

状況とルールのパターンマッチングを学習する システムの構築と評価

Development and Evaluation of a System for Learning Pattern Matching of Situations and Rules

森田 純哉¹, 三輪 和久², 中池 竜一³, 寺井 仁^{2,4}, 齋藤 ひとみ⁵, 小島 一晃⁶

Junya Morita, Kazuhisa Miwa, Ryuichi Nakaike, Hitoshi Terai, Hitomi Saito, Kazuaki Kojima

¹ 北陸先端科学技術大学院大学, ² 名古屋大学, ³ 京都大学, ⁴ JST/CREST, ⁵ 愛知教育大学, ⁶ 早稲田大学

Japan Advanced Institute of Science and Technology, Nagoya University, Kyoto University, CREST JST, Aichi University of Education, Waseda University

j-morita@jaist.ac.jp

Abstract

This paper presents an approach for cultivating deep understanding of the nature of human cognition. Our approach uses a computational cognitive model to highlight the nature of human cognitive system. The learning support system provides several support functions for building a production system model. Through using the system, learners observe unintentional fires of rules caused by the pattern matching mechanism implemented in a production system. We consider these observations lead to learner's deep understanding of the nature of human cognition. In this paper, educational practices in which our system was used are presented.

Keywords — Production System, Education for Cognitive Science, Learning Support System

1. はじめに

人間の認知システムの重要な特性の一つに、知識利用の柔軟性がある。人間は、経験によって得られた知識を、その知識が得られた状況だけでなく、類似した状況、もしくはパターンの共通した状況へ適用させる。また、長期記憶中の具体的な経験は、適切なレベルに抽象化され、多様な状況へ転用されるようになる。このような知識利用の柔軟性により、人間は新たな環境に適応し、さらには新たな知識を創造することになる。

人間が通常保持する知識利用の柔軟性は、それが欠如した認知システムとの対比により際立つ。これまで、発達のプロセスにおける過剰般化、サヴァン症候群における具体的な事例への固執、チンパンジーの保持する写実的な記憶などが明らかにされてきた。これらの研究は、個別の研究対象の認知的特質だけでなく、通常の成人が意識せずに行う知識処理の特質を我々に気づかせてくれる。

人間の知識利用の柔軟性に気づくためのより手軽な方法は、計算機上に実現される認知モデルの利用である。認知モデルの伝統的なフレームワークとしては、プロダクションシステムを考えることができる。プロダクションシステムは、認知科学の草創期より用いられており、現在でも多くのシミュレーション研究に用いられている。

プロダクションシステムは明示的な制御構造をもたず、変数への値の代入が直接的になされない。変数への値の束縛は、ワーキングメモリ内での状況表現とIF節のパターンマッチングを介してなされる。この仕組みにより、プロダクションシステムにおけるルールは、単一の状況だけでなく、多様な状況へと適用されることになる。Anderson (2007) は、パターンマッチングのメカニズムこそ人間の認知の特殊性を再現するメカニズムであると述べている。

しかしながら、通常、初学者にとって、プロダクションシステムにおけるパターンマッチングを理解し、人間の知識利用を再現するモデルを構築することは容易ではない。プロダクションシステムのモデリングのプロセスでは、コーディングされたルールが、モデル構築者の意図しなかった状況においてしばしば発火する。つまり、モデル構築者の想定を超えた形で、ルールが「過剰適用」される。そのような意図せぬ発火を抑制するために、モデル構築者は、人間の知識利用の制約を探り、ルールの発火条件を修正していくことになる。

著者らは上記のようなプロダクションシステムにおけるモデリングのプロセスは、認知科学の初学者にとって、人間の知識利用の特質に気づく重要な契機となると考える。通常、パターンマッチングがモデル化する知識利用のプロセスは、人間によって明示的に意識されるものではない。よって、先述のような認知科学の研究を知らない初学者にとって、人間の知識利用の柔軟性は、通常は気づ

くことが困難であると考えられる。著者らは、プロダクションシステムのモデリングの中で、ルールの予期せぬ発火を観察し、ルールの発火条件を見つめ直して行くことによって、日常的には無意識にこなしている人間の知識利用の柔軟性に気づくようになると考えている。

本研究は、上記の考えのもとに開発されたプロダクションシステムによる認知モデリングの学習支援システム、およびそれを利用した教育実践を報告する。本研究で開発した学習支援システムは、具体的なコーディングスキルを養うものではなく、対象領域を限定したビジュアルプログラミング的環境となっている。また、プロダクションシステムの学習に不可欠なWMやルールの概念の学習を可能な限り効率化し、パターンマッチングの挙動を学習することに焦点を当てている。

