

中学校幾何論証問題の意味理解に関する研究(2)
repair theoryに基づく学習者モデル

古田貴久 松田稔樹 坂元昂
東京工業大学

あらし 高度個別化の実現手段の1つとしてITSの研究が進められてきた。このITSの重要な構成要素の1つに学習者モデルがある。適切な個別指導を行う上でこの学習者モデルは実際の人間の問題解決過程をとらえなくてはならないと考えられる。本稿では認知科学の知見を取入れ、repair theoryに基づく学習者モデルを形成した。幾何の作図問題におけるある誤答について、集団調査、プロトコル分析を行い、誤答発生過程をrepair theoryに基づいて詳細に調べた。この分析で得られた知見に基づき、TMSとabstractionを用いて学習者モデルを構築した。このモデルは、impasseおよびそれに対するrepairを再現し、かつまた、複数の問題に対して人間の学習者と同様な誤答を示したという点で、心理学的妥当性が高いと考えられる。

A study on understanding geometric problems at lower secondary schools level(2)
a student model based an repair theory

Takahisa FURUTA Toshiki MATSUDA Takashi SAKAMOTO
Tokyo Institute of Technology

Abstract Intelligent Tutoring Systems have been studied for 20 years. Student model is one of important component of ITS. We think it is necessary for a student model being psychologically appropriate. The purpose of this study is to develop a system which has a psychologically appropriate student model. We conducted psychological experiments and found that students realize they might be using an incorrect procedure while they attempt to repair impasses occurred in problem solving activities. This has not been considered by existing ITSs. We followed repair theory to develop a student model which can simulate impasses, an inevitability in human problem solving activity.

学習における高度個別化を実現する手段の1つとして、ITS(intelligent tutoring system)は、これまでおよそ20年にわたって研究・開発が進められ、さまざまな特色をもった数多くのシステムが構築されてきた。ITSは、初期においては、単に学習者が正しい解法手続きを適用できるようになるまでの訓練を行なうものであった。その後、学習者の意図を認識しようとする試み[10]がなされたり、誤答の発生原因を知識の不足ではなく、記憶していた手続きにおける誤りにも求めたり[13]、認知的プロセスに求める試み[13, 5]、あるいは、問題領域の原理に対する理解に求めたもの[9]のように多様化してきた。

しかしながら、これまでのITS研究の目的は、どちらかというと人工知能技術の応用であったと思われる。たとえば、池田ら[6]はShapiroのMIS(Model Inference System)とdeKleerのATMS(Assumption-based Truth Maintenance System)を組み合わせた学習者モデルを提案した。MISは帰納推論であるため、システムに与えられるオラクル¹の数が無限大のときにモデルが同定される。しかしながら、オラクルを与えるのは学習者であるため、学習の進行や不注意などのために、オラクルの一貫性は保証されない。そこで、ATMSを組み合わせることで一貫性を管理しようという極めて巧妙なシステムであるが、それが実際の人間の問題解決プロセスにおける何に対応するものであるかは述べられていない。他のシステムについても、現実の人間は、証明経路を始めから頭にもっているわけではないし、自分の持つすべての知識をいつでも利用できるわけでもない。

このように、これまでのITS開発においては、実際の学習者の内的なプロセスは、ほとんど考慮されることがなかったといえよう。実際の人間の問題解決における内的プロセスをITSに反映させるためにも、もっと認知科学の知見を参考にすべきである。人間の問題解決における内的プロセス、および、誤答発生メカニズムのモデルとして参考になるものとして、repair theory[1, 14]があげられる。

以下では、まずrepair theoryについて簡単に説明する。次に、高校生であっても中学校レベルの作図問題で誤ることを示し、問題領域の原理、基礎概念に対する理解を踏まえた上で、誤答の原因について考察する。これによって得られた知見をもとに、repair theoryに基づく学習者モデルを構築する。

本稿では、図形概念について十分に理解させることを目的として、作図を教授対象領域とした。

¹この場合は学習者の応答

$$\frac{-71}{172}$$

図1: 引き算の例

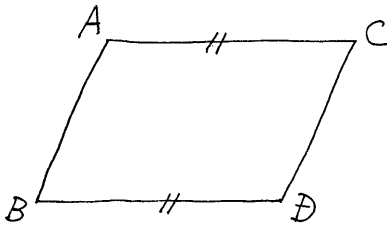
2 repair theory

人間の学習途上では、必要な知識を持っていないために、問題解決を続行することができない状態に陥ることがある。このような状態は行き詰まり(impasse)と呼ばれる。それにもかかわらず回答が要求されていると、その時点での行き詰まりをとりあえず解消し問題解決を続行するためにheuristicsを用いてもっともらしい回答を生成することで、その場をしのぐことが行なわれる。このような時に用いられるheuristicsを間に合わせ方略(repair heuristics)という。ただし、そのようにして生成された回答は、あくまでその場しのぎのもので、実際には誤っているものである。このようにrepair theoryでは、誤答原因を追求する範囲を、知識の不足やバグでだけで終らせるのではなく、impasseを解消しようという行動にも拡張するのである。

