第五世代コンピュータから スキルサイエンスへ

- 論理プログラミング・アプローチ -

慶應義塾大学　名誉教授
嘉悦大学　教授
人工知能学会フェロー
古川康一

目次

- 第五世代コンピュータシステムプロジェクト
  - スライド3～29
- スキルサイエンスへの挑戦
  - スライド30～66

第五世代コンピュータシステムプロジェクト

1982~1992

第五世代コンピュータプロジェクトとは?

- 1982年~1992年の11年間に、通産省（現経産省）の後押しで進められた国家プロジェクト
- 通産省は、IBMに追いつき追い越すことを目標とした。
- そのためには、画期的な技術の開発が望まれた。
- 並列推論マシンの構築を目指した。

- プロジェクトリーダ：渕一博（ICOT所長）
- その他の中心的メンバー：古川康一（ICOT次長）、横井俊夫、村上国男、内田俊一、上田和紀、近山隆、松本裕治、長谷川隆三、竹内彰一、国藤進、新田克己、井上克巳、瀧和男、橋田浩一、向井国昭、…

第五世代コンピュータ・プロジェクト前夜

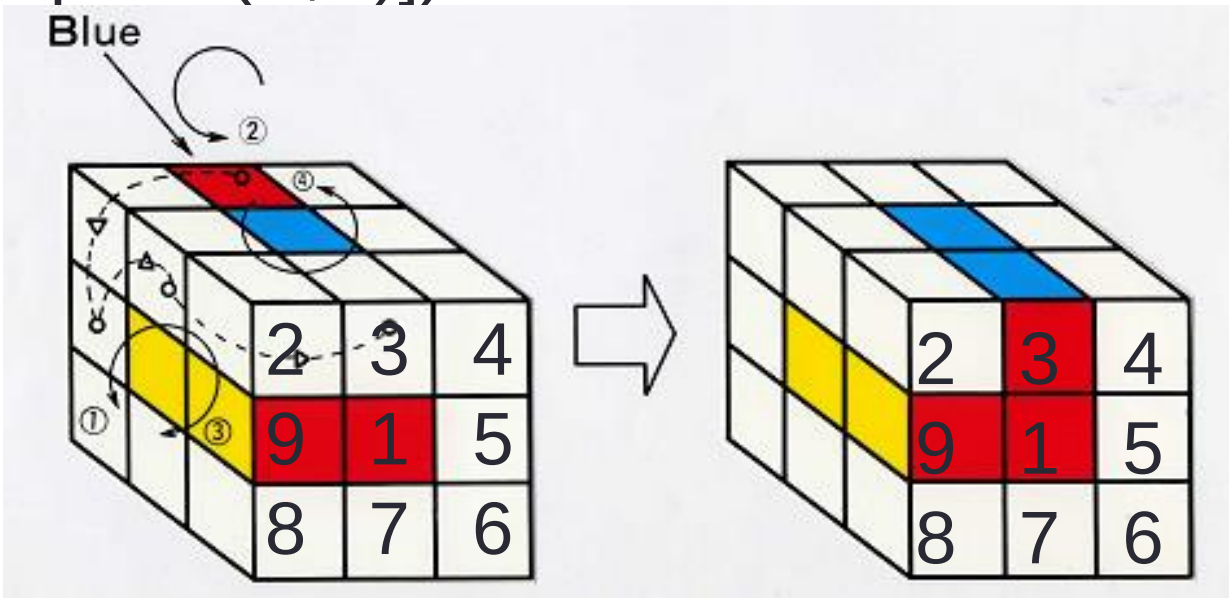
- アメリカ留学中に、スタンフォード・リサーチ・インスティテュート（SRI)で、コルメラワの書いたPrologインタプリタのリスティングを発見し、ETLに持ち帰る(1976年）。
- 渕さんが、そのリスティングを解読し、走らせる。
- Prolog熱の始まり。
- Prologで、プロダクションシステムを作り、ルービックキューブプログラムを動かした。

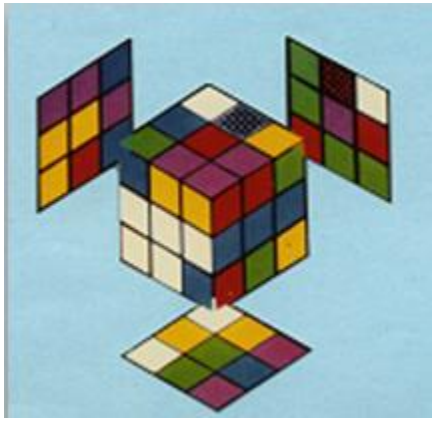
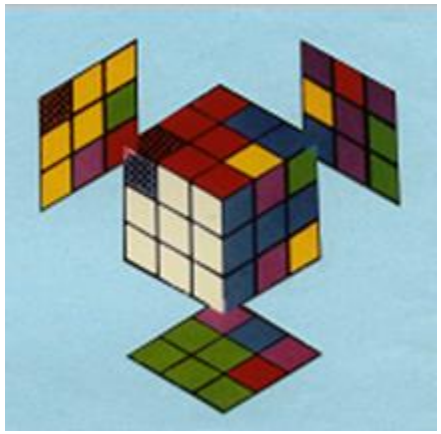
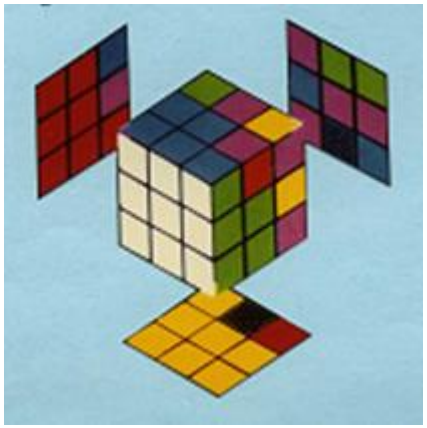
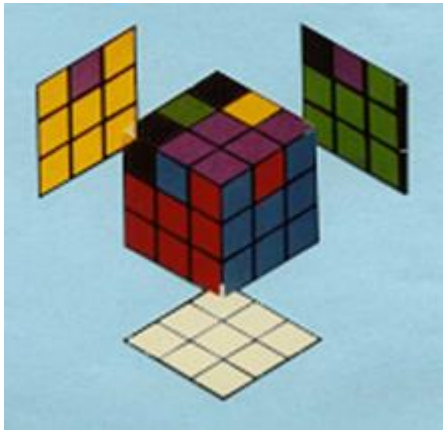
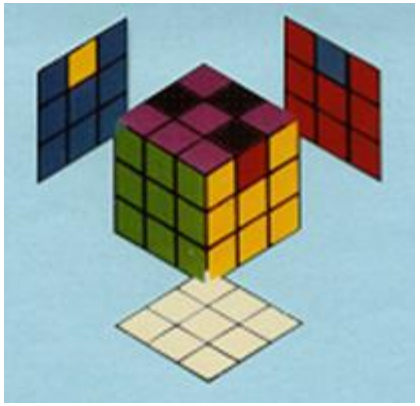
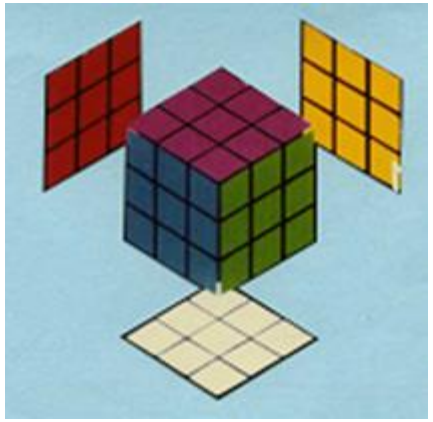
Prologで書いたプロダクションシステム

```
prodSystem(WM, FinalState) :-
        member(FinalState,WM).          %終了条件
prodSystem(WM, FinalState) :-
        member(Fact,WM),
        recognize(Fact,WM,RHS),         %recognize-act
        act(RHS,WM,NewWM),              %cycle
        prodSystem(NewWM, FinalState).
recognize(Fact,WM,RHS) :-
        rule(LHS=>RHS),                 %ruleのrecognize
        member(Fact,LHS),               %条件
        deduce(LHS,WM).
```