2. システム

本研究において開発したパターンマッチングの学習支援システムは、中池ら(2011)によって開発されたどこでもプロダクションシステム(どこプロ)をベースとしている。どこプロは、Ruby on Railsを利用して開発された教育用プロダクションシステムである。Webアプリケーションとして実装され、学習者は教室、自宅を問わず同一の環境でプロダクションシステムの学習を進めることができる。

図1はシステム全体のスクリーンショットを示している。スクリーン右側に、中池ら(2011)が開発したどこプロが配置され、スクリーンの左側に、本研究において開発したシステムが配置される。2つのシステムは、共用のリレーショナルデータベースにアクセスする。データベースにはWMやルールが格納される。

なお、図1に示されるように、本研究において開発したシステムは、高校物理などで扱われる滑車の原理に関わる問題を対象領域とする。この課題は、プロダクションシステムモデルの古典的研究(Larkin and Simon, 1987)において扱われたものである。Larkin and Simon (1987)は滑車の問題に対するプロダクションシステムのモデルを示すことで、文章的思考に対する図的思考の効率性を議論した。彼らの研究のトピックが示すように、滑車の問題は、問題状況の視覚的な表現が容易な課題領域である。この課題領域の性質により、後述する問題状況の視覚的な表現が可能になった。以降、簡便化のため、本システムをMIPP (Modeling Interface for Pulley Problems)と呼んでいく。

2.1 MIPPの構成

MIPPは、「問題生成」、「ルール記録」、「実行」と呼ばれるWebページ(erbファイル)から構成される。それぞれのページには、MIPPの画面右上に表示されるリンクをクリックすることで移動できる。

3つのページは、共通して、WMに格納される表現を図示するイメージディスプレイが組み込まれている。イメージディスプレイに表示されるオブジェクトは、天井、2種類の滑車(動滑車、定滑車)、おもり、ロープである。それぞれのオブジェクトには、IDと値(張力、重さ)が表示される(「|」の左にID、右に値を示す)。

図2に、イメージディスプレイとWM中の記号表現の対応を図示した。WMの記述には、述語と引数からなるリストが用いられる。述語には、「Ceil」、「Pulley」、「Weight」、「Rope」、「Support」、「Value」の6つが用いられる。はじめ4つの述語は、Image Display上のオブジェクトの種類を区別するものである。「Support」はオブジェクト間の支持関係を示す。上部に配置されるオブジェクトを第1引数、下部に配置されるオブジェクトを第2引数とする。「Value」はオブジェクトの値の状態を表し、第1引数にオブジェクトのID、第2引数にオブジェクトの値を示す。

以下では、イメージディスプレイを組み入れた3つのページについて、それぞれの機能を説明していく。

2.1.1 問題生成

問題生成のページには、「オブジェクトの追加」、「削除」などのメニューが配置され、イメージディスプレイにオブジェクトを配置することができる(図3)。また、「値の設定」と表示されるプルダウンメニューとテキストボックスにより、問題解決開始前(初期状態)のオブジェクトの値を設定する。滑車の問題では、既知のオブジェクトの値をもとに、滑車の原理に関わる知識を適用し、他のオブジェクトの値を埋めていくことで問題の解決がなされる。

なお、イメージディスプレイにおいて上記の操作が行われるたびに、イメージディスプレイの状態が随時どこプロのWMに書き出されていく。新規に書き出されたリストは、どこプロのWM中で赤字で表現される。図3はobject7の値に100を設定した直後の画面の状態を示している。この操作と対応し、現在のWMに(Value object7 100)が赤字で表示されている。このように、MIPPはイメー

