

言語処理で生成する形式表現とシミュレータの接合による 大学入試物理の力学問題の自動解答

Simulation-based University Entrance Examination Physics Problems Solver
using Logical Form Generated by Language Processing

岩根 秀直 *1*2
Hidenao Iwane

横野 光 *1
Hikaru Yokono

岩ヶ谷 崇 *3
Takashi Iwagaya

五十嵐 健夫 *4
Takeo Igarashi

*1株式会社富士通研究所
Fujitsu Laboratories Ltd.

*2国立情報学研究所
National Institute of Informatics

*3サイバネットシステム株式会社
Cybernet Systems Co., Ltd.

*4東京大学
The University of Tokyo

We are developing a physics test solver using physics simulator. Our system supports to first translates problem text and images into logical form by language processing and computer vision techniques, and then rewritten into the modeling language of physics simulator. Finally, physics simulator solves the problem. In this paper, we describe our method to generate initial conditions of physics simulator from logical form.

1. はじめに

本稿は、国立情報学研究所が開始した人工知能プロジェクト「ロボットは東大に入れるか」[新井 12]において開発中の物理問題の解答システムの成果に関する内容である。大学入試物理の力学問題では、受験者は与えられた状況やそこで行われる操作を理解し問題を解く。これは、物理シミュレーションによって再現することができ、初期設定を手で作成すれば多くの力学問題を解くことができる。

横野らは物理シミュレータを用いた問題解答システム[横野 14, 横野 15]を提案している。図 1 はその処理手順を示す。システムは、最初に、自然文やイラストなどで記述される問題を入力として受け取り、その意味内容を表した形式表現に変換する。本システムでは、複数のシミュレータを使用することを想定しているため、一度、共通の形式表現に変換する。形式表現では、問題を物体とその属性、物体に対して行われる操作、物理現象、問題の解答形式などを意味する一階の論理式として表現する。形式表現は、言語処理により構築可能な範囲で設計している。次に、形式表現を元に、問題の状況を理解し、シミュレーションの実行に必要な条件を構築し、物理シミュレータが利用するモデリング言語に変換する。最後にシミュレーションを実行し、その結果を元に問題が要求する解答を構築する。

このシステムでは予め物理問題で良く現れる質点、床、バネ、糸などの要素をシミュレータ上でモジュールとして準備し、それらを利用して解くことを想定している。シミュレーションの実行には、モジュールとして何をを用いるか、モジュールの位置・角度・速度などの初期設定を行う。例えば、モジュールの位置はすべてのモジュールが要求し、3次元空間上の座標として与える必要があるが、問題文中には通常書かれていない情報である。モデリング言語への変換では、与えられた問題に対して、必要なモジュールを選択し、それらの初期設定が必要となり、形式言語とモデリング言語の間には大きなギャップがある。しかし、これまでは形式表現からシミュレーションの初期設定の生成は実現できておらず、図 1 のモデリング言語を手で作成して、評価しているような状態であった。そのため、自然

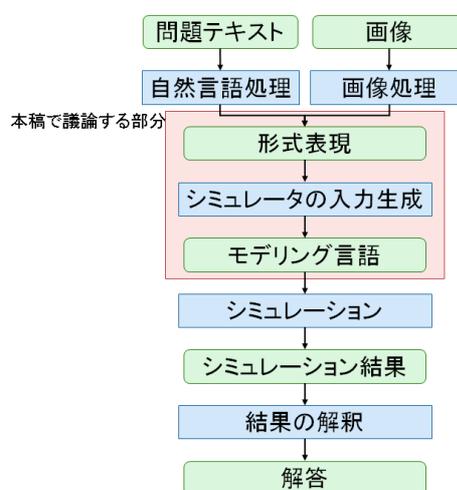


図 1: 力学問題自動解答器のフロー

言語処理との接合の実現可能性が見えていない状態だった。

本稿では、自然言語処理が生成可能として設計された形式表現を元に、シミュレーションに必要な初期条件の生成方法(図 1 において赤で囲まれた部分)について述べる。提案手法の評価は行った。評価の際には言語処理や画像処理は行っておらず、現状の言語処理技術の延長で生成できると考えられる問題のみを対象として、人手で形式表現を記述して求解を行った。また、本稿では扱っていないソルバーの開発も行っており、画像を入力とした静的な物理問題に対する自動解答[五十嵐 16](静的)なども利用している。

```

d
u,v,g
mass(A)
floor(Z)
horizontal(Z,e1) # Z は水平である
arai(Z,e1) # Z はあらい
dfc(A,Z,var2) # var2(e) は A と Z の間の動摩擦係数
assign(u,var2,e1) # var2(e1) = u
on(A,Z,e1) # A は Z の上にある
velocity(A,var3,var4)
assign(v,var3,e1)
slip(A,e1) # 滑る
stop(A,e2) # 止まる
position(A,var5)
assign(p1,var5,e1) # e1 での小物体の位置を p1 とする
assign(p2,var5,e2) # e2 での小物体の位置を p2 とする
assigndistance(p1,p2,d)
ga(g) # 重力加速度
-----
value(d) # 解答: v^2/(2*u*g)