引き算を習いたでの生徒が図2に示す問題に次のように答えたとしよう。この時点でこの生徒は、8-2のように、1桁での、大きい数から小さい数の引き算はマスターしているが、左の桁から借りてくる手続きはまだであるとする。この生徒における誤答発生メカニズムをrepair theoryに基づいて説明すると、1例として次のようになる。

まず、1の位に注目して、3-1を行ない、2と書く。次に、10の位に注目するのだが、0-7を実行することができない。0から何かを引く時の手続きが十分獲得できていないからである。このような行き詰まりがimpasseと呼ばれるものである。答えが2であるところの生徒が確信してしまえば、それが回答になるのだが、まだ解き終わっていないと思っているならば、回答を継続しなくてはならない。そのため、なんらかの手を打って、事態を変える必要がある。そのために使われる、事態を変える方略をrepair heuristicsという。この例において採用されたrepair heuristicsは、swap-verticallyとしてよく知られる、上の数と下の数を入れ換えるというものであった。上下の数を入れ換えることで、7-0という、彼に解ける形になったため、あとは、またもとの引き算問題に戻って、回答を続行し、その結果、103-71=172という回答が生成されたのだという説明がなされる。

impasse解消のためにrepair heuristicsが使われる



平行四辺形 ABCD において $AC=BD$ とする。

図 2: 予備調査における誤答例

ということは、その人が *impasse* を認識したこと、および、*repair heuristics* は *impasse* になるまでの問題解決においては使われないことを意味する。これまでの学習者の捉え方では、学習者はあたかも計算機のように、正しい手続きと誤った手続きを区別なく適用し、行き詰まればそこで回答をやめると仮定されているようである。しかし、人間は問題解決の行き詰まりを認識し、そのため回答をやめよう場合があるし、また、あえて答を出そうとして *impasse* 解消するためにやむをえず *heuristics* を使う場合もあるのである。

一般に、人間のシステムエラーへの対処は、そのシステムに対する理解に依存すると考えられる。問題解決における *impasse* も、問題解決が中断されたという意味で、エラーの一種であると考えられる。したがって、*impasse* に対処するためにその人がとった行動は、その人の、問題および問題のドメインに対する理解が顕著に現れているであろう。

このように *repair theory* に基づく学習者モデルを形成することは、モデルの心理学的な妥当性を高め、かつ、学習者の理解にかんする情報を与えてくれるという点で極めて適切であると考えられる。

3 作図における誤答の分析

幾何の論証問題が苦手である中学生は少なくない。入試を経験した高校生であっても、図形の基本的性質に対する理解が不十分であることは少なくない。ここでは、1つの誤答に焦点を当て、その誤答が発生する時の学習者の内的なメカニズムを *repair theory* に準拠して考え、適用された *repair heuristics* について考察する。

3.1 誤答例

図 2 は本稿で扱う誤答の例であり、実際の実験から得られたものである。AC, BD は対角線であるか

ら、これは結局は長方形であるが、題意に沿って描いた場合、少なくとも、一般的な平行四辺形を描いて、その対角線が等長である記号をつけることが期待される。ところが、この被験者は、対角線をかかなかっただけでなく、頂点の名前のつけかたの約束を破ったことに注意してほしい。

回答直後のインタビューから次のことが確認された。

1. 被験者は、「平行四辺形 ABCD とあれば、頂点の名前は右回りか左回りに A, B, C, D としなくてはならない」ことは知っていた
2. それにもかかわらず、あえて図 2 のように作図した理由は、おおむね「問題文に $AC=BD$ と特に断って書いてあるから、特別な平行四辺形なんだと思った」からであり、「こうでもするしか、他に描きようがない」からであった。
3. この後被験者に「対角線を知っているか」と尋ねると、「ああ、なるほど」と言って、描き直した。