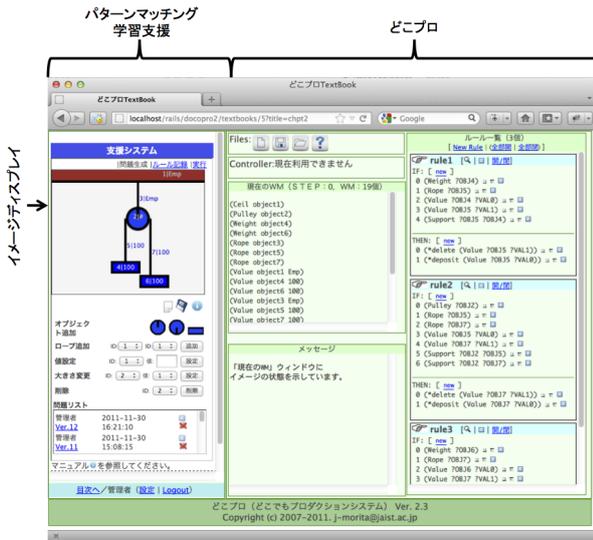


図 1 システムの画面構成

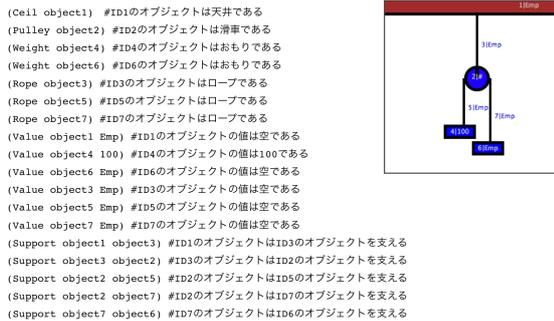


図 2 イメージディスプレイとWMの対応

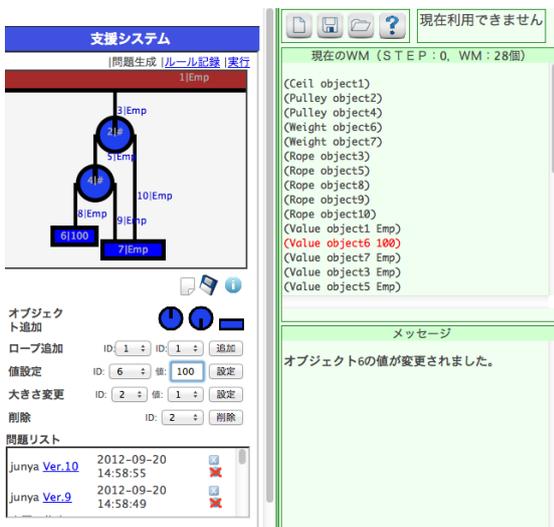


図 3 問題生成の画面

ジディスプレイの操作とWMの対応関係をフィードバックすることで、学習者に対して、状況を記号的に表現することの意味を効率的に教授することを狙っている。

問題生成により構築された問題は、MIPPのデータベース (Pulley data) に保存される (図3左下)。データベースは、ユーザ間で共通のものが用いられるため、授業における問題の共有が容易になる。

2.1.2 ルール記録

問題解決とは、「初期状態から目標状態に向けた状態変化の系列を発見すること」と通常は定義される。プロダクションシステムの枠組みでは、問題解決における各時点での状態はWMに格納され、状態の変化はルール (オペレータ) の段階的 (ステップごとの) 適用によって成し遂げられる。逆に考えれば、WMの状態 WM_t から WM_{t+1} への変化を導くルールは、条件節に WM_t 、実行節に WM_t と WM_{t+1} の差分を持つものとして構成されることになる。

ただし、実際には、 WM_t に含まれる全ての項目を、条件節に含むルールを考えることは妥当ではない。人間は直面する状況 WM_t の全てに注意をむけるわけではなく、そのうちの一部の要素のみを条件としてルールを適用させるものと考えられる。

また、ルールは、変数を含むか否かという観点からも区別できる。定数のみで構成されるルールは、個別の状況においてのみ発火する事例的ルールであり、変数を含むルールは多様な状況において発火する一般的ルールということができる。ルールに含まれる変数が増加すれば、その一般性は高まるが、適切ではない状況においてルールが発火する危険性も高くなる。

MIPPにおけるルール記録は、上記のようなプロダクションシステムにおけるルールの基本的性質を学習者に意識させつつ、簡便にルールを記述させることを支援するものである。

ルール記録において、学習者が行うことは、問題生成のページで構成されたイメージディスプレイを対象に、問題解決のプロセスを再現することである。学習者は予め設定された初期値に基づいて、連続的にオブジェクトの値を埋めていく。図3の問題を例とすれば、object6の初期値が100と与えられているので、まずはそれを条件にobject8に100を設定する。続いて、object8の値が100であるので、object9の値に100を設定する。残りのオブジェクトについても、滑車の原理に関わる知識を適用することで値を埋めていく。このような一連

の問題解決のプロセスを記録し、分節化（ステップに分割）することでルールが構成される。

図4に示される4つの画面は、1つのルールを記録する際の画面の変化を示している。はじめの画面のメッセージウィンドウに表示されるStartボタンをクリックするとルール記録が開始する。その後、ルールのIF節に含めるオブジェクトをイメージディスプレイ上の矩形（図4の3つめのスクリーンにおける点線の枠）、あるいは「対象オブジェクト」と表示されたチェックボックスをクリックすることで決定する。オブジェクト選択の結果は、逐次メッセージウィンドウに表示されるルールのひな形に反映される（選択されたオブジェクトに関わるWM中のリストがIF節に設定される）。オブジェクトの選択の後、選択範囲内のオブジェクトの値を変更する。イメージディスプレイに対する操作を反映し、値の変更を表現するTHEN節がメッセージウィンドウのひな形に表示される。FINISHボタンを押すことでひな形に表示されるルールがどこプロのデータベースに格納される。