ルービック･キューブの解法ルール

```
rule([cube(front(FC,_,_,_,_,_,_,_,_),
           back(_,_,TC,_,_,_,_,_,_),
           top(TC,_,FC._,_,_,_,_,_),_) =:X]
     =>
     [call(apply([lup,bccw,ldown,tright],X,Y)),
      replace(X,Y)]).
```



FGCS1981の開催

- 1981年10月19~22日、京王プラザホテルでFifth Generation Computer Systemに関する第1回国際会議が開催された。
- Key Note:元岡 達
- 分科会報告：唐津一、渕一博、相磯秀夫
- Problem Solving and Inference Mechanismsに関する計画の発表：古川
- 渕一ファイゲンバウム論争

渕-ファイゲンバウム論争

- ファイゲンバウム:「アメリカの専門家たちは、人工知能の研究を25年間やってきています。知識工学に限っても、研究が始まってからすでに16年です。しかし日本はまだ、せいぜい数年でしょう。日本は、よそに追いつくために、大きな努力をしなければならない、と思います」
  （何でLISPをベースにしないのか?）
- 渕:「皆さんは日本を赤ん坊のようにおっしゃるが、私は少年ぐらいにはなっていると思います。少年は、大人の意見をよく聞き、学ばなければならないのは確かです。しかし、自分で判断することも大切ではないか。それによって少年は自ら成長するのです」
- 渕:「われわれは若いだけに、なんでも取り入れる柔軟さを持っています。だからこそ、オープンにこの国際会議も開いたのです。あなた方は確かに大人でしょう。でも、経験がありすぎて、ものが正しく見えなくなっているのではありませんか」
  （将来を見据えてLogic Programmingを選択する、という判断を下した）

（今岡和彦:我が志の第五世代コンピュータ、 TBSブリタニカ、1989）

Alan Turingの構想

- Alan Turingは、ComputerによるAIの実現可能性を論じている(1950)。
- 3つの案

(1) AI by programming,

(2) AI by *ab initio* machine learning,

(3) AI using logic, probabilities, learning and background knowledge.

- 第3のアプローチが最も有望であると主張した。

Turing, A.M. (1950). Computing machinery and intelligence. Mind, 59, 433-460.

S. Muggleton (2014). Alan Turing and the development of Artificial Intelligence. *AI communications*, Vol.17, No.1, 3-10.

プロジェクトの中心理念

- ハードウエア：非フォンノイマンコンピュータ
- ソフトウエア：知識情報処理
- プログラミング：並行論理プログラミング

古川康一，第五世代コンピュータのプロジェクト運営と人工知能の未来，人工知能，Vol.29, No.2, 2014

並行論理プログラミング

- Prologの問題：並行処理が記述できない。そのため、OSが書けない。
- 並行論理プログラミングの出現
  - Keith Clarkによる「論理プログラミングの枠組みでの並行プログラミングの提案」に着目(1981)

Keith Clark論文(1981)

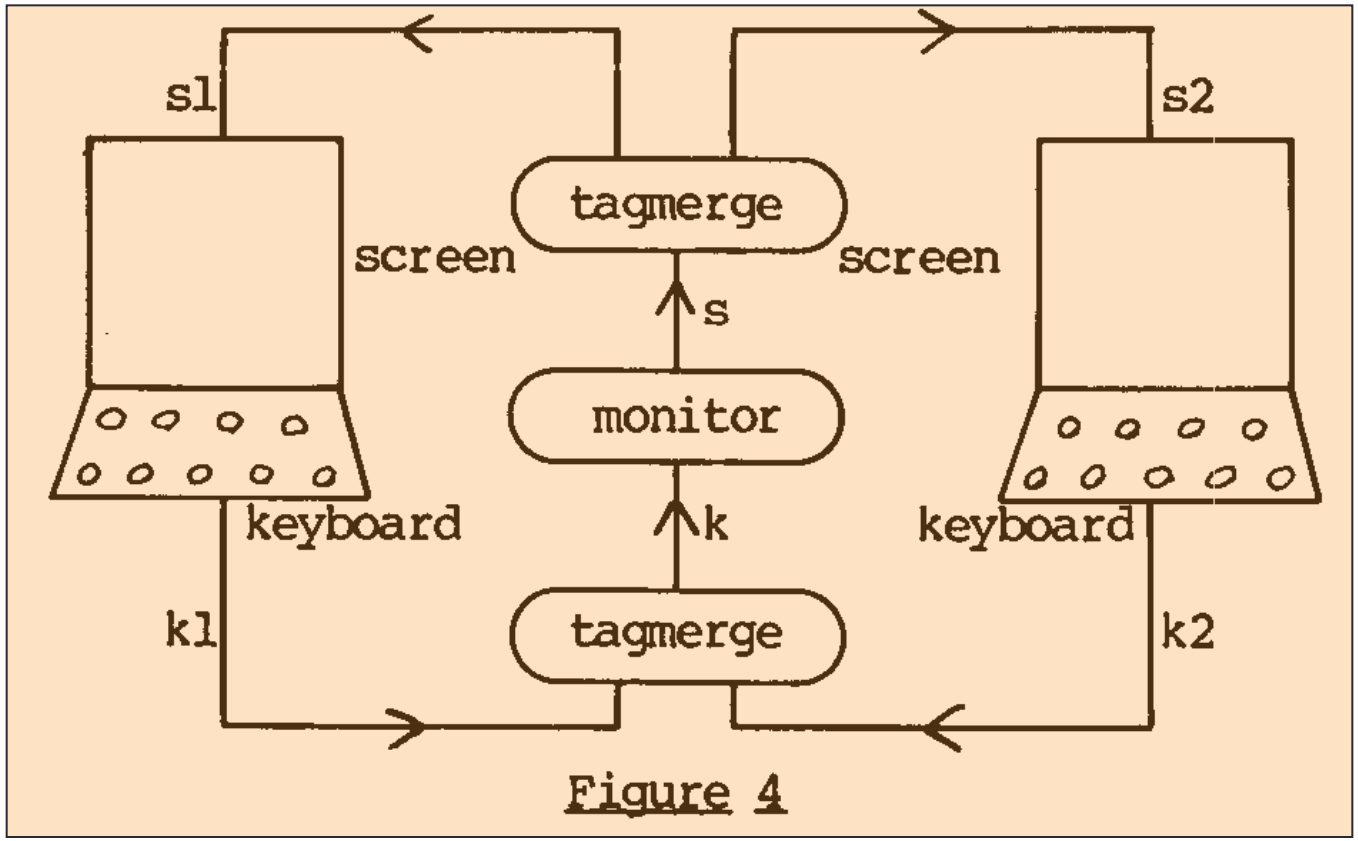


Figure 4

Keith L. Clark and Steve Gregory. 1981.
A relational language for parallel programming. In Proceedings of the 1981 conference on functional programming languages and computer architecture, 171–178. ACM Press.