```

図 2: 例 1 に対応する形式表現

2. 形式表現

本節では、形式表現について例を利用して説明する。以下は大学入試センター試験 2009 年物理 I 第 1 問, 問 1 を、距離を求める問題に修正したものである。

例 1 水平なあらい面上で小物体を初速度 v ですべらせた。小物体が静止するまでに進んだ距離 d を求めよ。ただし、小物体と床の間の動摩擦係数を u 、重力加速度の大きさを g とする。

図 2 は問題に対応する形式表現を手で構築したものである。形式表現では、床を表す `floor(Z)`、小物体（質点）を表す `mass(A)` などの物体の定義、`arai(Z, e1)`（あらい）、`horizontal(Z, e1)`（水平な）、`stop`（静止する）などの物理現象に関する述語、`dfc(A, Z, v)`（動摩擦変数）の変数宣言と `assign()` での代入操作による物理量の設定、`add_force`（力を加える）などの物体に対する操作に関する述語などから構成される。

`e1, e2` などはイベント変数と呼んでいるもので、各操作・状況がそれぞれの時刻で行われていることを表す。本稿では、イベント変数 e_i が発生した時刻も e_i で記述し、同一視する。時刻には、 $e_1 \leq e_2 \leq \dots$ の関係があるが、複数のイベントが同時に起こっている可能性があることに注意する。

種類	数
物体の定義	24
物理量 (1 項)	24
物理量 (2 項)	12
操作	13
物理現象	46



図 3: シミュレーションの初期条件の生成手順

3. シミュレーションに必要な属性値とその生成

本節では、図 1 の「シミュレータの入力生成」部であるシミュレーションの実行に必要な属性値の自動生成方法について述べる。表 2 にシミュレータ上に構築したモジュールの一部とそれらが要求する属性値の一覧を示す。これらはすべてシミュレーションの実行の際に呼び出し元で適切な数値を設定する必要がある。

例えば、例 1 において、小物体の位置（シミュレーション上の座標）や質量は記述されていない。また、動摩擦係数や重力加速度は変数で与えられているが、シミュレーション実行には具体的な数値を適切に設定しなければならない。

表 2: シミュレーション・モジュールとその属性の例

コンボ	属性
質点	初期位置, 初期速度, 質量
箱	初期位置, 初期速度, 大きさ, 質量, 摩擦係数
床	位置, 角度, 摩擦係数
バネ	バネ定数, 自然長, 両端点の接続先
力	力の大きさ, 力の向き, 作用点

図 3 にその処理手順を示す。本システムでは、物理問題の特性として、物理現象に変化があった場合にはその状況が必ず記述されると仮定する。例えば、小物体が時刻 e_i に床の上に置かれた場合には、「飛び出す」などの状況変化が記述されない限り、それ以降の時刻 e_{i+1}, e_{i+2}, \dots において小物体は床の上にあると仮定する。以下、例 1 を元に手順を説明する。

3.1 各時刻毎の条件生成

最初の段階では、形式表現を解析し、物体の属性がもつ条件を抽出し、構成する。各条件は、一階述語論理式で記述することを想定している。それぞれの物体は時刻毎に表 3 に示す属性を持つとし、の条件式を形式表現から構成していく。例えば、床は、位置 (x, y, z) を通る平面と定義しており、`on(A, Z, e1)` (A は Z の上にある) はダミー変数 t を用いて次のように変換する（床が水平の場合）。

$$A.x = Z.x + t_x \wedge A.y = Z.y \wedge A.z = Z.z + t_z$$

ここで、物体 P の時刻 e_1 における属性 a を $P_a^{e_1}$ とし、 x, y, z はそれぞれ属性 x 座標、 y 座標、 z 座標を表すとす。また、各時刻で成立する関係式もこの段階で生成する。例えば、運動方程式などを構築する。