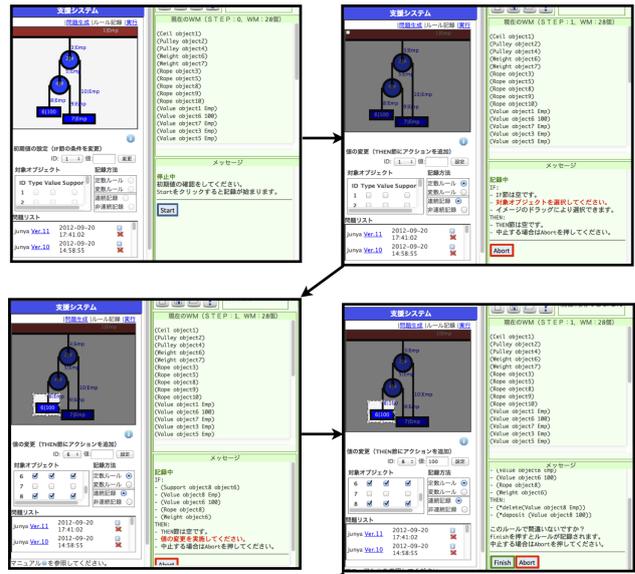
図5は、図4の手続きによって構成されるルールを示している。rule1はラジオボタンから「定数ルール」を選択することで構成されるルールであり、rule2は「変数ルール」を選択することで構成されるルールである。変数ルールは定数ルールに含まれるオブジェクト名（e.g., object1, object2）を機械的に変数（e.g., ?OBJ1, ?OBJ2）に置換したものである。

このようなルール記録の手続きを通してルールを構成することで、プロダクションシステムにおけるルール（状態変化のオペレータ）の概念的理理解が促されると考える。また、先述したように、変数ルールは記録されたままの状態では一般性が高すぎるため、適切には動作しない。学習者は、記録された変数ルールを足場かけ（スキヤフォールド）とし、適宜修正していくことで、問題を解決するモデルを構成することになる。

2.1.3 実行

実行ページは、どこプロの推論エンジンに接続し、推論を1ステップずつ進める。推論が上手く行けば、WM中に保持されるValueを述語とするリストが書き変わっていく。そして、WMの変化に対応する形で、イメージディスプレイ上のオブジェクトの値が変化していく（図4）。これにより、学習者は、構成されたルールによる問題解決のプロセスを視覚的に把握できる。

また、どこプロは、現在のWMの状態と各ルールの変数との間の束縛リストをメッセージウィンドウ



次のステップの記録

図4 ルール記録の画面。

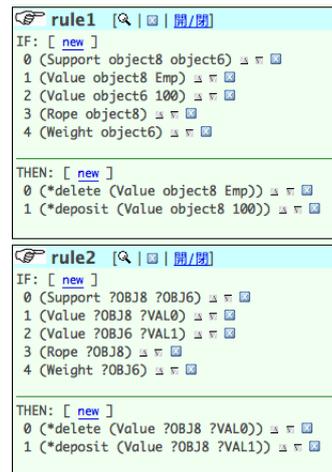


図5 記録されるルール。

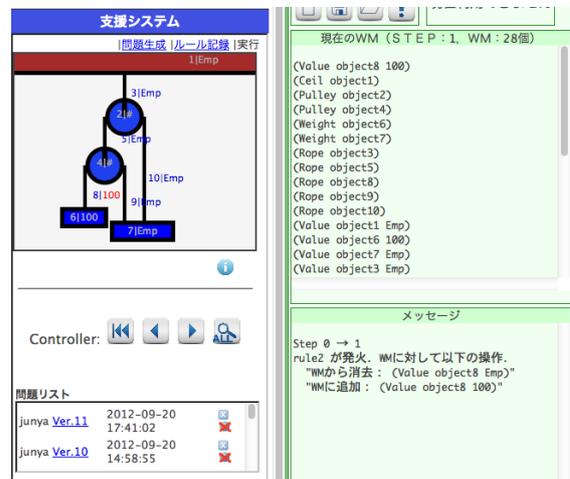


図6 実行画面。

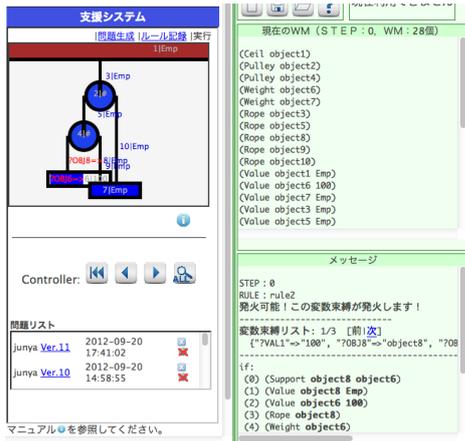


図7 変数束縛の表示．

に表示する機能を保持する(中池ら, 2011). MIPPでは, 束縛されている変数をイメージディスプレイ上のオブジェクトに対応付けて表示する. 図7は図5のrule2に登場する変数とイメージディスプレイ上のオブジェクトの束縛結果を示している. ルール中の?OBJ6がobject6と, 同じく?OBJ8がobject8に束縛されている. この束縛の結果は, 図4に示されたルール記録の手続きを正しく再現するものである. つまり, このルールがこの束縛で発火すれば, 結果としてobject8に100の値が埋められることになる.