Prologと並行論理プログラミング

- Prologプログラム

  ?- keyboard(K1), keyboard(K2), append(K1,K2, K),
  monitor(K,S), append(S1,S2,S), screen(S1), screen(S2).

  - 各リテラルが順次実行される。

- 並行処理を含んだ論理プログラム

  ?- keyboard(K1), keyboard(K2), tagmerge(K1,K2, K),
  monitor(K,S), tagmerge(S1,S2,S), screen(S1), screen(S2).

  - 各リテラルが並列に実行される。

核言語の設計

- 候補
  - Keith Clark, Steve Gregory : PARLOG
  - Ehud Shapiro : Concurrent Prolog
- 上田和紀は、新たな言語**Guarded Horn Clauses(GHC)**を提案(1984　クリスマス)
- GHCの利点:
  - ガード部のデータフロー実行規則による並列制御
  - アノーテーション、モード宣言・コンパイルが不要
  - 最もシンプル!

GHC開発の意義

- 高いオリジナリティ
- ハードウエア（並列推論マシン）の開発と、ソフトウエアの開発を同時に進めることが可能になった。
  - 並列推論マシンPIM（瀧和男）
  - 並列オペレーティングシステムPIMOS（近山隆）

- 問題点
  - バックトラッキングができない。
  - 探索機能の喪失

Manthey, Bryによるボトムアップ定理証明器*SATCHMO*

- ***SATCHMO***　:たった8つの節から成るフルセットの一階述語論理のための定理証明プログラム

```
satisfiable :- is_violated(C), !, satisfy(C),  satisfiable.
satisfiable.
is_violated(C) :- (A--->C), call(A), not(C).
satisfy(C) :- component(X,C), asserta(X),
              on_backtracking(retract(X)),
              not(false).
component(X,(Y;Z)) :- !, (X=Y; component(X,Z)).
component(X,X).
on_backtracking(X).
on_backtracking(X) :- call(X), !, fail.
```

Manthey, R. and Bry, F. (1988): *SATCHMO: a theorem prover implemented in Prolog*. Proceedings of CADE 88 (9th Conference on Automated Deduction),SpringerVerlag.

GHCによる定理証明器の推進

- 研究目標：SATCHMOのGHCによる実装
  - 古川による論文の発見、プロジェクトの発足
  - 渕コーディング
  - 長谷川、藤田によるインタープリタの開発
    - MGTP(Model Generation Theorem Prover)へと発展
    - 並列推論マシン上で、高速実行を達成（256プロセッサで220倍）
- **MGTPの実現は、並列推論の可能性を示した。**

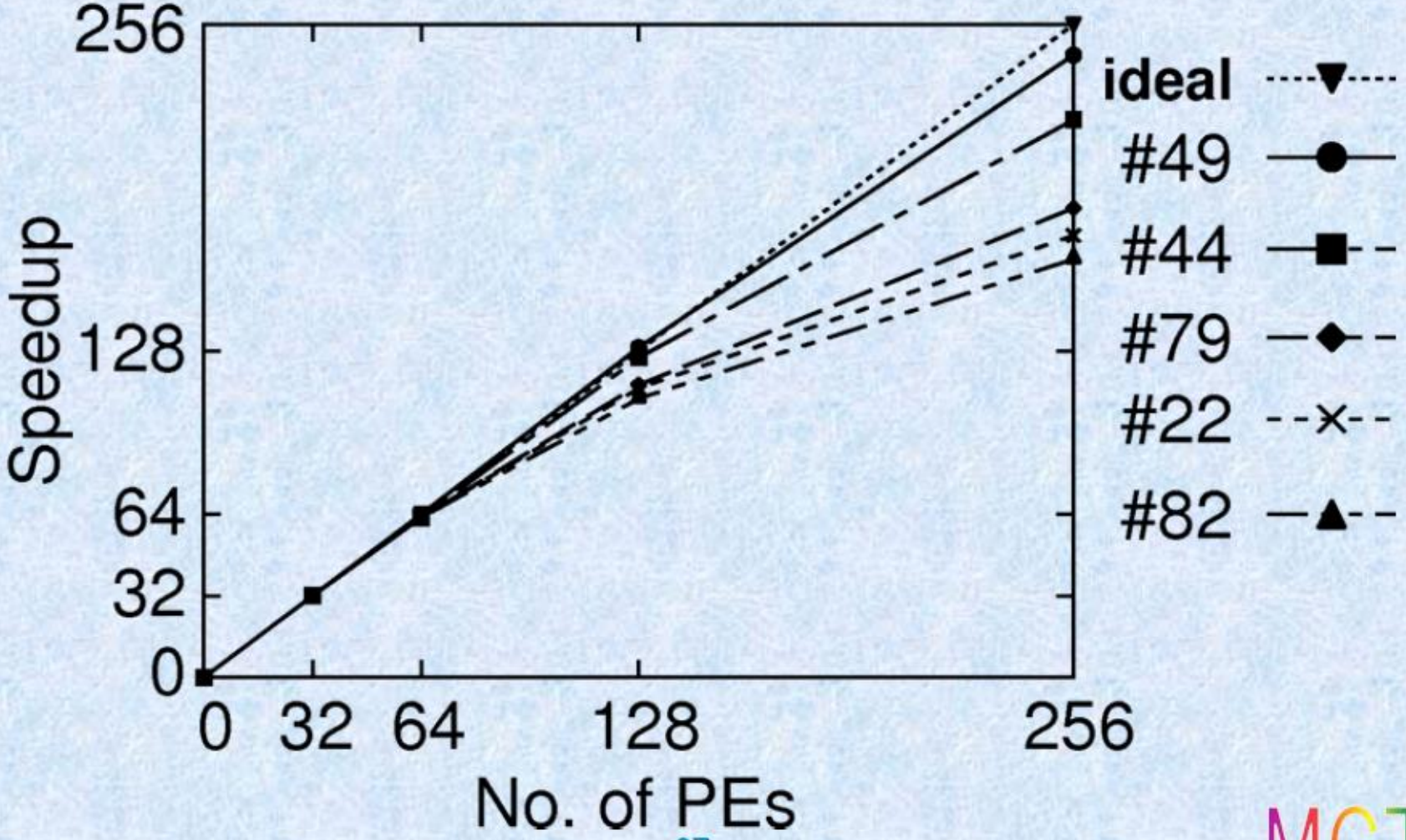
SATCHMO 実装のトリック

- GHCの問題点
  1. GHCは、OR並列が出来ない。
  2. ユニフィケーションにも制限がある。呼び出されたプロセスしか変数に値を代入することができない。
- 問題回避の方法
  1. OR並列機能は、AND並列機能によって代用した。
  2. ユニフィケーションの制限は、***SATCHMO***における値域限定条件の性質により回避できた。