表 4: 物理現象に対する意味記述の例

述語	変換後の論理式
$horizontal(A, e_i)$	$A_{angle}^{e_i} = 0$
$arai(A, e_i)$	$A_{dfc}^{e_i} > 0$
$slip(A, e_i)$	$ A_{velocity}^{e_i} > 0$
$stop(A, e_i)$	$ A_{velocity}^{e_i} = 0$
$assigndistance(A, B, d)$	$d \geq 0 \wedge (A.x - B.x)^2 + (A.y - B.y)^2 + (A.z - B.z)^2 = d^2$

表 3: 物体とその属性の例

コンボ	属性
質点	位置, 速度, 加速度, 加えられる力, 質量
箱	左下方の位置, 速度, 加速度, 加えられる力, 質量, 大きさ, 角度, 摩擦係数
床	位置, 角度, 摩擦係数
バネ	バネ定数, 自然長, 伸び, 左端点の位置, 角度
力	力, 接続先

3.2 イベント間の保存関係の生成

イベント間の保存関係を生成するために, まず, 時刻 e_{i-1} と e_i が同時 ($e_{i-1} = e_i$) に発生したを判定する. 本システムではイベント e_i において以下のような述語が含まれるかで判定する: cut (糸を切る), add_force/acceleration (力/加速度を加える), tidimeru (バネを縮める), release (手を離す), throwup/down (投げ上げる/下ろす), hikiageru/sageru (引き上げる/下げる). また, $e_i = e_{i+1}$ は, イベント e_i において以下の述語が含まれるかどうかで判定する: crash (衝突する). 例えば, 衝突するイベントでは, その直後の状況の説明が必ず記述される. 糸の切断や手を離す, 投げ上げるなどのイベントはその前の状況が記述され, その状態を保ったままイベントが行われるため, 同時刻に行われるとしたほうが都合がいい.

$e_{i-1} = e_i$ の場合には, その時刻間において, 物体の位置, 長さなどが変化しない. また, 変化したことが記述されない場合, 速度も変化しないと仮定する. 例えば, 衝突のイベントが発生した場合には, それに関連する物体以外では速度は変換しない. 衝突した物体の速度は通常変化するが, 運動量は保存するのでそれを条件として追加する.

$e_{i-1} \neq e_i$ の場合には, エネルギー保存などの関係式を条件として追加する.

また, 前の時刻において設定された条件が変化する述語がなければそれは継続しているとして, 条件を付加する.

システムは例 1 に対して 38 個の変数を用いた 79 個の条件を生成した.

3.3 シミュレーション開始時刻

実装上の課題であるが, 現状では複数のシミュレーション条件を設定する手段がない. 例えば, 例 1 で, 「速度が v' になったときに風が吹いた」というような問題には対応できない. そのため, シミュレーションの開始時刻は追加設定がない最後の時刻にする必要がある. また, 自然現象ではなく外部からの操作が加えられているもので, シミュレーションで扱うのは都合が悪いためそのようなイベントはシミュレーションの対象外としたい. そこで, 時刻 e_1 が以下の述語が含まれる時刻のうち最も遅い時刻をシミュレーションの時刻として採用する: cut (糸を切る), add_force/acceleration (力/加速度を加える), tidimeru (バネを縮める), release (手を離

す), throwup/down (投げ上げる/下ろす), hanareru(離れる), disconnect(切断する), drop(落とす).

3.4 不要なパラメータの除去

各物体の属性値のうち, 問題文中に出てこない変数を前節で生成した条件に数式処理技術を用いて消去する. 不要な変数には, 存在限量記号 \exists を付与して実閉体上の限量記号消去 [Tarski 51] により消去することができる. 大学入試物理の力学問題においては, 斜面を扱う場合などに角度 θ を扱う三角関数が現れるが, ほぼすべての問題で, ダミー変数 c, s を用いて, $c = \cos(\theta) \wedge s = \sin(\theta) \wedge c^2 + s^2 = 1$ の条件を用いて実閉体の論理式に帰着させることが可能である. 筆者らは, 限量記号消去ツール SyNRAC [Iwane 14] を拡張し, 三角関数が含まれる場合にも変数の消去を実現している. 多くの問題において, すべての変数の消去はできないことに注意する. 例えば, 例 1 における小物体 A の質量は, 問題文中に現れる変数 v, u, g では表現できないため, 最後まで残ることになる. 最終的に, 表 5 が得られる. そのため, 消去しない変数の選択がこの段階では重要な要素となる.