しかし, 滑車の問題を対象としたモデルでは, 多くの場合で変数ルールが記録された状況とは異なる状況で発火することになる. 図8は, 記録時とは異なる形でルールが発火する状況を示している. 図8左に示される3つの状況は, 全てrule2が適用され, object8の値が埋まった後の問題の状態を示している. ルール記録においてなされた手続きを再現するのであれば, 図8右のrule3の適用によって, object9の値が100に変化しなければならない. しかし, 図中の3つの変数束縛はいずれもそのような問題解決プロセスを再現するものではない.

上段の束縛は, object8の値を埋めるために形成されたrule2が, object8の値が埋まった後に再度適用されてしまった状況を示している. どこプロにおける競合解消は, ルールリストの上部に配置されるルールが優先的に発火するものになっている. そのため, rule2を抑制しなければ, 本来発火させるべきrule3の発火がなされない. rule2の変数束縛のリストを見れば, ?VAL0と?VAL1に100の値が束縛されていることがわかる. よって, この状況におけるrule2の発火を抑制するためには, object8の値を示す?VAL0がEmpであり, object6の値を示す?VAL1がEmpではないことを明示する条件をIF節に付け加える必要がある. どこプロの書式では, それぞれ

- (*test-equal ?VAL0 Emp)
- (*test-not-equal ?VAL1 Emp)

と記述されることになる.

中段と下段はともに発火させたいrule3とのマッチングを示しているものの, 変数束縛のパターンが意図するものと異なっている. 中段では?OBJ8と?OBJ9という2つの変数がともにobject8に束縛されている. このような束縛を抑制するためには, 2つの変数が異なるものであることを示す条件をIF節に付け加える必要がある. この条件はどこプロの書式では,

- (*test-not-equal ?OBJ8 ?OBJ9)

のように記述される.

下段における束縛は, object4に束縛させたい?OBJ4がobject2に, object8に束縛させたい?OBJ8がobject10に, object9に束縛させたい変数?OBJ9がobject5に束縛されている. 滑車が2本のロープ支えているという状況が本来発火させたい状況とこの状況とで共通するためにこのような束縛が形成される. この束縛を抑制するためには, 上段と同じく?VAL0がEmpであり, ?VAL1がEmpではないことを明示する条件をIF節に付け加える必要がある.

上記のような意図せぬ変数束縛は, MIPPのルール記録によって形成される変数ルールで典型的に生じる現象である. このような現象が生じないルール記録機能を学習者に提供することは技術的には可能である. しかし, パターンマッチングの学習支援システムであるMIPPは, あえて意図しない変数束縛の結果を学習者に表示し, その原因を探求させ, ルールを修正することを求める. このプロセスをへることで, 暗黙的な人間の知識適用に対する内省的視点が生じ, 人間の知識利用の柔軟性の理解に繋がると考える.

3. 授業デザイン

授業における本システムの運用では, 3つの機能の循環的な利用を求める. 図9に授業デザインのフローを示す. 以下, 図9における各学習のステップを説明する.

1. 所与の問題: まず受講生に滑車の問題を画像ファイルで提示し, それを問題生成機能を用いて作らせる. イメージディスプレイへオブジェクトを配置し, オブジェクトに値を設定することで, 視覚的表現とWM表現との対応を学習する. なお, 後述の授業実践では, 所与の問題としては図3に示される問題を与えた.
2. 定数ルールの記録: 所与の問題を対象に, 定数ルールを選択させたとうえで, ルール記録を