SATCHMOのGHCインタープリタ（一部）

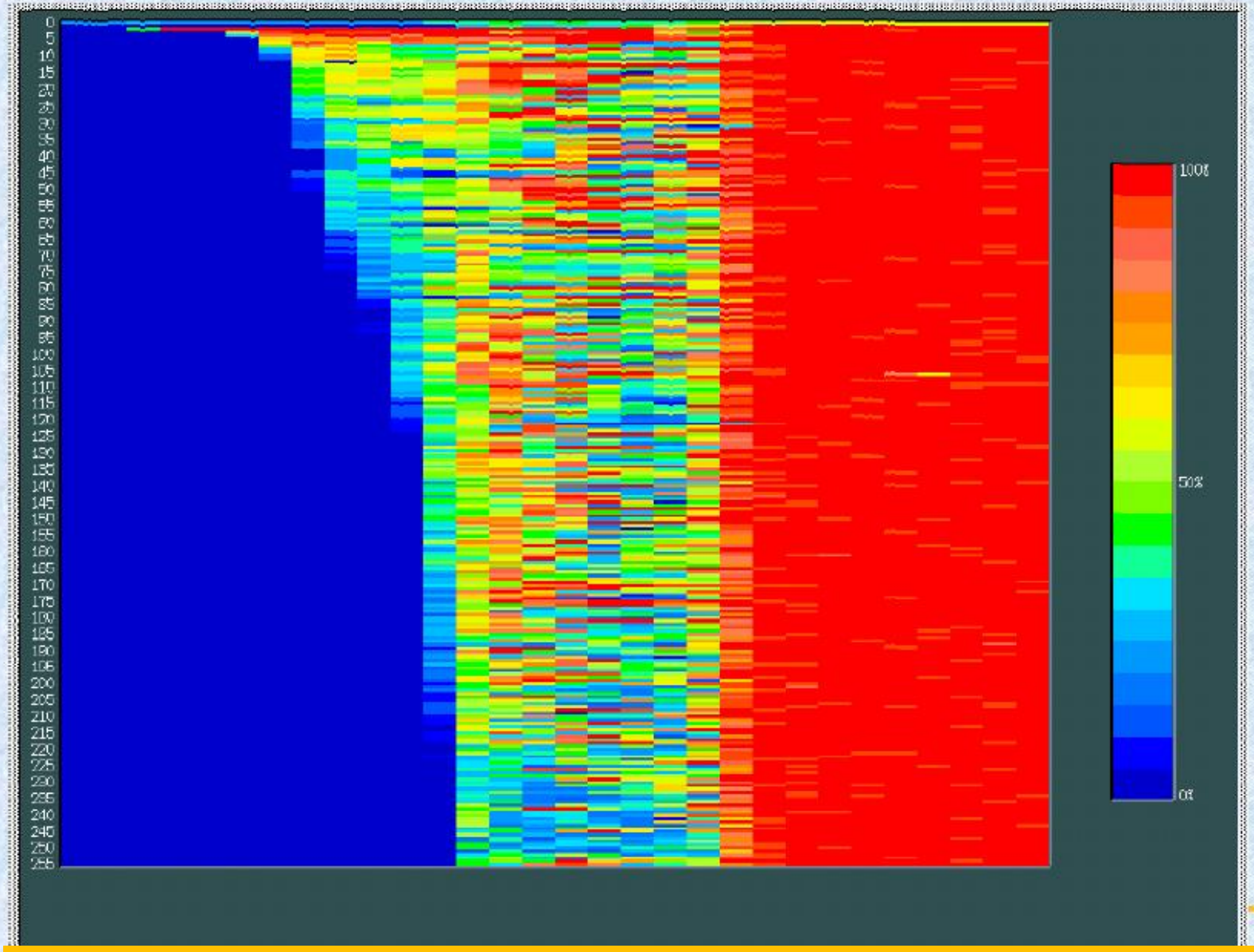
```
do(A) :- true | satchmo_problem:model(M), false(M,A).
false(M,A) :- true | satchmo_problem:nc(NC),
                     satisfy_clauses(0,NC,M,cl_sat,A).

satisfy_clauses(Cn,NC,M,A2,A) :- Cn < NC |
    Cn1 := Cn + 1,
    satisfy_ante(Cn1,[],[true|M],M,A2,A1),
    satisfy_clauses(Cn1,NC,M,A1,A).
satisfy_clauses(NC,NC,_,sat(M1), A) :- true | A = sat(M1).
satisfy_clauses(NC,NC,M,cl_sat, A) :- true | A = sat(M).
satisfy_clauses(NC,NC,M,unsat(Ms),A) :- true | A = unsat(Ms).

satisfy_ante(Cn,GS,[P|M2],M,cl_sat,A) :- true |
    satchmo_problem:c(Cn,P,GS,R),
    satisfy_ante1(Cn,R,P,GS,M2,M,A).
satisfy_ante(Cn,_,[],_,cl_sat,A) :- true | A=cl_sat.
otherwise.
satisfy_ante(_,_,_,_,A1,A) :- true | A=A1.
```

AND並列化(台数効果)