つまり、図 2 に示した頂点名の付け方をしたことは、対角線を思いつかなかった被験者の窮余の一策だったのである。

3.1.1 誤答と問題領域の理解

一般にこのような事後報告の信憑性は低いと言われるが、直後の内観報告ならばそれほど低いとは言えない。このやりとりから、次のような点が明らかになる。

- 被験者は、対角線を知っていたが、想起できなかった
- 自分の作図が既有知識と矛盾することに気づいていた
- 「 $AC=BD$ と書いてあるから」ということで、自分の作図の矛盾を正当化していた

問題文を読んだ時点で、被験者の頭の中には次のような情報が存在していたと考えられる。

1. 平行四辺形 ABCD を描く
2. 辺 AC, BD を描く
3. 頂点の名前の付け方には規則がある
4. 辺は図に明示的に存在しなくてはならない

1, 2 は問題文に書かれていた外的な情報であり、3, 4 は被験者の持つ知識である。そして、解は、1 から 4 までの制約を満たさなくてはならないという意味で、制約充足問題になっている。ところが、対角線

に気づかない限り競合が発生するので解が求まらない。つまり、impasse に陥ったのである。そこで、この被験者は制約 3 を無視することで、impasse 解消を図ったのである。なぜ 3 が選ばれたかという、それは作図という作業そのものに対するこの被験者の理解に基づくと考えられる。

作図とは、言語表現を図的表現に変換するという作業である。そして、この 2 つの表現が持つ情報は等価でなくてはならないことは、正しい作図の条件であり、いわば作図の原理の 1 つである。問題文に記述されたすべての図形要素が図に表されていることはその必要条件であり、制約 4 はこの原理を満たしている。さらに、それまでにこの被験者に提示されてきた図は完成図ばかりであった、すなわちこの条件が守られていた図であったということが、この制約を経験からサポートしてきたと考えられる。一方、3 はいわば教科書的な知識である。授業で教わったことだが、おそらく作図体験が乏しかったため、経験によるサポートが制約 4 と較べると弱かったのであろう。そのため、習ったことで知識として持っていたが、あえて無視したのだと考えられる。

頂点の名前の付け方の規則を無視して、辺を明示するという heuristics を適用したことは、被験者が図と問題文の情報の等価性という作図の根本条件を理解しているのだが、等価性の基準が幾何学とは違っていたことを示していると考えられる。このように、ここに示した誤答は、対象領域の原理に対する理解とそれにもとづく repair 行動、教科書的な知識と体験によってサポートされている知識との関係などを考える材料として興味深いものである。

また、これまでの repair theory に基づく誤答の分析においては、なぜその repair heuristics が選択されたかということについて考察されることはなかった。しかし、この例からは実は repair heuristics の選択においては、学習者の問題領域に対する理解のしかたが密接に関わっていることが示唆される。

次に、この誤答はこの被験者に固有なものではないことを、高校生を対象にした調査によって明らかにする。また、この誤答で適用されたと考えられる repair heuristics が上に述べたような生徒の理解に基づいているという仮説を、複数の問題に対しても同様な repair heuristics を適用することで検証した。

3.2 誤答の一貫性とその背景

高等学校の 1 年生 45 名を対象に、例えば四角形 ABCD において線分 AC の作図を要求するような、非明示的な図形要素を作図しなくてはならない問題を用いて集団調査を行なった。問題と手順を記したメモが数学科教師に郵送され、その教師が実施した。

