

初等力学の問題を解くシステム

有居 正仁 志村 正道

(東京工業大学工学部)

1 まえがき

人工知能は、人間の思考過程および認識過程を明確にすることと、人間に近い知的能力を持つ機械を作ることとを目的とする研究分野であり、具体的には、計算機に人間の知的能力のすぐれた機能を与えることによって、従来人間にしか行なえなかったことを計算機に代行させようとするものである。本論文では、人間の知的能力の1つである“物理の問題を解く能力”を計算機に与えることを課題としてとりあげ、自然言語で記述された物理の問題を解いてその答を出力する“物理の問題解決システム”の実現方法について考察する。

自然言語で記述された物理の問題を解くシステムに関する研究には [Novak 77] や [Bundy 79] 等がある。[Novak 77] において述べられている I S S A C は、問題の内部表現として標準的対象フレームを用いており、剛体の静力学の問題を解くことができる。[Bundy 79] において述べられている M E C H O は、探索を制御するためにメタレベル推論を行ない、理想的な滑車の問題等の力学の問題を解くことができる。しかし、いずれのシステムも問題で仮定される世界における対象の状態は一定でなければならないため、扱うことができる問題が限られてしまう。

本論文で述べるシステムは、次のような特徴を持つ：

- ・対象の動作や対象の状態に関する情報を表現するために action-model を用いる。
- ・経験的な知識に基づいた定性的な推論を行なう。
- ・運動方程式を作成するために対象の状態に応じた物理モデルを作成する。

システムは、これらの特徴によって問題例で示すような、問題で仮定される世界における対象の状態が変化する場合も扱うことができる。

2 システムの概要

一般に計算機が何らかの問題を解くためには、問題を理解すること（問題解釈）と問題を解くこと（問題解決）が必要である。システムが問題を理解するとは、システムが問題を解くのに必要な情報を適切な形で得ることである。システムは問題を理解するために、問題文の自然言語処理を行なう。自然言語処理には構文解析と意味処理がある。本システムでは構文解析のためのパーザとして N 進木拡張 L I N G O L [畝見 80] を使用している。システムは構文解析を行なうので比較的柔軟な自然言語文の処理を行なうことができる。意味処理は構文解析によって得られた構文解析木を評価することによって、問題文を内部表現へ変換するものである。システムは意味処理によって問題の内部表現である action-model、static-model（対象の属性と性質に関する情報及び対象間の属性に関する情報を表現する）、goal（問題において要求されている物理量を表現する）を得る。

問題によっては、問題文の自然言語処理によって得られた action-model では、システムが問題を解くためには不十分な場合もある。このような場合には、システムは自

然言語処理によって得られたaction-modelについて定性的な推論を行ない、問題を解くために必要なaction-modelを得る。システムは問題解釈を終了すると問題解決を行なう。システムの問題解決は、手段-目的解析(means-end-analysis)に基づく制御戦略によるもので、具体的には、運動方程式(物理モデル)を作成したり、等加速度運動の公式を用いたり、エネルギー保存則を用いたりすることによってgoalを達成する。

システムが解くことができるのは、質点の静力学の問題、質点の運動学の問題、滑車の問題等の高校初年級レベルの力学の問題である。実際にシステムでとり扱う問題の具体例を図1に示す。

質量4 kgの物体1をposition1 に置いたところ滑り出した。
面1の傾きは45度 面2の傾きは0度である。
物体1と面1の運動摩擦係数は0.2 である。
物体1と面2の摩擦は無視できるものとする。
物体1がposition2 に到達したときの速度はいくらか？

