

Arithmeticus : A DPS – Based Model for Arithmetical Competence

Jl. Of Interactive Research (2000) 11 (3/4), 465-484

JOSEPH KLEP

1. The Position of Arithmeticus Between other Student Models

- Petrushin & Sinitsa(1990)、学習者モデル、熟練者ルールのセット
- Arithmeticus、メタ数学モデル、計算と議論のライン
- 核(kernel)、一般的な(generic)で認識論的記述
- 生徒の回答の注釈、MathMirror、生徒の表現と Arithmeticus による回答の比較
効率、速度、自動化の程度、使用された知識のルートなどで同定(qualify)
- 生徒がやってることを理解する友達、後退や新しいものの生産をマーク
- アルゴリズムや推理(argumentation)の表現と構築における2つのレベルのエラー
受け入れない転移(transition)：論理的、統語論的なエラー
汚い推理：形式的には正しいが不適切な推理
- 論理的、統語論的なエラー、ユーザインタフェースで探知
汚い推理、生徒の遂行後 Arithmeticus により探知
- Arithmeticus にはエラーは重要ではない
生徒が獲得した知識をいかに合成するのか
- 心理学的 shell、知識やスキル
Arithmeticus、生徒の回答の質に関するコミュニケーション
認識論的側面、数学学習に関するアイデア、DPS-based microworlds

2. Education

- 言語に表現されるカオス的な思考の評価
表現の受容有無に関する文化的決定
概念と主張のラインは成功的な表象である必要はない
要点を記述することにより概念を定義する scholastic 方法、
知識を表象する方法ではあるが学習の興味ではないかも知らない
- 熟練者システムは scholastic 知識表象モデルに基づく
科学的結果と方法を記述、説明、批評するには適切だが
創造的思考を促進するには不適切
- すべての生徒が熟練の数学者のように振舞う必要はないが、
数学者との対話で感覚的な、思考で創造的であることはできる
- 教示は創造的な思考の促進、
コミュニケーションにおける reflection に焦点をあわせるべき、

ABOUT CONCEPTS

- ・ 社会的構築主義者(socio-constructivist)の観点(Varela,1990)
概念は経験のかたまり(cluster)
伝統的な論理における「関係 is」は哲学や構築主義心理学には大事

例文

- ・ しかし、思考に関して考える際はそれほど有用ではないかも知らない
本に関するより有効な言語哲学的な質問の例
- ・ imagination の流れとしての思考の強調(Varela)
- ・ 言われたこと行われたことを理解する別の方法

MATHEMATICAL METHOD

- ・ ISMA プロジェクト、思考(thinking)と推論(reasoning)の差
思考は創造的で、ルーズで、構造化されてないプロセス
推論は思考の流れをラインとして表現すること
- ・ コミュニティにおける数学の学習、他者の反応における reflecting

Figure 1

THINKING AND REASONING

- ・ 推論は思考の language-bound part で、方法論 based であるが
思考は imagination の流れである
- ・ 数学コミュニケーションにおける 2 つの表現の反応(reaction)
形式的な反応：論理的な正誤の判断
意味論的な反応：相手が表現したことに関する理解や同意

1. Dynamic Problem Spaces (DPS) and Mathematical MicroWorlds (MMW)

- ・ 問題と連合された知識は力動的な知識である
Actual knowledge は dynamic problem space(DPS)として呼ばれる

- ・ Klep(1992)の mathematical microworlds(MMW)
 - 生徒の DPS の活性化を助ける
 - 生徒は推論と計算を表現できる
 - 生徒の活動に関して feedback を提供
 - ・ MathMirror は生徒の回答と活動の reflection を支援するより MMW である
 - Arithmeticus は MMW において適切な情報、提案、道具を表現する生徒 Math Model(PMM)
 - ・ 生徒が poor DPS を持つと問題はそれほど meaningful でないが、rich DPS を持つとより meaningful になる
2. Experts, Expert-Knowledge and Expert Systems.
- ・ 熟練者は rich DPS を持って、様式化された方法論の使用に訓練されている
 - ・ エキスパートシステムにおいて、知識は事実とルールでモデル化
 - ただ、知識や思考プロセスそのものではなく、主張のラインと方法として表象する
3. Competence versus Performance in Reasoning
- ・ 思考と推論の差、理解と説明の差
 - 訓練された方法で数学を解ける生徒が poor DPS を持っているかも知らない