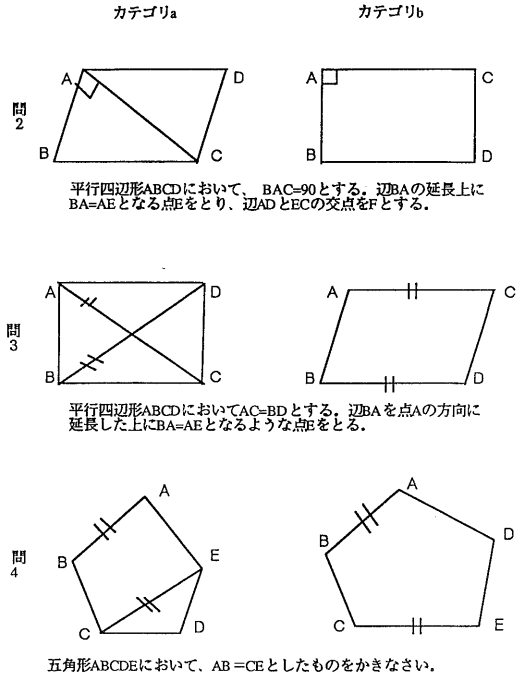


図 3: 各カテゴリの例

所要時間は 25 分であった。分析対象は、調査終了後返送された問題 1,2,3,4 に対する解答である。

3.2.1 回答の分類

多角形の頂点名の付け方に注目して回答を分類した。その結果を表 1 に示す。縦は問題番号で、横は回答の属するカテゴリである。図 3 に、それぞれのカテゴリの作図例を示す。カテゴリ a は正答、b は誤答のうち問題文中で明示された線分を多角形の辺としたもので、これらの作図は、さきに示した repair heuristics の適用に基づくという説明が可能である。回答カテゴリ c は、カテゴリ b に属さない誤答で、カテゴリ d は白紙、あるいはいくつも作図をしているが解として示した図はなかったという無答、図は描かれているが、頂点に名前がつけられていなかったもの、および、問題文をあきらかに誤読したと見なせるものである。表 1 の各セルは、そこに分類された被験者の割合である。

問 1 の誤答者率が $(64.4+11.1)=75.5\%$ と高くなっているが、これは被験者が問題文を途中まで読んだだけで作図したこと、および、国語力の問題という

表 1: 調査の結果

(それぞれのカテゴリに属する回答を行なった生徒の割合 [%])

問題	回答カテゴリー			
	a	b	c	d
1	20.0	64.4	11.1	4.4
2	57.8	17.8	0.0	24.4
3	42.2	28.9	24.4	4.4
4	75.6	8.9	0.0	15.6

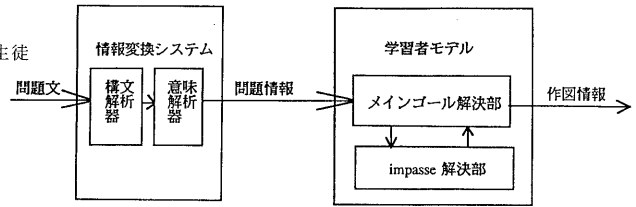


図 5: システムの全体構成

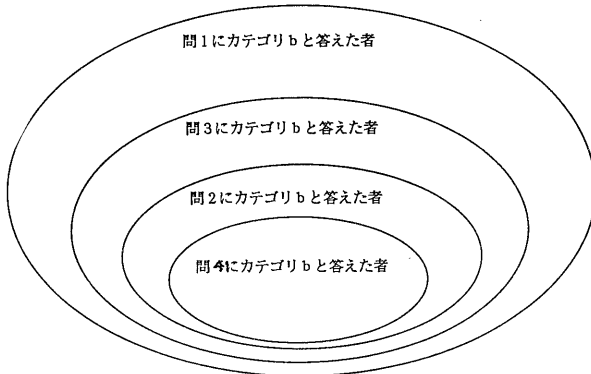


図 4: 誤答者の包含関係

要因が関係していることが考えられる。

3.2.2 誤答原因の考察

被験者側の要因として誤答分析を行った。問 4 において、カテゴリ b に属する回答を行った被験者は 8.9%であったが、問 2 に対してもカテゴリ b に属する回答を行った。同様に、問 2 に対してもカテゴリ b に属する回答を行った被験者は 17.8%であったが、問 3 に対してもカテゴリ b に属する回答を行った。さらに、問 3 に対してもカテゴリ b に属する回答を行った被験者は 28.9%であったが、問 1 に対してもカテゴリ b に属する回答を行った。これらに対して、誤答の内容から同一の repair heuristics を適用したためであるという説明が可能である。すなわち、複数の問題に対して、知識だけではなく行き詰まり解決の根拠となった原理に対する理解の点でも、一貫性が維持されていたといえる。

ところで、問 3 で誤った被験者のすべてが問 2 や問 4 では必ずしも誤っているわけではない。この点をより詳細にするため、別の高校において、複数の 2 年生から作図でのプロトコルデータを採取し分析した。その結果、被験者は、何を描くか決めてから作

図する方略と、特徴的な部分を先に描く方略とを使い分けているらしいことがわかった。前者の方略では、問題を読んだ後しばらく考えこんでから、図 2 のような作図を行なった。後者の方略をとると、例えば問 4 の「五角形 ABCDE において $AB=CE$ とする」の場合、線分 AB と CE を等長にとったあと、適当なところに点 D をとって、A,B,C,D,E と結んでいった。このように、図 2 を描いた被験者であっても、五角形 ABCDE を描く場合があることがわかった。ここで見た方略のような、幾何知識の有無以外の要因の研究は今後の課題である。