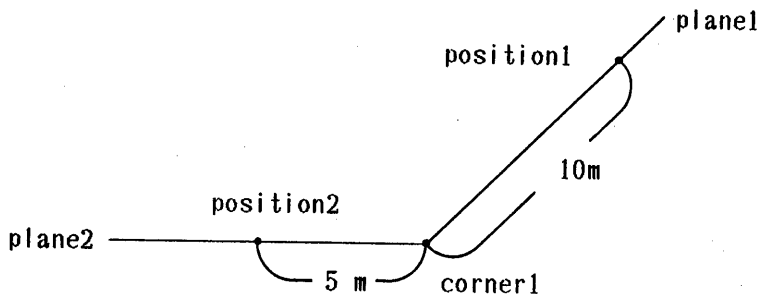


図1：問題例

3 問題の内部表現

計算機を用いて問題解決を行なうためには、問題を表現するための適切な内部表現方法を決定する必要がある。問題の内部表現は、問題の持つ情報を正しく反映したものでなければならない。物理の問題は、一般に、架空の世界における特定の状況を設定し、その状況における対象の動作ならびに対象の状態に関する情報と、対象の属性と性質に関する情報及び対象間の属性に関する情報を与えて、何らかの質問をするという形態をとる場合が多い。ここでは、問題の内部表現について述べる。

3.1 event

event とは、時間的に点(point)として表現できる出来事であり、対象の瞬間的な動作に関する情報を表現するものである。event には、自然言語処理によって生成されるもの(assert)と定性的推論によって生成されるもの(model)がある。またevent には、次の3つのタイプがある：

1) normal

event の前後においてstate が異なるもの(衝突を除く)

例 “物体1をposition1 に置く。”

2) special

event の前後においてstate は同一であるもの

例 “物体 1 が position2 を通過する。”

3) collision

衝突を表現するevent

例 “物体 1 と物体 2 が衝突する。”

event はevent フレームによって表現される。<event>スロットは、event そのものを表現し、対象の瞬間的な動作が記述される。<information>スロットは、そのevent における対象の速度や位置等に関する情報が記述される。<information>スロットは、定性的推論や問題解決において使用され、問題文の自然言語処理が終了した段階では、まだ満たされていない場合もある。

3.2 state

state とは、時間的に線(line)として表現できる状態であり、対象の連続的な動作や対象の状態に関する情報を表現するものである。state における対象の運動状態は一定であり、state の物理モデルは唯一に定まる。state には自然言語処理によって生成されるもの(assert)と定性的推論によって生成されるもの(model)がある。以下にstate の例を示す：

例 “物体 1 が面 1 を滑り落ちている。”

state はstate フレームによって表現される。<state>スロットは、state そのものを表現し、対象の連続的な動作や対象の状態が記述される。<information>スロットは、そのstate における対象の初速度や加速度等に関する情報や、そのstate における対象の運動状態（等速度運動、等加速度運動）等に関する情報が記述される。<information>スロットは、定性的推論や問題解決の際に使用され、問題文の自然言語処理が終了した段階では、まだ満たされていない場合もある。

3.3 action-model

action-modelは、対象の動作ならびに対象の状態に関する情報を表現するもので、event やstate の時間的前後関係を明示したものである。action-modelは、<pointer>スロットによって時間的前後関係を明示されたevent フレームとstate フレームによって構成されるフレームシステムである。図1で示した問題例の場合には、自然言語処理によって得られるaction-modelは、次の3つのevent とstate によって構成される：

event1: “物体 1 を position1 に置く。”

state2: “物体 1 が (面 1 を) 滑る。”

event3: “物体 1 が position2 に到達する。”

問題例のaction-model（自然言語処理終了後）のフレームによる表現を図2に示す。

```

(event1 (self (value event))
  (kind (value assert))
  (type (value normal))
  (event (value (put object1 position1)))
  (pointer (next-state state2)
    (after-event event3)))
(state2 (self (value state))
  (kind (value assert))
  (state (value (slide object1)))
  (pointer (last-event event1)
    (after-event event3)))
(event3 (self (value event))
  (kind (value assert))
  (event (value (reach object1 position2)))
  (pointer (before-event event1)
    (before-state state2)))