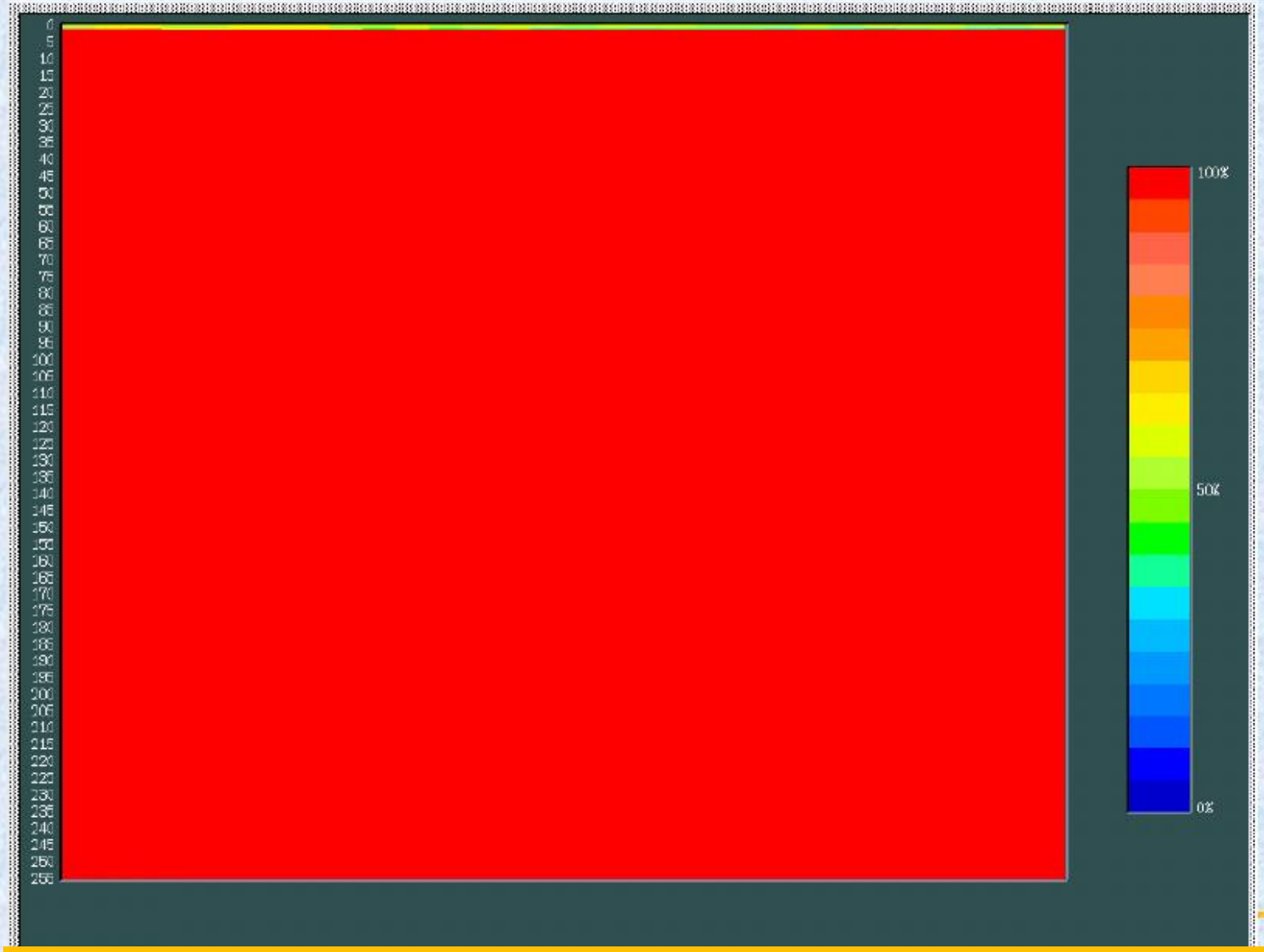
(長谷川 モデル生成型定理証明系 MGTPMGTPの要素技術、渕一博記念コロキウム『論理と推論技術:四半世紀の展開』)

N-逐次方式(開始時)



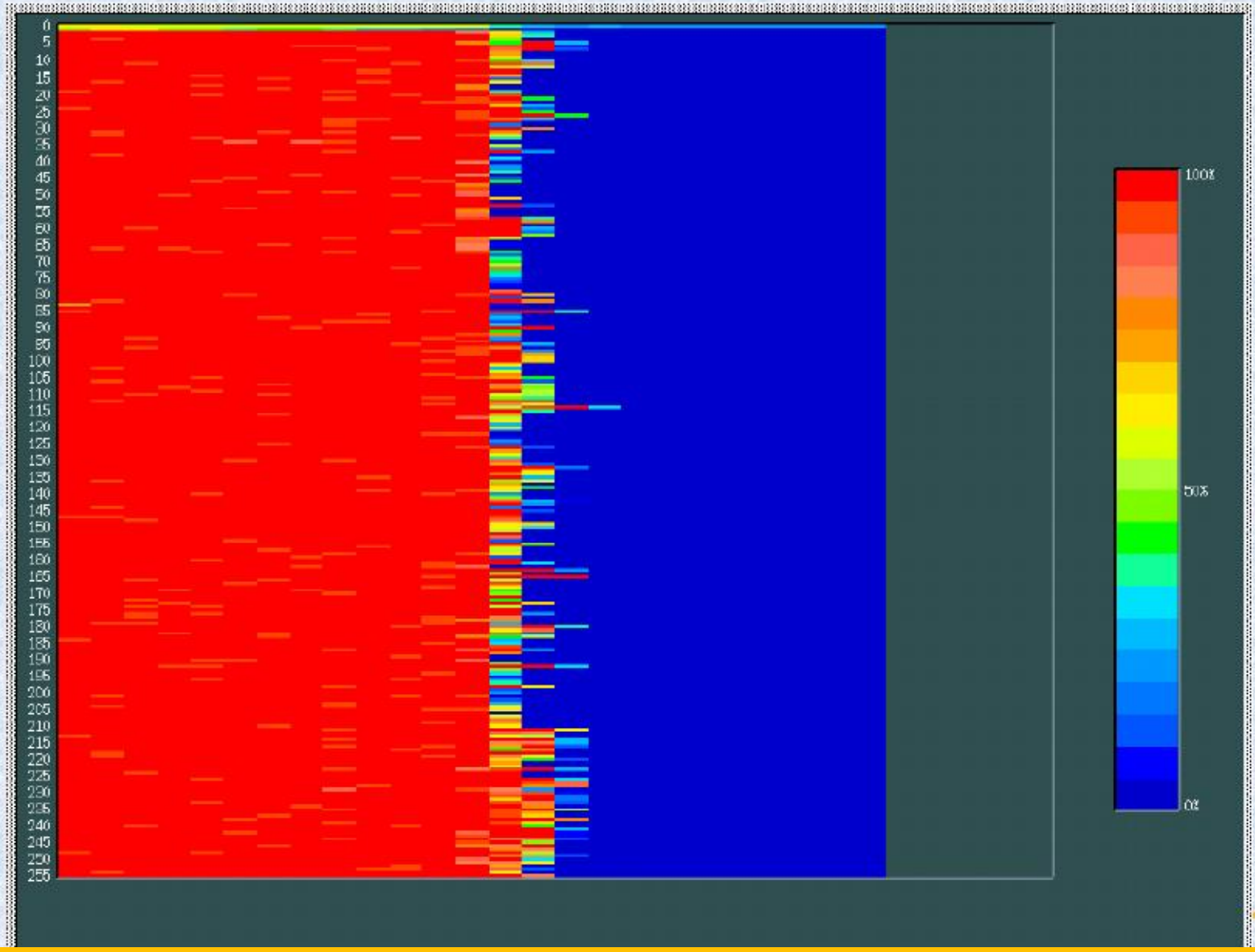
(長谷川　モデル生成型定理証明系 MGTPMGTPの要素技術、渕一博記念コロキウム『論理と推論技術:四半世紀の展開』)

N-逐次方式(途中)



(長谷川　モデル生成型定理証明系 MGTPMGTPの要素技術、渕一博記念コロキウム『論理と推論技術:四半世紀の展開』)

N-逐次方式(終了時)



(長谷川　モデル生成型定理証明系 MGTPMGTPの要素技術、渕一博記念コロキウム『論理と推論技術:四半世紀の展開』)