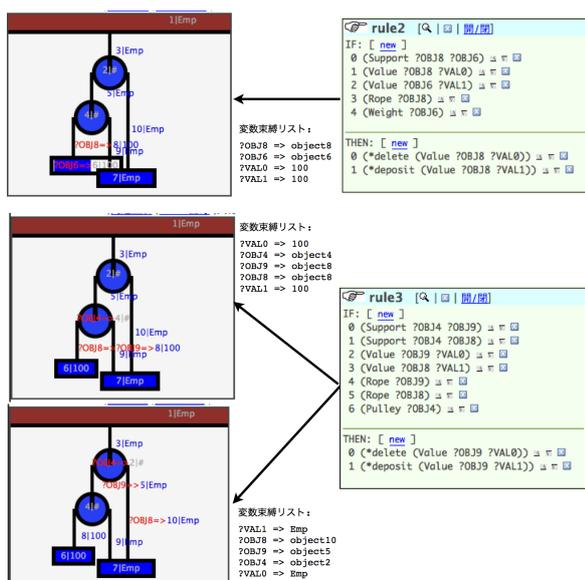


図 8 ルール記録と整合しない変数束縛。

実行する。これにより、問題解決の手続きとルールとの対応関係を学習する。

- 問題解決プロセスの再現を確認: 実行画面において定数ルールを実行する。実行の結果、記録された問題解決プロセスが再現されたことを確かめる。問題解決プロセスが再現されなかった場合、再度、定数ルールを記録する。
- 変数ルールの記録: 変数ルールを選択した状態で、もう一度ルール記録を行う。
- マッチングの確認とルール修正: 実行ページにおいて、マッチングを確認させる。意図せぬルールの発火が生じた場合には、その原因を探らせ、ルールの修正を行わせる。ルール修正の際には、講師が具体的な修正方法を受講生に教示するのではなく、2.1.3に示したものと同程度の情報を記載したwebページへのリンクをヒントとして与えるにとどめる。
- オリジナル問題の作成: 変数ルールの生成が完了した後に、所与の問題とは別のオリジナルの問題をイメージディスプレイに生成させる。既に作られた変数ルールでその問題が解決できるか試させる。解決できなかった場合、解決に必要な変数ルールを追加させる。オリジナルの問題を教室内で公開し、出来るだけ多くの問題を解決できるようにルールを追加させる。なお、受講生にオリジナルの滑車の問題を構成させ、教室内で強有させる授業デザインは、Miwa et al., (2012) において示されたものに従っている。多様な問題に対して適用可能なルールセットを構成することで、知識利用の柔軟性を理解できるようになると考えられる。

4. 授業における利用事例

4.1 授業の概要

著者らは、2008年度から2012年度までに計5回、滑車の問題を題材としたプロダクションシステムの授業実践を継続してきた。2節におけるシステム、および3節における授業デザインは継続的な授業実践の中で段階的に改善していったものである。それぞれの授業の受講者数、授業時間などの情報を表1に示した。また、各年度における授業の概要、および年度間での改善を以下に示す。

● 2008年度:

MIPPを用いず、どこプロのみを利用して滑車の問題を解決するモデルを生成させた。実習の前にプロダクションシステムに関する講義を実施した。実習において、受講生は、滑車の問題の例題、滑車の原理を自然言語で記述した用紙を受け取り、滑車の問題を自ら解決した。その後、どこプロ上で滑車の問題を解決するルールを記述することを求められた。3時間(2日間)の実習の中で、WMの状態記述、滑車の問題を解決する個々のルールを段階的に提示し、モデルの完成を導いた。授業内の実習後、自宅で自習することを教示し、後日テストを実施した。

● 2010年度:

プロトタイプのみ(MIPP(森田ら, 2010))を用いた授業実践を実施した。プロトタイプのみ(MIPP)は、どこプロと併置されず、ポップアップウィンドウでどこプロとは独立に提示された。また、問題生成ページにおけるイメージディスプレイの変化を赤字で提示する機能(2.1.1節を参照)は実装されていなかった。講義の流れは3節に示した授業デザインに従うものであった。講師は、受講生の進行をモニタしつつ、段階的に実習の進行を導いた。前年度と同様、滑車の問題の実習前にプロダクションシステムの講義を実施し、後日テストを実施した。

● 2011年度

本年度に使用したMIPPは、基本的に2節に説明されているものである。ただし、ルール記録機能に複数のバグが含まれており、ルール記録に関わる部分は授業外の自習で行わせた。授業の進行は、3節に示される授業デザインに従った。受講生には図9を直接提示し、各自の進行に任せて実習を行わせた。実習前のプロダクションシステムの講義は省略し、代わりに積み木の問題を例としてルールを記述させる自習用教材を行わせた。前年度までと同

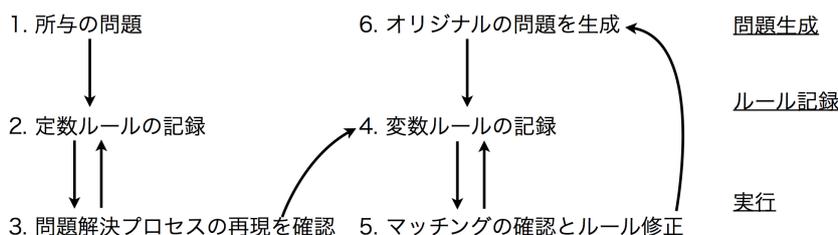


図9 授業デザイン .