4 学習者モデルの構築

これまでの分析によって得られた知見をもとに、学習者モデルを構築した。学習者モデルは repair theory[1, 14] をもとにしている。問題解決行動全体のうち、impasse に陥るまでをメインゴール解決部、impasse の解決を impasse 解決部でシミュレートする。impasse の解決のためには、既有幾何知識と矛盾したルールを一時的に適用することになるので、学習者モデルは、それら 2 つに対応するサブシステムによって構成した。

学習者モデルの構成を図 5 に示す。学習者モデルへの入力、教材情報データベース作成用変換システム [7] の出力を用いた。教材情報データベース作成用変換システムの出力は問題文の表層的な情報のみをもっている。したがって、被験者の問題文の表層情報の抽出過程は、変換システムによってシミュレートしたとみなすことができる。学生モデルは教材情報変換システムが出力する初期フレームを入力とし、図に相当する情報を出力する。また、本システムにおける図形の表現系は、Greeno[3] を参考にした。これにより、この学習者モデルを、将来的には論証に適用することが可能であると考えられる。

```
rule( [sikaku(P 1,P2,P3,P4),
      hen(P1,P2), hen(P2,P3), hen(P3,P4), hen(P4,P1),
      parallel(hen(P1,P2),hen(P3,P4)),
      parallel(hen(P1,P4),hen(P2,P3))],
      heikousihenkei(P1,P2,P3,P4))
```

図 6: ルールの例

4.1 メインゴール解決部

メインゴール解決部は、TMS[2]を参考にしたプロダクションシステムで構成されている。ノードは、情報変換システムが問題文から抽出した意味情報を本システムの表現形式に変換した P-node(problem node) と、P-node から演繹された D-node(derived node) の 2 つに分けられる。また、とりあえず、学習者が impasse に陥る原因は、その学習者の知識に欠落があるためであるという前提をおくことにする。

どのルールを適用してもノードが変化しなくなった状態を、メインゴール解決部の一時停止条件とする。このとき、new である P-node が存在すれば、問題文中の情報を使い切っていないことが意味されるので、impasse 解決部を起動する。new である P-node がなければ、メインゴール解決は終了したものとす。

impasse の解決のために、P-node を無効にすることがある。無効にされたノードが発生すれば、そのノードから演繹されたすべてのノードは無効にされなくてはならない。このようなノード間の依存関係に基づいた処理を実現しやすいため、メインゴール解決部に TMS を採用したのである。

4.1.1 ノード

ノードには、そのノードが推論において利用されたかどうかを示す属性 (new / notnew) を付加した。

図 7 は図 2 の問題を、対角線知識を持つシステムに解かせた実行例である。initial nodes は、システムに与えられた P-node である。これらに、システムの持つ幾何知識を適用する。main-goal-solving node は、推論の結果である。new である P-node がいないため、impasse にならずに終了する。

対角線知識を持たないシステムに解かせた実行例は図 9 に示した。ノード 1 と 5 から 13 は平行四辺形 ABCD を作図するよう出力しているが、ノード 2 から 4 が使われていないため、impasse になる。これが図 2 に示した被験者が陥った impasse に相当する。

```
yes
| ?- solveMain.

initial nodes.
node(1, heikousihenkei(a, b, c, d), zentei, in, new).
node(2, hen(a, c), zentei, in, new).
node(3, hen(b, d), zentei, in, new).
node(4, prop(hen(a, c), hen(b, d), 1), zentei, in, new).
```

start applying rules...

```
main-goal-solving nodes.
node(1, heikousihenkei(a, b, c, d), zentei, in, notnew).
node(2, hen(a, c), zentei, in, new).
node(3, hen(b, d), zentei, in, new).
node(4, prop(hen(a, c), hen(b, d), 1), zentei, in, notnew).
node(5, sikaku(a, b, c, d), [konkyo, [1]], in, notnew).
node(6, hen(a, b), [konkyo, [5]], in, notnew).
node(7, hen(b, c), [konkyo, [5]], in, notnew).
node(8, hen(c, d), [konkyo, [5]], in, notnew).
node(9, hen(d, a), [konkyo, [5]], in, notnew).
node(10, prop(hen(a, b), hen(c, d), 1), [konkyo, [6, 8, 10]], in, notnew).
node(11, prop(hen(b, c), hen(d, a), 1), [konkyo, [7, 9, 11]], in, notnew).
node(12, parallel(hen(a, b), hen(c, d)), [konkyo, [1, 6, 8]], in, notnew).
node(13, parallel(hen(b, c), hen(d, a)), [konkyo, [1, 9, 7]], in, notnew).
```

```
yes
| ?-
```