```

図2：問題例のaction-model（自然言語処理終了後）

3.4 static-model

対象には、質量(mass)や長さ(length)といった属性がある。また、対象は、軽い(light)、なめらかである(smooth)といった性質を持つ。対象間には、運動摩擦係数(d-fric-coef)や反発係数(repul-coef)等の属性がある。static-modelは、これらの情報を表現するものである。これらの情報は、問題で設定された状況においてはstaticである。問題例の場合には、自然言語処理によって得られるstatic-modelは図4のようになる。

3.5 goal

goalは、問題において要求されている（物理）量は何であることを表現するものである。goalはgoalフレームによって表現される。問題例の場合には、自然言語処理によって得られるgoalは図3のようになる。

```

(goal (self (value goal))
  (target (object object1)
    (want veloc)
    (case event3)))

```

図3：問題例の自然言語処理によって得られるgoal

```

(plane1 (self (value plane))
  (inclination (value 45))
  (d-fric-coef (pointer d-fric-coef1))
  ...)
(plane2 (self (value plane))
  (inclination (value 0))
  (friction (pointer friction1))
  ...)
(object1 (self (value object))
  (mass (value 4))
  (d-fric-coef (pointer d-fric-coef1))
  (friction (pointer friction1)))
(d-fric-coef1 (self (value d-fric-coef))
  (value (value 0.2))
  (object (value object1 plane1)))
(friction1 (self (value friction))
  (value (value disregard))
  (object (value object1 plane2)))

```

図4：問題例の自然言語処理によって得られるstatic-model

4 定性的推論

われわれが物理の問題を解くときに、具体的に方程式をたてて問題を解く前に、経験的知識に基づいて‘おそらくこうなるであろう’といった定性的な推論を行なって、問題を解くのに必要な情報を得ることがある。例えば問題例の場合、われわれは次のように考えるであろう：

“物体1はposition1 に置かれて滑りだしたので、面1を滑り落ちてcorner1 に到達するだろう。corner1 で面1から面2へ移り、面2を滑ってposition2 へ到達するだろう。”

システムが物理の問題を解くときにもこうした定性的な推論を行なう場合がある。それは、問題文の自然言語処理によって得られたaction-modelでは問題を解くためには不十分な場合である。このような場合にはシステムは定性的な推論を行なって自然言語処理では得られないevent やstate に関する情報を補って、問題を解くために必要なaction-modelを得る。問題例の場合には、定性的推論終了後のaction-modelは、次の5つのcaseによって構成される：

event1： “物体1をposition1 に置く。”
state2： “物体1が面1を滑り落ちる。”

event4: “物体 1 が (corner1 で) 面 1 から面 2 へ移る。”
 state5: “物体 1 が面 2 を左向きに滑る。”
 event3: “物体 1 が position2 に到達する。”

問題例の action-model (定性的推論終了後) のフレームによる表現を図 5 に示す。

```
(event1 ...
  (event (value (put object1 position1)))
  (information (value (object1 position position1)
                      (object1 veloc 0)))
  ...)
(state2 ...
  (state (value (slide-down object1 plane1)))
  ...)
(event4 (self (value event))
  (kind (value model))
  (type (value normal))
  (event (value (transit object1 plane1 plane2)))
  (information (value (object1 position corner1)))
  ...)
(state5 (self (value state))
  (kind (value model))
  (state (value (slide-left object1 plane2)))
  ...)
(event3 ...
  (type (value special))
  (event (value (reach object1 position2)))
  (information (value (object1 position position2)))
  ...)
```

図 5：問題例の action-model (定性的推論終了後)

5 問題解決

求められている(物理)量を得るためには、システムは問題解決を行なわなければならない。システムの問題解決の制御戦略は手段-目的解析に基づくもので、システムは目標を達成するためにオペレータを適用する。それぞれのオペレータには達成できる目標、目標を達成するための手段、オペレータを適用するための前提条件、オペレータの選択条件がある。前提条件には、そのオペレータを適用するときに既知でなければならない(物理)量が記述されており、前提条件が満たされなければオペレータは適用できない。また、オペレータに選択条件がある場合には、選択条件が満たさ