その他の主な研究成果

- 竹内・藤田によるPrologの部分計算プログラム
  - メタインタプリタの高速化（Production System、Bottom Up Parserなど）
  - エキスパートシステムの診断ルールの部分計算による導出(Goebel, Poole, Furukawa)
  - 自己適用可能部分計算プログラム (H. Fujita)
- 非単調推論、発想推論、帰納推論、類推の展開
  - 帰納論理プログラミングシステムPROGOLの調査
  - 発想論理プログラミングシステムALPの調査
  - 原口誠氏による類推の形式化

Prolog部分計算器

- Prologメタインタプリタ

```
solve(true).
solve((P,Q)) :- solve(P), solve(Q).
solve(G) :- clause(G,B), solve(B).
```

- Prolog部分計算器

```
psolve(true,true).
psolve((G1,G2),(R1,R2)) :-
    psolve1(G1,R1), psolve1(G2,R2).
psolve(G,R) :- clause(G,B), psolve1(B,R).
psolve1(G,R) :- proceed(G), psolve(G,R).
psolve1(G,G) :- stop(G).
```

国内外との連携

- 国内の大学関係者との連携：研究協議会の設置
  - 自然言語処理ワーキンググループ
  - 基礎理論ワーキンググループ、など
- 日英、日仏、日瑞伊、日米などの2~3国間ワークショップ
- 海外著名研究者の招聘(J.A.Robinson, Robert Kowalski, Ehud Shapiro, Keith Clark, Randy Goebel, Stephen Muggletonなど)
- 海外諸大学・研究所との研究交流

FGCSプロジェクトの総括

- FGCSは失敗したのか？
  - キラーアプリケーションの開発に失敗し、実用化に至らなかった。
- 成功したところも数多い。
  - 数多くの研究成果。
    - テクニカルレポート：911件
    - テクニカルメモ：1,322件
  - 人材の育成
  - 最も重要なポイント：国際的なイニシアティブ

スキルサイエンスへの挑戦

1995~

スキルサイエンス研究のきっかけ

- 慶応大学SFCキャンパスの風土
  - 新しい研究領域の開拓欲求
- 私自身のバックグラウンド
  - アマチュアチェリスト

    ➔大学オーケストラ、アマチュアオーケストラ、

    Logic Programming Trio

    (*Piano*: J.A. Robinson (*Resolution Principle*),

    ***Violin*: Jacques Cohen,**

    ***Cello*: Koichi Furukawa)**

Logic Programming Trioの活動

- ロジックプログラミング国際会議での演奏
  - JICSLP92　Washington　メンデルスゾーン　ピアノトリオ
  - ICLP95　Tokyo　ベートーベン　ピアノトリオ「大公」
  - ICLP96　Bonn　ベートーベンピアノトリオ　第3番
  - ICLP97 Leuven ドボルザーク　ピアノトリオ「ドゥムキー」
  - JICSLP98 Manchester　チャイコフスキー　ピアノトリオ「偉大なる芸術家の思い出」

スキルサイエンス関連プロジェクト

- 慶応義塾大型研究「帰納的方法論に基づく暗黙知の言語化」, 研究代表者, 1995-1996
- 科学研究費特定領域研究「発見科学」（研究代表者：有川節夫）研究分担者，サブテーマ「ILPによる知識発見」, 1998-2000
- 科学研究費基盤研究（A）「知識発見技術による身体スキルの言語化」, 研究代表者, 2005-2007
- 科学研究費基盤研究（C）「ルールアブダクションとアナロジーによるスキル創造」, 研究代表者, 2012-2014（研究分担者：藤波努、原口誠、金城啓太、研究協力者：尾崎知伸、升田俊樹、西山武繁）

人工知能学会全国大会でのセッション

- 近未来チャレンジセッション「身体知の解明に向けて」
  - 2003~2007
  - 人間は、芸術、スポーツの分野で計り知れない能力を発揮します。その速さと言い，精度と言い、驚くべきものがあります。…プロが発揮するこのような驚くべき能力の源は何なのか。それを解明するのが、「身体知の解明」の目的です。
- オーガナイズドセッション「身体知の表現と獲得」
  - 2008~（主査：藤波努）
  - 本セッションは、技の習得を可能にしている知的なものを身体知と捉え、それがどのように表現されるのか、またいかに伝えられるのかを議論する。

目次

- スピッカート奏法の習得
- ルールアブダクションによる説明構造の抽出
- アナロジカルアブダクション
- 比喩表現の役割

チェロのスピッカート奏法の習得について

古川康一　嘉悦大学
升田俊樹　チェリスト
西山武繁　慶応大学大学院政策メディア研究科
日本知能情報ファジィ学会誌，Vol.24 No.1．2012.

スピッカートとは？

- 高速（1秒間に3回～6回の往復運動）で弓を飛ばす奏法
- スピッカートは、弓を弾ませる上下方向と、音を出す左右方向の、両方の繰り返し運動を必要としている。
- 右手の動きに合せて、左手を動かす。
- https://www.youtube.com/watch?v=cJWjYLG3B7o

強制振動によるモデル化

- なぜ強制振動か？
  - 手首を支点とした振り子運動
  - 振り子が減衰しないようにエネルギーを注入
- ブランコ漕ぎからのヒント
  - 最大振幅のあと、最下点に達するまでの間に加速する。
- まりつきからのヒント
  - 外力による加速のショックをクッション動作により吸収する。

二つのヒントによる練習

- スピッカートの練習時に、「外力を最大振幅直後に与える」ことができるか？
  - 現象が速すぎる。
  - 遅いテンポで感触を掴むことはできる。
- 「クッション動作」は可能か？
  - 人差し指で弓を弦に押さえつけて、外力を与える。
  - クッション動作 =「柔らかく押さえつける」。
  - 弓の速い周期に合せなければならない。
  - **実は、ほかの方法があった。**

練習とメタ認知

- 繰り返し練習を加速するために、メタ認知を行った。
- メタ認知による気付きの例
  - 「手首を時計回りに回転させる」
  - 「いきなり出来るようになった」（1/14）
  - 「弓の毛を見るよりも，弓の棹を見る」（1/27）

- 練習開始（2010年12月23日）

- 突然できた時点（2011年1月15日）

- より進歩した時点（2011年1月28日）

- **ほぼ完成（2011年5月）**

ミニレッスンによる知見

- 専門家にミニレッスンを受け、弓の保持に薬指を使うことを指示された。
- その結果、不安定な動きが一気に解消され、安定的なスピッカートが可能になった。
- **薬指の利用は、「クッション動作」を可能にすることが判明した。（人差し指の調節ではなかった!）**

発想推論に基づく着眼点の発見

古川康一　慶応義塾大学
井上克巳　国立情報学研究所
小林郁夫　慶應義塾大学 SFC研究所
諏訪正樹　慶應義塾大学 環境情報学部
第23回人工知能学会全国大会 2009.