図 7: メイン解決 : 対角線知識を持つ場合

4.2 impasse 解決部

impasse 解決部は、メインゴール解決部の推論結果を入力とし、それらに対して repair heuristics を適用し、一部の P-node に変更を加え、メインゴール解決部に制御を戻す。つまり、impasse 解決のために問題の一部を変更するのだが、この点は repair heuristics の与え方に依存している。すべての repair heuristics を適用しても状況が改善されないのは「どうしようもない」状態であり、回答が生成されない。

本研究では、impasse 解決部にこれまで人工知能の類推研究で用いられている abstraction[4] を採用した。abstraction について簡単に説明する。人間は、新しい領域にかんする利用可能な知識はもっていないものである。そのような場合、人間は、記憶から新しい領域と類似性のある過去の経験を検索し、その想起された類似例を参考にして、新しい領域で不足している情報を補うことを行なう。このような問題解決方法を一般に類推という。計算機上で類推を行う場合には、過去の経験を直接利用するよりも、過去の経験を抽象化した情報を持たせることで、効率的な推論が可能になる [1]。このような抽象化された過去の経験を abstraction という。

本システムにおいて repair heuristics を abstraction で表現した理由は次のようなものである。まず、人間が問題解決において類似例を利用することは知られることだが abstraction は類推研究で利用される技

```
abs( Draw(X,Y,Z,W),Fig(A,B,C,D),
    hen(X,Y),hen(Y,Z),hen(Z,W),hen(W,X))-
Draw=Fig,permutation([X,Y,Z,W],[A,B,C,D]),
cancel(Fig(A,B,C,D)).
```

図 8: repair heuristics の abstraction

法であるため、この点で親和性が高いと考えられる。さらに、repair heuristics は impasse 解決のための間に合わせ方略とはいえ、問題を解決するためのものなので、その被験者の対象領域に対する理解に反することはできないと考えられる。対象領域に対する理解とは、対象領域を支配する原理に対する暗黙の理解ということができるが、そのような理解は抽象的なものである。この点でも、ルールを明示的に表現しようというプロダクションシステム流の態度と比較して、abstraction の抽象性が、repair heuristics の表現に向いていると考えられる。

4.2.1 abstraction

図 2 に示した誤答を生成する heuristics を言葉で表せば「多角形において、問題文に明示された辺は、図にも明示されるよう作図する」であるが、これを abstraction で表現すれば、例えば四角形については図 8 のようになる。

abstraction は prolog の節形式で与える。head は抽象化された幾何知識を表し、body は prolog の述語によって変数間に制約条件を与えている。Fig,Draw は述語変数であり、平行四辺形、長方形といった四角形の具体的な名前とマッチする。また、A,B,C,D,X,Y,Z,W はアトム変数であり、それぞれ頂点名 1 つとマッチする。この abstraction はつぎのように解釈する。

```
Draw(X,Y,Z,W) という四角形は
hen(X,Y),hen(Y,Z), hen(Z,W),hen(W,X)
という 4 つの辺によって構成される。Draw
と Fig は四角形の種類としては同じものである。また、四角形 XYZW と四角形 ABCD
の頂点名は、この 2 つの四角形において共通して存在していればよい。四角形 ABCD
は取り除くことにする。
```

abstraction は、次のように使われる。問題が「 $AC=BD$ なる平行四辺形 ABCD」の作図を要求したとする。この問題は、次のように表現を変えて、図 8 に示す abstraction に与えられる。

- heikousihenkei(a,b,c,d)
- hen(a,c)

- hen(b,d)

heikousihenkei は arity(引数の数) が 4 なので、Draw(X,Y,Z,W) とでも Fig(A,B,C,D) とでもマッチしうるのだが、以下、Fig(A,B,C,D) にマッチした場合で説明を続ける。Fig=heikousihenkei, A=a, B=b, C=c, D=d に具体化されると、hen(a,c) は述語名が決まっており arity が 2 なので、abstraction の hen(X,Y) とマッチして、 $X=a, Y=c$ となる。つづいて、hen(b,d) だが、hen/2 で arity が b,d にマッチしうるのは、hen(Z,W) である。これで、 $Z=b, W=d$ となり、頂点名が決まった。次に、この abstraction を prolog のプログラムとして実行し、制約条件を満たすことを確認する。もし満たされなければ、バックトラックして再び具体化を試みる。なお、図 8 に示したように、hen/2 のような特定要素にマッチすればよい要素はあらかじめ具体化しておくことで、abstraction の探索効率が向上する。