表1 授業実践の概要 .

	2008年度	2010年度	2011年度	2012年度(1)	2012年度(2)
最終受講者数	8名	20名	5名	13名	27名
総授業時間	10.5時間	10.5時間	12時間	12時間	12時間
滑車の実習時間	3時間	3時間	4.5時間	3時間	4.5時間

様，実習後は各自で自習させ，後日テストを実施した．

- 2012年度(1)
2011年度に用いたMIPPのルール記録機能に含まれるバグを修正した．それ以外の教材，手続きは前年度と同様である．
- 2012年度(2)
これまでの授業実践とは異なり，この実践は2日間にわたる集中講義の1日分を割り当てて行った．実習時間は4.5時間であり，これまでの授業実践とは異なり，授業外で自習をする機会はなかった．実習の直後，テストが実施された．

4.2 実習の効果

4.2.1 パターンマッチングの理解

5つの授業では全て，実践後にパターンマッチングの理解を問うテストが実施された．テストでは，受講生に，WMの表現と変数を含むルールの対応づけを答えさせた．2008年度のテストでは，複数のルールと単一のWMを提示し，マッチするルールとそこでの変数束縛を答えさせた．以降のテストでは，複数のWMと単一のルールを提示し，マッチするWMとそこでの変数束縛を答えさせた．2008年度とそれ以降のテストではWMとルールを対応付ける方向が異なっているものの，この形式の違いは本質的ではないと考え，比較することとした．受講生の回答は，以下のように採点された．

- 3点: 発火するルール(もしくはWM)の名前が正確に記述され，全ての変数束縛を正確に記述．
- 2点: 発火するルール(もしくはWM)の名前が正確に記述され，一部の変数束縛を正確に記述．

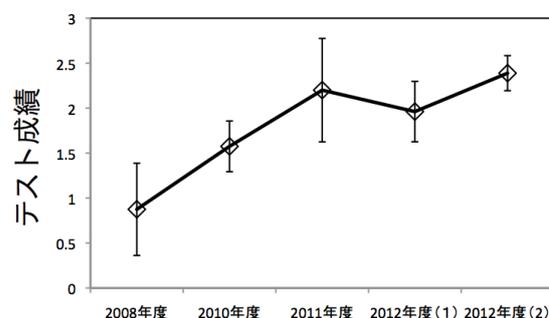


図10 テスト成績の推移 .

- 1点: 発火するルール(もしくはWM)の名前が正確に記述されたものの，変数束縛を正確に記述できなかった．
- 0点: 発火するルール(もしくはWM)の名前を正確に記述できなかった．

5回の講義でのテスト成績の推移を図10に示す．図10より，年度を経るごとにパターンマッチングの理解を問うテスト成績が向上していくことが分かる．被験者間1要因分散分析の結果，授業間のテスト成績に差が認められた ($F(4, 68) = 3.35, p < .05$)．HSD法による多重比較の結果，2008年度のテスト成績と2012年度(2)のテスト成績の差が有意となった ($p < .05$)．

5つの授業の間では，受講生の性質，講師の説明スキルの向上など，多様な差異が存在する．よって，授業間の差の原因を特定することは困難である．しかし，有意な差が初年度の講義と最終年度の講義で得られたことから，本研究で示した授業デザインが，より良い方向に改善されていっていることは確認できる．

4.2.2 人間の知識の柔軟性に対する理解

本研究の目的は人間の認知の柔軟性を理解することを支援するシステム、および授業デザインを提案することであった。実習後に実施したアンケートでは、1節において議論したプロダクションシステムのモデリングの効用を支持する回答が得られている。アンケートでは、「プロダクションシステムによる認知モデリングの利点」、「プロダクションシステムによる認知モデリングの欠点」、「プロダクションシステムと比較した人間の認知の特質」、「人間の認知と比較したプロダクションシステムの特質」、「認知モデルを作ることの意味」を自由記述させた、以下に回答の例を示す。

- (認知モデルを作ることの意味に対する回答) 逆説的な話であるが、人間の認知の仕組みの優秀さを改めて実感した。ノイズやエラーが発生しても何とか処理しようとする人間の認知システムは、どれほどコンピュータが進化してもシュミレートしきれないのではないかと。
- (プロダクションシステムと比較した人間の認知の特質に対する回答) ルールなどに決められていない、パターンにない状況に出会っても、自分で考え柔軟に対応できることだと思います。関数の処理速度には勝てなくても、新しいものに対して新しくルールを作っていくことが人間の認知の特質だと思います。また、物事を決められた順番に処理をしていくのではなく、時には順番を変えたり、同時に処理していると思います。