「コツ」と発想推論

- 「驚くべき事実」は、パースの発想推論(アブダクション、Abduction)に出てくる言葉で、その事実を説明するための仮説を求めるのが発想推論である。
- 「コツ」＝「驚くべき事実」は、新しいスキルを獲得する際のきっかけを与える。
- **「コツ」の有効性を発想推論によって示す！**

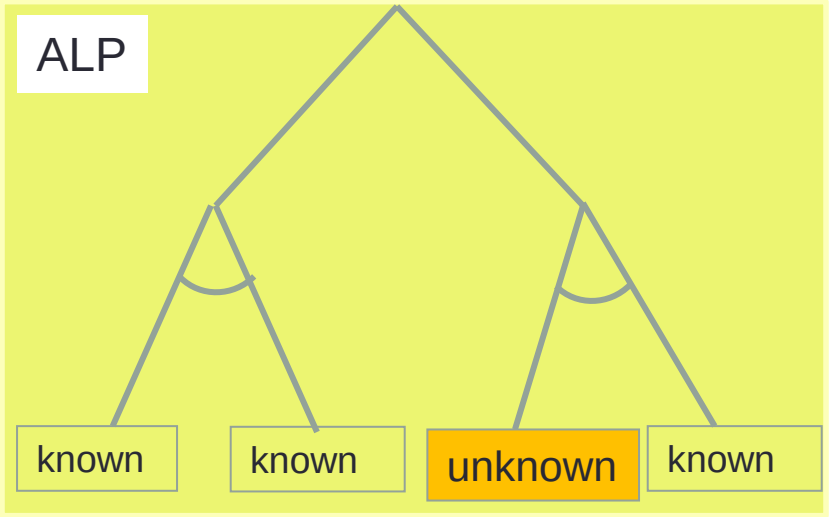
コツとアブダクション（発想推論）

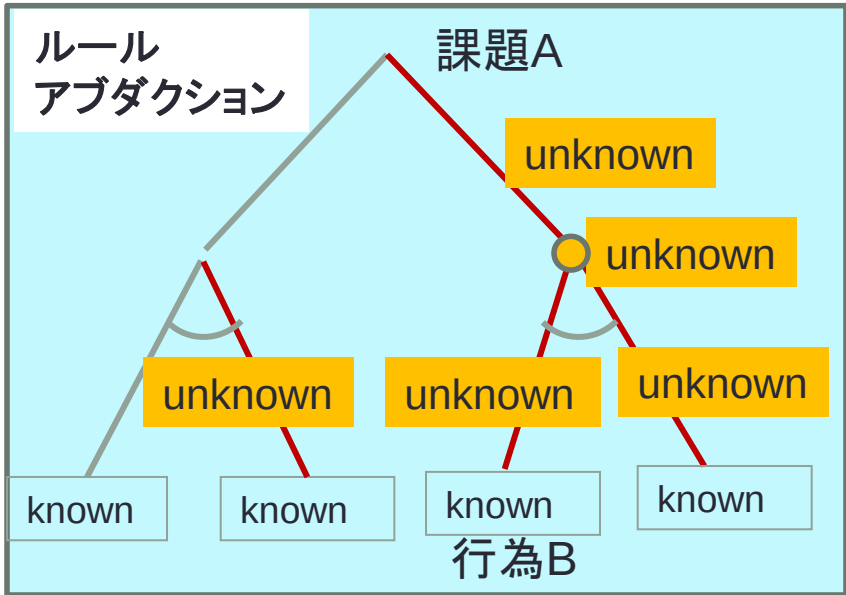
- 「コツ」は自明ではなく，腑に落ちる説明が必要である.
- コツの構造：「課題Aをこなすためには行為Bを行えばよい」
- 課題Aの証明図式の末端に行為Bが出現し，その間に欠落している仮説が存在する。
- すなわち「ルール」の仮説を必要とする.

- **ルール・アブダクション**の問題
- メタレベルアブダクションの利用 [古川09] [Inoue 09].

ALPの限界とルールアブダクション

- ALP（Abductive Logic Programming)の限界
  - ALPでは、ファクトしか仮説として提案することができない。
- 一階述語論理上のアブダクション＋メタレベルアブダクション
  - ルールも仮説として生成できる。
  - 述語発見ができる。





SOLARによるアブダクション

- SOLARは、フルセットの一階述語論理上の定理証明器
  - 背景知識**B**，観測事象**G**が与えられたとき，**アブダクション**は以下の2つを満足する仮説**H**を仮定可能述語**Γ**の中から求める.
  - 1. **B**∪**H** |= **G**　　(**B**∧¬**G** |= ¬**H**)
  - 2. **B**∪**H**は無矛盾である.

  B∧¬**G**の定理であって **B**の定理ではないような結論¬**H**を計算し、否定をとることで導く(Consequence Finding)。
- この際、SOL導出およびタブロー法を用いて効率的に**H**を計算するシステムがSOLARである。

Nabeshima, H., Iwanuma, K., and Inoue, K.: SOLAR: A Consequence Finding System for Advanced Reasoning, *Proc.* International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2003), LNCS, Springer, Vol.2796, pp.257-263(2003)

メタレベルアブダクション

- 因果関係（課題Aは行為Bにより達成される）をメタレベルのcaused述語によりアトム表現する。

  caused(spiccato, support_bow_with_ringfinger)

- 因果連鎖を表現するため、以下の公理を導入する。

  caused(A,B)←connected(A,B).

  caused(A,B)←connected(A,X)
  ∧caused(X,B).

  - “connected”：直接的な因果関係

オブジェクトレベルとメタレベル

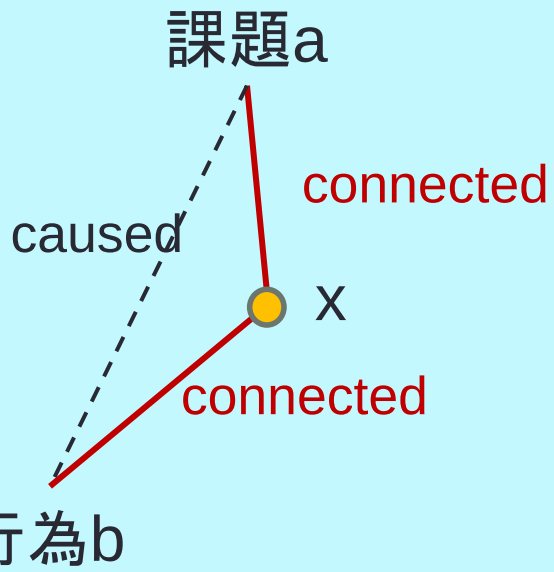
オブジェクトレベル

ルールアブダクション

?- a.

a:- x.
x :- b.

b.