れていない状態では、そのオペレータは選択されない。オペレータにはエネルギー保存則を用いるもの、運動方程式（物理モデル）を作成するもの、等加速度運動の公式を用いるもの等がある。システムはこれらのオペレータを用いて問題解決を行なう。問題例の場合には、問題解決過程は次のようになる：

goal（主目標）：event3におけるobject1 の速度を求めよ。
subgoal1（goalの下位目標）：state5におけるobject1 の初速度を求めよ。
subgoal2（subgoal1の下位目標）：event4におけるobject1 の速度を求めよ。
subgoal3（subgoal2の下位目標）：state2におけるobject1 の終速度を求めよ。
subgoal4（subgoal3の下位目標）：state2におけるobject1 の加速度を求めよ。
subgoal5（subgoal3の下位目標）：state2におけるobject1 の初速度を求めよ。
subgoal6（subgoal5の下位目標）：event1におけるobject1 の速度を求めよ。

問題例のaction-model（問題解決終了後）のフレームによる表現を図6に示す。

```
(event1 ...)  
(state2 ...  
  (information (value (object1 accel 5.54)  
                     (object1 i-veloc 0)  
                     (object1 f-veloc 10.5)))  
  (model (value model1))  
  ...)  
(event4 ...  
  (information (value (object1 position corner1)  
                     (object1 veloc 10.5)))  
  ...)  
(state5 ...  
  (information (value (object1 i-veloc 10.5)))  
  ...)  
(event3 ...  
  (information (value (object1 position position2)  
                     (object1 veloc 10.5)))  
  ...)
```

図6：問題例のaction-model（問題解決終了後）

6 物理モデル

システムは、問題解決の際に対象の加速度や対象に働く力等を求めるために運動方程式を作成することをオペレータとして選択する場合があるが、その際に物理モデル（state-model）を作成する。state-model はstate のモデルであり、運動方程式を

たてるためのモデルである。state-model において対象は物理対象としてモデル化される。物理対象とは質点や剛体等の実際の対象の特徴を理想化したもの、あるいは抽象化したものであり、対象がどの物理対象によってモデル化されるかは、それぞれの state における対象の状態によって決定される。したがって同じ対象が異なる state では異なる物理対象としてモデル化されることもありうる。例えば、次のような2つの state を考える：

state-1 : “球 1 が面 1 を滑っている。”

state-2 : “球 1 が面 1 を転がっている。”

state-1 の物理モデルにおいては、球 1 はおそらく質点としてモデル化されるであろうし、state-2 の物理モデルにおいては、球 1 はおそらく剛体球としてモデル化されるであろう。問題を正確に解くためには、適切な物理対象を選択することが必要である。もし不適切な物理対象が選択されると、解くことができない運動方程式が抽出されたり、間違った運動方程式が抽出されたりして正しい答が得られないからである。

state-model が作成されると運動方程式が抽出される。抽出された運動方程式については数式処理が行なわれ、求める値が得られる。なお本システムは連立一次方程式を解くことができる。

7 あとがき

簡単な力学の問題を解くことができるシステムについて述べた。システムが解くことができる問題の分野は、現在のところかなり限られたものである。今後より広範囲の分野の問題を解くシステムを考えるとときには、次のような問題を解決していかなければならないであろう：

・系を考察する能力の強化 ・数式処理能力の強化

また、システムがより知的なものとなるためには、図形を認識する能力とかより強力な自然言語処理能力を持つことが必要であることはいうまでもない。

なお、本研究は文部省科学研究費特定研究(1) 課題番号59118003によるものである。

参考文献

[Bundy 79] Bundy, A., Byrd, L., Luger, G., Mellish, C., Palmer, M. : Solving mechanics problems using meta-level inference, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, pp.1017-1027 (1979).

[Novak 77] Novak, G.S. : Representations of knowledge in a program for solving physics problems, Proceedings of the Fifth International Joint Conference on Artificial Intelligence, pp.286-291(1977).

[畝見 80] 畝見達夫 : L I N G O L の n 進木への拡張, 東京工業大学総合理工学研究科修士論文(1980).