の弧を作成する。

記述形式 $achieve(P)$ の第2の側面、すなわち、 P が成立しているときに行為が特に遂行されないことがあることは、この形式で指定される構成要素行為があるとき、これを構成要素行為から除去し、 P を前提条件にすることで取り扱える。そのために、次の手続きを導入する。

手続き4 (前提条件導入) e_a を $rsubactions$ 値が $(\gamma_j, \dots, \gamma_n)$ の活性弧で、 $\gamma_j = achieve(P)$ であるとする。 $E = (effects(e_a))\theta$ で、 t を $end(e_a)$ とすると、 $tm_holds((t, t, P\theta), E)$ のときに $P = (preconditions(e_a))\theta$ 、他のときに $(preconditions(e_a))\theta \cup \{(t, t, P\theta)\}$ で、効果-前提条件衝突も前提条件-前提条件衝突も生ぜず、かつ、 $constraints(e_a)$ を充足する各最汎単一化子 θ に関して、 e_a と同一範囲を覆う次のような弧 e を作成する。すなわち、 $action$ 値が $(action(e_a))\theta$ 、 $rsubactions$ 値が $(\gamma_{j+1}\theta, \dots, \gamma_n\theta)$ 、 $constraints$ 値が $(constraints(e_a))\theta$ 、 $effects$ 値が E 、 $preconditions$ 値が P 、 $aend$ 値が $(aend(e_a))\theta$ 、 $tree$ 値が $(tree(e_a))\theta$ のものである。 e が不活性弧であるとき、 $aend$ 値を end 値に束縛する。

4.4 行為可能化行為

第2.2節で説明したような行為可能化行為の取り扱いはいは次の手続きで行える。

手続き5 (前提条件吸収) e_1 と e_2 を隣接不活性弧とする。 e_1 の効果が充足する e_2 の前提条件の集合が空ではなく、 $E = (effects(e_1) \cup effects(e_2))\theta$ 、 $P = \{(t, t, \psi) | ((t, t, \psi) \in (preconditions(e_1) \cup preconditions(e_2))\theta) \wedge \neg tm_holds((t, t, \psi), E)\}$ で、効果-前提条件衝突も前提条件-前提条件衝突も生ぜず、かつ、 $constraints(e_1) \cup constraints(e_2)$ を充足する各最汎単一化子 θ に関して、 $start(e_1)$ から $end(e_2)$ への次のような不活性弧 e を作成する。すなわち、 $action$ 値が $(action(e_2))\theta$ 、 $constraints$ 値が $(constraints(e_1) \cup constraints(e_2))\theta$ 、 $effects$ 値が E 、 $preconditions$ 値が P 、 $tree$ 値が $(\alpha_2, T_1, T_2)\theta$ のものである。ここで、 α_2 は $action(e_2)$ で、 T_1 と T_2 はそれぞれ $tree(e_1)$ と $tree(e_2)$ である。

4.5 状態依存解釈

行為の直前の状態は初期状態と先行行為の効果により決定されるから、状態依存解釈を行うためには両者の影響を捉えなければならない。初期状態の影響を捉えるために、初期状態をもたらず特殊な弧を導入する。

手続き6 (初期状態導入) Φ を初期状態を表す事実記述の集合とする。0から0への次のような不活性弧を作成する。すなわち、 $action$ 値が $initialize$ で、 $effects$ 値が $\{(0, 0, \varphi) | \varphi \in \Phi\}$ で、 $constraints$ 値と $preconditions$ 値が空集合で、 $tree$ 値が $(initialize)$ のものである。

この手続きで作成される弧は、0からの不活性弧 e_i の前提条件を充足するとき、手続き5でこの弧と組み合わせられ、結果として初期状態を反映する弧が作成される。

効果と前提条件の計算は、観察行為の状態に依存した解釈を可能にする。例えば、「同一観察行為列を覆う任意の2つの不活性弧が表す行為に関して、未充足前提条件数が少ない方を選好する」というような選好規則を定義できる。このような状態に依存した選好規則は、第2.3節で述べたような状態に選好される解釈の選択を可能にする。

4.6 プラン解析の再初期化

プラン解析での初期状態の情報の利用は、長い対話の新しい処理方法を与える。対話状態の情報を用いない従来のプラン解析では、発話の先行発話に依存した解釈を行うために、対話の最初から問題の発話までの観察行為列全体を1つのプランとして解析している。しかし、入力の高さは処理時間に大きな影響を与える。文脈自由文法の形式でプランを取り扱う場合でも、処理時間は入力の3乗に比例する。この問題に関して、効果の計算と初期状態の利用により以下が可能になる。すなわち、対話の開始時点の初期状態と対話の途中のある時点までの言語的行為の効果からその時点での状態を計算し、これを新たな初期状態として、この時点以降の観察行為を解析すること、すなわち、プラン解析の再初期化である²。この再初期化により、観察行為列の短縮による解析の効率化が期待できる。

5 おわりに

本稿では、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。この手法は従来のプラン解析手法では不可能であった、言語的行為による対話状態の変化に依存したプラン解析を可能にする。また、行為者がある行為を遂行するために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る効果と前提条件は、この行為者が達成を望んでいた命題や、行為者が成立を信じていた命題を推論するために使用できる。

参考文献

- [1] Dean, T. L. and McDermott, D. V.: Temporal Data Base Management, *Artificial Intelligence*, Vol. 32, No. 1, pp. 1-55 (1987).
- [2] Litman, D. and Allen, J. F.: A Plan Recognition Model for Subdialogues in Conversation, *Cognitive Science*, Vol. 11, pp. 163-200 (1987).
- [3] 中野幹生, 島津明, 小暮潔: 分散協調処理による自然言語解析, *コンピュータソフトウェア*, Vol. 12, No. 5, pp. 33-44 (1995).
- [4] Sidner, C. L.: Plan Parsing for Intended Response Recognition in Discourse, *Computational Intelligence*, Vol. 1, No. 1, pp. 1-10 (1985).
- [5] Vilain, M.: Getting Serious about Parsing Plans: a Grammatical Analysis of Plan Recognition, *Proc. 8th AAAI*, pp. 190-197 (1990).

²これは、対話において前に正確にどのような表現が用いられていたのかは憶えていないが、大体どのような情報が伝達されたのかを憶えているということに相当するかもしれない。