4.3 シミュレーション

3.2 節において被験者に出题した作図問題を計算機に出题した。図 9 に、問 2 に対する問題文の問題情報ノードと、impasse 解決の結果を示す。また、他の問題に対しても、前節で見た被験者の誤答と一貫して一致する回答を行なった。これより、この学習者モデルは被験者の問題解決プロセスのモデルとして妥当性が高いといえる。

5 まとめ

本稿では、作図の誤答例について、プロトコルおよび手段調査から得られたデータをもとに、repair theory に準拠した誤り原因の説明と学習者モデルの形成を行なった。これにより、心理学的妥当性の高い学習者モデルが形成された。

今後は、repair heuristics を拡充する必要がある。repair heuristics を与えるには、ただ単にパフォーマンスを説明できる heuristics を決めていけば良いというわけではない。なぜそのようなパフォーマンスを示したのかを、本稿で行なったような、その被験者の問題や問題対象領域に対する理解に基づいて考察したり、あるいは、本稿では扱わなかったが過去の例からの類推について十分な考察が必要となろう。

参考文献

- [1] J. S. Brown and K. VanLehn: "Repair theory: A generative theory of bugs in procedural skills", Cognitive Science, 4, p.379-426(1980)

```

| ?- solveMain.

initial nodes.
node(1, heikousihenkei(a, b, c, d), zentei, in, new).
node(2, hen(b, a), zentei, in, new).
node(3, hen(a, c), zentei, in, new).
node(4, angle(hen(b, a), hen(a, c), 90), zentei, in, new).

main-goal-solving nodes.
node(1, heikousihenkei(a, b, c, d), zentei, in, notnew).
node(2, hen(b, a), zentei, in, new).
node(3, hen(a, c), zentei, in, new).
node(4, angle(hen(b, a), hen(a, c), 90), zentei, in, new).
node(5, sikaku(a, b, c, d), [konkyo, [1]], in, notnew).
node(6, hen(a, b), [konkyo, [5]], in, notnew).
node(7, hen(b, c), [konkyo, [5]], in, notnew).
node(8, hen(c, d), [konkyo, [5]], in, notnew).
node(9, hen(d, a), [konkyo, [5]], in, notnew).
node(10, proportion(hen(a, b), hen(c, d), 1), [konkyo, [1, 6, 8]], in, notnew).
node(11, proportion(hen(b, c), hen(d, a), 1), [konkyo, [1, 9, 7]], in, notnew).
node(12, parallel(hen(a, b), hen(c, d)), [konkyo, [1, 6, 8]], in, notnew).
node(13, parallel(hen(b, c), hen(d, a)), [konkyo, [1, 9, 7]], in, notnew).

display unused nodes...
node(2, hen(b, a), zentei, in, new).
node(3, hen(a, c), zentei, in, new).
node(4, angle(hen(b, a), hen(a, c), 90), zentei, in, new).

yes
| ?- solveImpasse(Alternative).

Alternative = heikousihenkei(b, a, c, d), heikousihenkei(a, b, c, d), hen(b, a), Y
hen(a, c), hen(c, d), hen(d, b) ?

yes
| ?- solveMain.

initial nodes.
node(1, heikousihenkei(a, b, c, d), zentei, out, notnew).
node(2, hen(b, a), zentei, in, new).
node(3, hen(a, c), zentei, in, new).
node(4, angle(hen(b, a), hen(a, c), 90), zentei, in, new).
node(5, sikaku(a, b, c, d), [konkyo, [1]], out, notnew).
node(6, hen(a, b), [konkyo, [5]], out, notnew).
node(7, hen(b, c), [konkyo, [5]], out, notnew).
node(8, hen(c, d), [konkyo, [5]], out, notnew).
node(9, hen(d, a), [konkyo, [5]], out, notnew).
node(10, proportion(hen(a, b), hen(c, d), 1), [konkyo, [1, 6, 8]], out, notnew).
node(11, proportion(hen(b, c), hen(d, a), 1), [konkyo, [1, 9, 7]], out, notnew).
node(12, parallel(hen(a, b), hen(c, d)), [konkyo, [1, 6, 8]], out, notnew).
node(13, parallel(hen(b, c), hen(d, a)), [konkyo, [1, 9, 7]], out, notnew).
node(14, heikousihenkei(b, a, c, d), zentei, in, notnew).

main-goal-solving nodes.
node(1, heikousihenkei(a, b, c, d), zentei, out, notnew).
node(2, hen(b, a), zentei, in, notnew).
node(3, hen(a, c), zentei, in, notnew).
node(4, angle(hen(b, a), hen(a, c), 90), zentei, in, notnew).
node(5, sikaku(a, b, c, d), [konkyo, [1]], out, notnew).
node(6, hen(a, b), [konkyo, [5]], out, notnew).
node(7, hen(b, c), [konkyo, [5]], out, notnew).
node(8, hen(c, d), [konkyo, [5]], in, notnew).
node(9, hen(d, a), [konkyo, [5]], out, notnew).
node(10, proportion(hen(a, b), hen(c, d), 1), [konkyo, [1, 6, 8]], out, notnew).
node(11, proportion(hen(b, c), hen(d, a), 1), [konkyo, [1, 9, 7]], out, notnew).
node(12, parallel(hen(a, b), hen(c, d)), [konkyo, [1, 6, 8]], out, notnew).
node(13, parallel(hen(b, c), hen(d, a)), [konkyo, [1, 9, 7]], out, notnew).
node(14, heikousihenkei(b, a, c, d), zentei, in, notnew).
node(15, sikaku(b, a, c, d), [konkyo, [14]], in, notnew).
node(16, hen(d, b), [konkyo, [15]], in, notnew).
node(17, proportion(hen(b, a), hen(c, d), 1), [konkyo, [14, 2, 8]], in, notnew).
node(18, proportion(hen(a, c), hen(d, b), 1), [konkyo, [14, 16, 3]], in, notnew).
node(19, parallel(hen(b, a), hen(c, d)), [konkyo, [14, 2, 8]], in, notnew).
node(20, parallel(hen(a, c), hen(d, b)), [konkyo, [14, 16, 3]], in, notnew).
node(21, chouthoukei(b, a, c, d), [konkyo, [14, 2, 3, 4]], in, notnew).
no impasse. finished.
yes

