ロボットのプランニング

白井英俊 (中京大学工学部電気電子工学科)

2014 年 4 月 11 日

計算機のプログラムとロボットのプランニングを比較することで、プランニングについて紹介しよう。

計算機のプログラム作成では、入力が大抵の場合数値で与えられており (例えば、$x = 1.0$, $y = 3.0$)、ある計算の方法 (アルゴリズム) に従って計算が行われ、結果を得る。その計算の方法は人間が指定し、必要ならば、自分が過去に作ったり、他の人が作ったりしたライブラリ関数を用いて計算が行われる。

ロボットのプランニングの場合、入力はロボットが自力で行う。つまりロボットが現在いる環境のデータを集めてデータベース化する (得られたものを「知識ベース」という)。例えば、「ドア 1 がある」「机 1 がある」「本 1 がある」「ドア 1 は開いている」「ドア 1 の 1 m 前に机 1 がある」「机 1 の上に本 1 がある」「ドア 1 の向こう側に机 2 がある」というデータが、ロボットがいる環境を特徴付けるデータとなる。これを**状態の表現**という。このようにロボットは数値データだけではなく「論理データ」を扱う。

次に、ロボットが何をすべきかは、人間によって与えられる。それも論理データの形を取ることが多い。例えば、「本 1 が机 2 の上にある状態にせよ」というようなものである。これは**目標の表現**といい、ひとつの論理データだけとは限らない。例えば、「本 1 が机 2 の上にある」「ドア 1 は閉じている」というように複数の論理データが指定されていても良い。

ロボットは、現在の状態の表現を手がかりにして、目標の表現が成り立つような状態に世界を変更する。そのために、ロボットはいくつかの「操作」を知っており、それを組み合わせて行為を行う。これはいわば計算機のプログラムにおいて、ライブラリ関数を組み合わせて計算をするようなものである。ただし、どのように組み合わせれば目標が達成されるかは簡単にはわからない。つまり予め人間がアルゴリズムを与えることはできない。そこで、ロボットは、現在の状態と、目標の状態、および知っている操作から、目標を達成する操作列を計算し、それを実行する。これはいわば、自動プログラミングのようなものである。

1 具体例：STRIPS

STRIPS は、プランニングのために設計された言語の一つで、最も有名なものである。STRIPS による行為の記述を示そう。行為は、次の 3 つの要素からなる:

- 行為の名前とパラメター (引数) リスト。
  例: Fly(p, from, to)
- 前提条件 (PreCond): 行為が実行できるために真でなければならない条件。
  例: At(p, from) $\wedge$ (from $\neq$ to)
- 効果 (Effect): 行為が実行された時に、状態がどのように変化するかを記述するもの。効果中で負 ($\neg$) がついた状態は偽となり、そうでない状態は真となる。
  例: $\neg$ At(p,from) $\wedge$ At(p,to)

ここで、現在の状況 (Init) と目標状況 (Goal)、それに空港間で荷物を空輸するための 3 つの行為の記述を紹介する。どのようにしたら、目標状況が達成されるか、行為の列を考えてみよう。

Initial State: At(C_1, SFO) ∧ At(C_2, JFK) ∧ At(P_1, SFO) ∧ At(P_2, JFK) ∧ Cargo(C_1)
∧ Cargo(C_2) ∧ Plane(P_1) ∧ Plane(P_2) ∧ Airport(JFK) ∧ Airport(SFO)

Goal State: At(C_1, JFK) ∧ At(C_2, SFO)

Actions:

Load(c,p,a)

PreCond: At(c,a) ∧ At(p,a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)

Effect: ¬ At(c,a) ∧ In(c,p)

Unload(c,p,a)

PreCond: In(c,p) ∧ At(p,a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)

Effect: At(c,a) ∧ ¬ In(c,p)

Fly(p,from,to)

PreCond: At(p,from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)

Effect: ¬ At(p,from) ∧ At(p,to)

2 有名な問題: モンキーバナナ問題

研究室にサルがいるとする。このサルはバナナを食べたがっている。研究室には 3 つの場所 A、B、C がある。最初、サルは A にいる。C には箱が置いてある。B にはバナナが天井からつるしてある。つまり、サルは箱を使ってバナナを取らなければならない。

Initial State: At(A) ∧ Level(low) ∧ BoxAt(C) ∧ BananasAt(B)

Goal State: Have(Bananas)

Actions:

Move(X, Y)

PreCond: At(X) ∧ Level(low)

Effect: ¬ At(X) ∧ At(Y)

ClimbUp(Location)

PreCond: At(Location) ∧ BoxAt(Location) ∧ Level(low)

Effect: Level(high) ∧ ¬ Level(low)

ClimbDown(Location)

PreCond: At(Location) ∧ BoxAt(Location) ∧ Level(high)

Effect: Level(low) ∧ ¬ Level(high)

MoveBox(X, Y)

PreCond: At(X) ∧ BoxAt(X) ∧ Level(low)

Effect: BoxAt(Y) ∧ ¬ BoxAt(X) ∧ At(Y) ∧ ¬ At(X)

TakeBananas(Location)

PreCond: At(Location) ∧ BananasAt(Location) ∧ Level(high)

Effect: Have(bananas)

空輸問題の解答: (これは一例) {Load(C_1,P_1,SFO), Fly(P_1,SFO,JFK), Unload(C_1,P_1,JFK)
Load(C_2,P_2,JFK),Fly(P_2,JFK, SFO), Unload(C_2,P_2,SFO) }