これらの回答は、プロダクションシステムによる認知モデルの潜在性について、厳密には正しくない理解も含まれる。しかし、プロダクションシステムのモデリングにより、人間の知識利用の柔軟性の理解が促進されるという本研究の前提とは整合するものである。さらに、人間の認知の柔軟性を認めつつ、プロダクションシステムによるパターンマッチングのポジティブな側面を指摘する回答もあった。

- (プロダクションシステムによる認知モデリングの利点に対する回答) 何より面白い点は、変数の束縛がステップごとに変化していき、流動的にシステムが進む点である。認知のモデリングとしては、C言語やJavaなどの一般的な言語より、プロダクションシステムのほうが、モデルをより包括的に捉えることができ、また創発的な振る舞いも発生しやすいのではないかと考える。
- (人間の認知と対比したプロダクションシステムの特質に対する回答) あらゆる条件をす

べて網羅することにより、人間では気づかなかった新たな視点やプロセスを発見することができる。

- (認知モデルを作ることの意味に対する回答) 人間の認知プロセスは人間自身には理解できない。しかし、認知モデルを作ることで、認知プロセスを具体的、厳密化することにより、人間はそのプロセスを視覚的に理解できる。また認知モデルを作る過程で自らの認知プロセスが、どのような構造なのかを知るきっかけにもなるのではないかと思う。

上記3つの回答のうちの1つめは、プロダクションシステムによるパターンマッチングが創発的な認知を支える機構としての可能性を認めるものである。この回答は、1節において述べたAnderson (2007) の見解と近いものと考えられる。2つめと3つめの回答は、プロダクションシステムの利用により、新たな視点の発見や人間の潜在的なプロセスが明瞭になることを指摘するものである。これらの効用は、森田・永井 (2011) において提案された認知モデルの利用法において提案されたものである。森田・永井 (2011) は、認知モデルを新たな視点を発見するためのリフレクション装置として利用することを提案した。

以上をまとめれば、授業実践によって、人間の認知の特質である知識利用の柔軟性に気づくという目的は達成されたと考えることができる。ただし、これらの回答が、直接的に実習の効果を示すとみなすことはできない。上記のような回答は、実習前後での授業内容、あるいは授業における講師の言葉などに影響されて記述されたた可能性は十分にあり得る。今後、アンケート内容の定量的、定性的分析を行うことで、提案する支援システム、および授業デザインの効果を検討していく必要がある。

5. まとめ

本研究では、人間の知識利用の柔軟性を理解させるためのツールとして、プロダクションシステムの学習支援システム、および授業デザインを提案した。提案したシステム、授業デザインは実際の授業において利用され、その効果が示された。

本研究において提案した学習支援システムMIPPの特徴は、人間による柔軟な知識の適用とプロダクションシステムによるパターンマッチングを対比させることを狙う点にある。変数ルールは、問題解決プロセスを機械的に分節化し、オブジェクト名を置換することで構成される。そのように機械的に切り出されたルールは、過剰適用され、学習者が思いもよらなかった状況に適用されることに

なる．そのような出力を目の当たりにし，学習者は人間の知識適用の柔軟さに気づく．また、ルールの過剰適用を抑制するプログラミングをすることで，モデリングスキルが向上する．

このような認知モデルの用いられ方は，哲学者によって「アンドロイド認識論」と呼ばれる人工知能の利用のされ方と類似する（戸田山, 2002）．アンドロイド認識論は，人間には実行できない合理的なアルゴリズムを実装することで，人間の認知を理解する方法論である．認知科学の学習者に，計算機による規範的なアルゴリズムと人間の認知メカニズムを対比させることで，人間の認知に対する発見が促される考えている．

参考文献

- [1] Anderson, J. R. (2007). How can the human mind occur in the physical universe? New York: Oxford University Press.
- [2] Larkin, J. H. and Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, 65-100.
- [3] 中池竜一・三輪和久・森田純哉・寺井仁 (2011) 認知科学の入門的授業に供する Web-based プロダクションシステムの開発. *人工知能学会論文誌*, 26, 536-546.
- [4] Miwa, K., Morita, J., Nakaike, R., & Terai, H. (in press). Learning through Intermediate Problems in Creating Cognitive Models. *Interactive Learning Environments*.
- [5] 森田純哉・三輪和久・中池竜一・寺井仁・齋藤ひとみ (2010) 問題の作成とモデルの構築を支援するユーザインタフェースの開発. 『人工知能学会第58回先進的学習科学と工学研究会資料』, SIG-ALST-A903-9, 45-50.
- [6] 森田純哉・永井由佳里 (2011). 類推の認知モデルの応用によるリフレクション支援システムの開発. *人工知能学会論文誌* 26 (5). 559-570.
- [7] 戸田山和久 (2002). 『知識の哲学』. 産業図書.