表 5: シミュレーションの初期条件 (変数)

物体	属性	値
A	質量	m where $m > 0$
A	初期速度	$\{uv, 0\}$ where $ w = 1$
A	初期位置	$\{Z_x + t, Z_y\}$
Z	角度	0
Z	位置	$\{Z_x, Z_y\}$
Z	動摩擦係数	u where $u > 0$

3.5 初期値設定

最後の段階で, 残った変数について適当に初期値を与える.

最初に, 問題文中で具体的な座標設定がない場合に平行移動しても一般性を失わないことを利用して, 原点を設定する. 例 1 の場合, システムは床 Z の位置を原点とした.

次に, 制約条件が残っていない変数の値を設定する. ここでは任意の値を設定できるが, 数値的に得られた結果が, 選択肢の式に代入して比較出来るようにするため, 異なる素数を設定する.

最後に制約条件が残っている変数の値を決定する. これは, 対象とする変数以外を存在量化し, 限量記号消去を用いて, 不要な変数を消去し, 得られた区間の中から任意に選択することで実現できる.

これらの操作により, 例 1 では, 表 6 が得られる.

4. ベネッセ模試タスクの結果

株式会社ベネッセコーポレーションのセンター試験模試「進研模試 総合学力マーク模試」を用いて, 本システムの評価を

表 6: シミュレーションの初期値

物体	属性	値
A	質量	7
A	初期速度	{5, 0}
A	初期位置	{13, 0}
Z	角度	0
Z	位置	{0, 0}
Z	動摩擦係数	3

行った．ここでは言語処理や画像処理は行っておらず，現状の言語処理技術の延長で生成できると考えられる問題のみを対象として，人手で形式表現を記述して求解を行った．物理シミュレータとして MapleSim を利用し，シミュレーション結果の解析は一部を除いて人手で行っている．

今年度のベネッセ模試タスクでは，シミュレーションを用いたソルバー（シミュ）の他，3.4 節の方法でシミュレータなしで解答を求めるもの（QE）や，本稿では扱わなかった，画像を入力とした静的な物理問題に対する自動解答 [五十嵐 16]（静的）なども利用した．シミュレーションのモジュールの整備なども行っており，昨年度に比べて対応できる問題の範囲も広がっている．

それぞれのソルバーでの回答数と解けなかった問題の理由を表 7 に示す．結果は，全体で 25 問中 16 問で正解し，62 点を獲得した．昨年度は，25 問中 10 問正解（42 点）のうち，シミュレーションによる回答は 3 問（13 点）であったので，モデリング言語から開始した場合に比べて，自動化の範囲を広げたタスク設定で同等以上の結果が得られた．解けなかった問題は，すべてソルバーが非対応なものであった．

表 7: ベネッセ模試タスク結果

	ソルバー	問題数	点数
解けた	シミュ	6	20
	QE	3	12
	静的	1	3
	その他	6	27
	合計	16	62
解けなかった (非対応)	分野		
	波	5	20
	つりあい	2	10
	円運動	2	8
	合計	9	38

5. おわりに

言語処理が構築可能と想定される形式言語を入力とし，シミュレーション実行までの自動化について述べた．現状では，言語処理や画像処理が生成した形式言語での評価ができていないだけでなく，解答の生成のためのシミュレーション結果の評価部には人の手を加える必要がある．また，対応できているのは力学の一部にとどまっているなど，完全な自動化へ向けては多くの課題が残っている．

参考文献

[Iwane 14] Iwane, H., Yanami, H., and Anai, H.: SyN-RAC: A Toolbox for Solving Real Algebraic Constraints,

in Hong, H. and Yap, C. eds., *Mathematical Software - ICMS 2014 - 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings*, Vol. 8592 of *Lecture Notes in Computer Science*, pp. 518–522, Springer (2014)

[Tarski 51] Tarski, A.: *A decision method for elementary algebra and geometry*, University of California Press (1951)

[横野 14] 横野 光, 稲邑 哲也: 論理演算と物理シミュレーションの結合による物理問題解答, 人工知能学会全国大会論文集, 第 28 巻 (2014)

[横野 15] 横野 光, 稲邑 哲也: シミュレーションによる物理問題解答のための問題分類, 人工知能学会全国大会論文集, 第 29 巻 (2015)

[五十嵐 16] 五十嵐 健夫, 横野 光, 岩根 秀直: 図形描画とテキスト入力を用いた力学に関する質問応答システム, in *24th Workshop on Interactive Systems and Software* (2016)

[新井 12] 新井 紀子, 松崎 拓也: ロボットは東大に入れるか? -国立情報学研究所「人工頭脳」プロジェクト-, 人工知能学会誌, 第 27 巻, pp. 463–469 (2012)