EDUCATION

- ・ 生徒の DPS の要素を反映する豊富な MMW
 - 教育は good expert が何をするかを提供することではなく、DPS を豊富にするための助けである

MAN-COMPUTER DIALOGUES IN THE ISMA PROJECT

- ・ 自然な person-to-person コミュニケーション(Figure1)
 - Man-computer interaction、 ISMA プロジェクト(Figure2)
 - Arithmeticus は argument のラインとして生徒の表現を解釈する

Figure2

- ・ ISMA プロジェクトで生徒の作業は MathMirror と呼ばれる
 - 生徒の推論遂行、DPS を反映し、回答の質は初期作業と比較される
 - starting では知識ベースは空で、
 - ルールと関係要素がベースであるアルゴリズムの生成のみ含む
 - 基礎知識は 0 と 1 の加算と引算である

1. Students Errors

- ・ ISMA プロジェクトでは手続き的エラーに焦点がない

エラーが起きるとステップか回答を recreate すべき

- 回答の質は生徒の学習暦と比較される
- 形式的エラーのない回答は Arithmeticus には正しい
回答が長すぎるか複雑な場合はコメントを与える

2. MathMirror

- MathMirror1 は説明的な MMW
 - MathMirror は最も適切な presentation を提供
 - MathMirror は準備の front-end で、
Arithmeticus は初期数学の学習のための生徒モデルである
- 初期数学における加算と引算などを減らすための方略を構築し訓練するためのプログラムを提供

Figure3

- Worksheet には $85+38$ の式が書かれて、生徒が答えを分からなかった場合、
次のステップが書かれる
- $$\begin{aligned} &= 85+30+8 \\ &= 115+8 \\ &= 123 \end{aligned}$$
- 生徒が何をすべきか分からない場合は小さいアイコンが使える
 $38 = 30+8$ あるいは $40-2$ のうち 1 つの選択が可能
 - 生徒は $38-15$ が 23 であることを記憶し途中のステップを減少するかも
 - 最後のステップ $100+23=123$ は簡単である
 - nice numbers、対案になるリストの提供

例 挿入

- MathMirror は生徒に有用な知識を提案する
しかし、 $85+38=100+23$ の提案を提供する前に、このステップが有用であるかどうかを
評価すべき
- Upper toolbar の説明は省略
- 代用(substitution)の数学的アイデアは marking と replacing により支援される
- lower bar の説明

図 挿入

- arrow-language worksheet

図挿入

- numberline worksheet
方略を表現するために数字を書く多様な技術が支援される
85+40 により 85+38 の推定が可能

図

- 間違った replacement のケース
MathMirror の反応

図

- MathMirror は生徒の DPS の counterpart を提供する
MMW(worksheet)は生徒の回答に従い、生徒の DPS に属するアイデアを提供する

ARITHMETICUS : A STUDENTS MODEL

1. Arithmeticus : Modelling Student's DPS's

- 生徒が回答を生成するいくつかの方法
換算 (reduction) は暗記(by heart)により知っている
よく知られたすべてのアルゴリズムが記憶できる
表現が推定できる
表現は複数の数字に分解(split up)できる
交換 (commutation) や代用(substitution)のような対数的なルールが使用される
- 85+38 を解くことは連合の chain を生成こと
生徒の換算は一連の数学的変換(transformation)としてモデル化される
- $45 + 19 = 60 + 4 = 64$ の後ろにある方略