```

図 9: 問 2 のシミュレーション

- [2] J. Doyle: "A truth maintenance system", Artificial Intelligence, 12, 3, p.231-272(1979)
- [3] J. G. Greeno: "A study of problem solving". In R. Glaser (Ed.), Advances in Instructional Psychology vol.1, Lawrence Erlbaum Associates(1978) [山口 修平・東洋訳 "問題解決の過程～幾何の課題による研究" サイエンス社(1985)]
- [4] R. Greiner: "Abstraction-based analogical inference", In D. Helman (Ed), "Analogical Reasoning", Kluwer Academic Publishers(1988)
- [5] 平島 宗, 河野 隆宏, 柏原 昭博, 溝口 理一郎, 豊田 順一: "教育タスクの分析とそれに基づく ITS の設計", 人工知能学会全国大会予稿集, p.711-714(1990)
- [6] 池田 満, 溝口 理一郎, 角所 収: "学生モデル記述言語 SMDL と学生モデルの帰納推論アルゴリズム SMIS", 電子情報通信学会論文誌, J72-D-11, 1, p.112-120(1989)
- [7] 松田 稔樹, 古田 貴久, 坂元 昂: "中学校幾何論証問題の意味理解に関する研究 (1)", 日本教育工学会研究報告集, JET90, 5, p.57-64(1990)
- [8] 文 部 省: "中学校学習指導要領(平成元年3月)", 大蔵省印刷局(1990)
- [9] 中村 祐一, 平島 宗, 上原 邦昭, 豊田 順一: "ITS のための説明機能に関する検討-ドメイン原理および類推を用いた学生モデル生成法-", 電子情報通信学会論文誌, J73-D-II, 3, p.399-407(1990)
- [10] 岡本 敏雄: "初等幾何 CAI システム-GEOMEX-", 情報処理学会論文誌, 29, 11, p.1301(1988)
- [11] 桜井 成一郎, 原口 誠: "アブストラクションによる論理プログラムの構成的帰納学習", 富士通第 7 回国際研シンポジウム予稿集(1990)
- [12] B. J. Reiser, J. R. Anderson and R. G. Farrell: "Dynamic student modelling in an intelligent tutor for lisp programming", Proc. of IJCAI, p.8-14(1985)
- [13] D. Sleeman and J. S. Brown: "Intelligent Tutoring System", Academic Press(1982)
- [14] K. VanLehn: "Mind Bugs", MIT Press(1990)