Table1

- 解釈 A は最も一般的 : 19 の分解
解釈 B は 19 を 20 に変換するために $a+b=(a-c)+(b+c)$ のルールに基づいている
解釈 C は、19 は単純に 20 に近いというアイデアに基づいている

解釈Dは、いつも $a+b=(a-c)+(b+c)$ のルールに基づいている

・ $45 + 10 = 55$ と $45 + 20 = 65$ のような reduction を知ってる、10 のステップでカウントしている生徒は、解釈Bの5 - 8のステップ, Cの7 - 10, Dの $60 + 4$ は余分なものである

式挿入

・ よい教師は $45 + 19 = 60 + 4 = 64$ という生徒の回答から、指導と関連するアルゴリズム、生徒に与える回答のステップなどを考える

・ 生徒の変換を理解するために、ありうる変換を収集、representation 問題と collecting 問題を持つ

しかし、これらの変換と実際の生徒の作業とマッチする問題がのこる

図

・ 生徒の学習の変化に応じた事実とアルゴリズムの力動的なセットが必要であることを示唆

ARITHMETICUS : GENERATION OF ALGORITHMS A STUDENT IS ABLE TO

- ・ Arithmeticus は変換(transformation)を結合することにより変換(reduction)を生産する
- ・ Arithmeticus は推論エンジンで、static ルールと dynamic ルールがある
 - static ルールには algebraic ルールと表象(representation)を変えるルールがある
 - dynamic ルールは現在の生徒に有効な事実(fact)とアルゴリズムである
- ・ 事実は記憶された関係である($3+4=7$ のように)
 - アルゴリズムはステップの数における表現の変換である
- ・ Arithmeticus における概念は「neighbourhood of」のように定義される
- ・ 力動的 transformation は現在の生徒が以前学んだ事実とアルゴリズムである
 - Arithmeticus は現在の生徒に新しいアルゴリズムを決め、生徒の知識ベースに追加する
 - このように力動的ルールと DPS モデルは成長する
- ・ Arithmeticus はアルゴリズムの一般的なモデルを提供、そのモデルのなかで生徒の新しい変換が解釈される
- ・ 生徒の変換は次のステップで解釈される
 - どの形式の変換が現在の生徒に可能であるのかを聞く
 - 現在の生徒の変換と Arithmeticus の回答をマッチする
- ・ マッチングプロセスにおける曖昧さがある場合は、より詳細に自分のアルゴリズムを説明するところから会話が始まる

ARITHMETICUS : SUPPORTING REFLECTION

- ・ アルゴリズムを比較する可能性

長さ、複雑性、整数の数、練習のレベルなどの特性を基礎にして、アルゴリズムの質が比較できる

現在のアルゴリズムと以前使ったものを同じ変換タイプで比較

生徒が変換における反映(reflect)を作るように Arithmeticus は以下のようなコメントをする

会話挿入

ARITHMETICUS :GENERIC PLANNING OF EXERCISES

- ・ Arithmeticus で最も興味のある側面は次の発達領域を計算することにより、練習(exercise)を計画する可能性である

目標練習タイプのセットと Arithmeticus の現在の状態が与えられると、現在の生徒のレベルにおいてどのタイプの練習ができるか計算できる

- ・ panning の方法は通常のカリキュラム定義と異なって、一連の練習がなく目標練習タイプのセットのみある
- ・ 2つの目標練習タイプは形式タイプとアルゴリズムタイプ

IMPLICATIONS OF ARITHMETICUS : GENERIC STUDENT – VERSUS EXPERT-MODELLING

- ・ Arithmeticus と MathMirror は生徒が個人方略を発達できる学習環境を提供する

そこには事前に定義された熟練者の回答がない

回答は生徒の学習暦と関連する

Arithmeticus と MathMirror は個人的思考を促進する会話に基づいた意味を提供することにより生徒の創造的思考を育てることができる