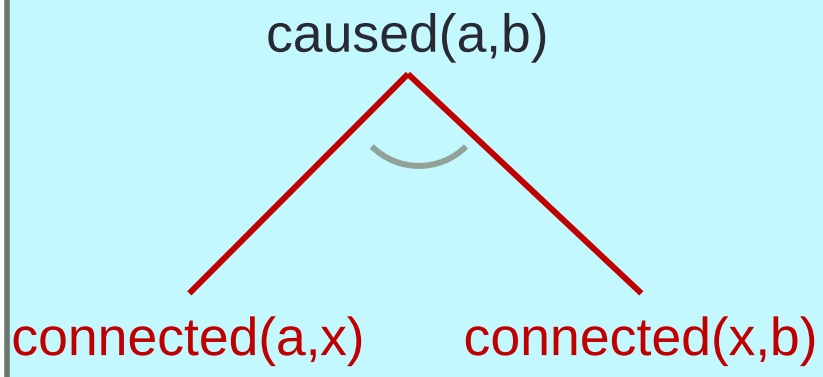
メタレベル

ファクトアブダクション

?- caused(a,b).

caused(X,Y)←connected(X,Y).
caused(X,Y)←connected(X,Z)
∧caused(Z,Y).

connected(a,x).
connected(x,b).



新ノードの導入

- SOLARでは、存在限量変数を含む仮説を生成できる。
- 以下のような仮説が生成されたとする。

$$\exists X.\ (\ \mathrm{connected}(g, X) \wedge \mathrm{connected}(X, s)\ ).$$

- X は新述語に対応している。

アブダクションの例

プログラム:

観測(G):caused(spiccato,
support_bow_with_ringfinger).

仮定可能述語:[connected/2]

背景知識(B):

解1:

connected(spiccato,
support_bow_with_ringfinger).

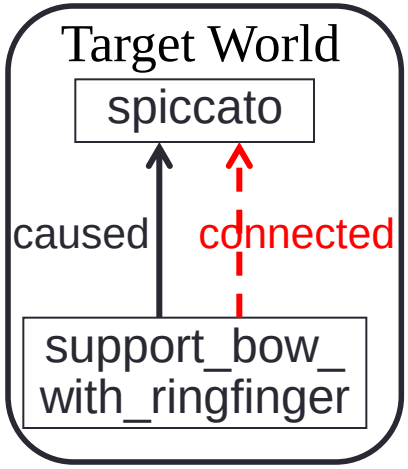
「薬指で弓を保持すればスピッカートができる」

解2:

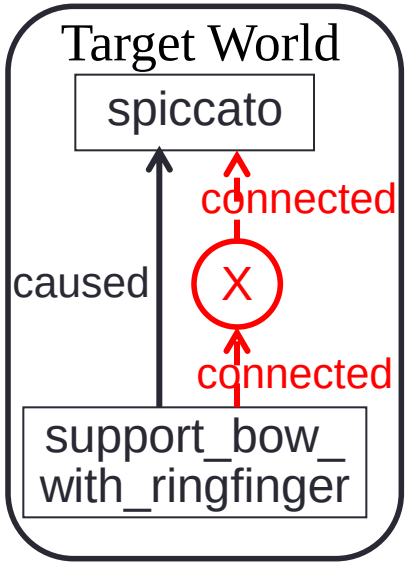
connected(spiccato, X) ∧
connected(X, support_bow_with_ringfinger).

「薬指で弓を保持すればXができ、Xができればスピッカートができる」

解1



解2



ルールアブダクションの効用と限界

- ルールアブダクションは、コツの説明に欠けていた”missing link”を明らかにする。
- すなわち、コツの説明の構造を導き出す。

- ただし、コツの腑に落ちる説明まではできない。
- 抽出されたコツの構造の解釈は、人間が与える。

- **この限界を打ち破るのが、アナロジカルアブダクション**

アナロジーを組み込んだルール発想推論によるスキル獲得支援

金城敬太　沖縄国際大学経済学部経済学科
尾崎知伸　日本大学文理学部情報科学科
古川康一　嘉悦大学大学院ビジネス創造研究科
原口　誠　北海道大学大学院情報科学研究科

アブダクションとアナロジー

- 事例「スピッカートをこなすためには薬指で弓を保持すればよい」というコツの説明を考える。
- ルール「薬指で弓を保持すればスピッカートができる」は，アブダクションで発見されるルールのひとつ
- しかし、アブダクションは妥当な説明の構造（証明木）しか与えない。

「ルールの根拠の説明」のためにアナロジーを用いる。

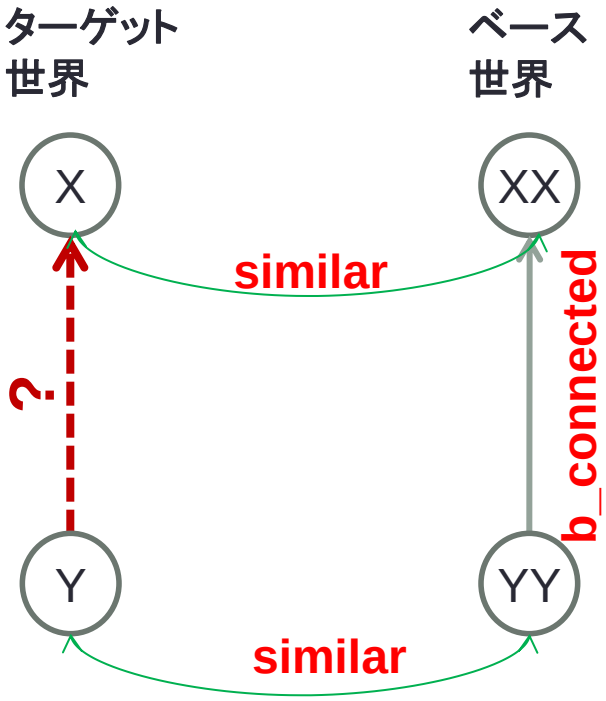
アナロジーの導入

- アナロジー:
  - ターゲット世界と類似のベース世界で成り立つ関係を元の世界に持ち込んで推論を続ける方法。
- ベース世界:
  - ベース世界で事象XXとYYの間に成り立つ直接因果関係: **b_connected(XX,YY)**
- ターゲット世界:
  - ターゲット世界で事象XとYの間に成り立つ直接因果関係: **t_connected(X,Y)**
- 類比:
  - ベース世界の事象XXとターゲット世界での事象Xの間に成り立つ類似性: **similar(X,XX)**

アナロジーの導入：類推公理

- 類推のための公理

connected_by_analogy(X,Y) ←
b_connected(XX, YY),
similar(X,XX),
similar(Y,YY).



ベース世界において**b_connected(XX,YY)**が成り立ち、さらに、類比**similar(X,XX)**、**similar(Y,YY)**が成り立つとき、ターゲット世界において**connected_by_analogy(X,Y)**が成立。

アナロジカルアブダクションの公理

ベース世界での公理

b_caused(X,Y):- b_connected(X,Y).
b_caused(X,Y):- b_connected(X,Z), b_caused(Z,Y).

ターゲット世界での公理

t_caused(X,Y):- t_connected(X,Y).
t_caused(X,Y):- t_connected(X,Z), t_caused(Z,Y).

類推公理

connected_by_analogy(X,Y):-
b_connected(XX, YY), similar(X,XX), similar(Y,YY).

連結の定義

t_connected(X,Y)←**originally_connected**(X,Y).
t_connected(X,Y)←**connected _by_abduction**(X,Y).
t_connected(X,Y) ←**connected_by_analogy**(X, Y)
∧print_connected_by_analogy(X, Y).

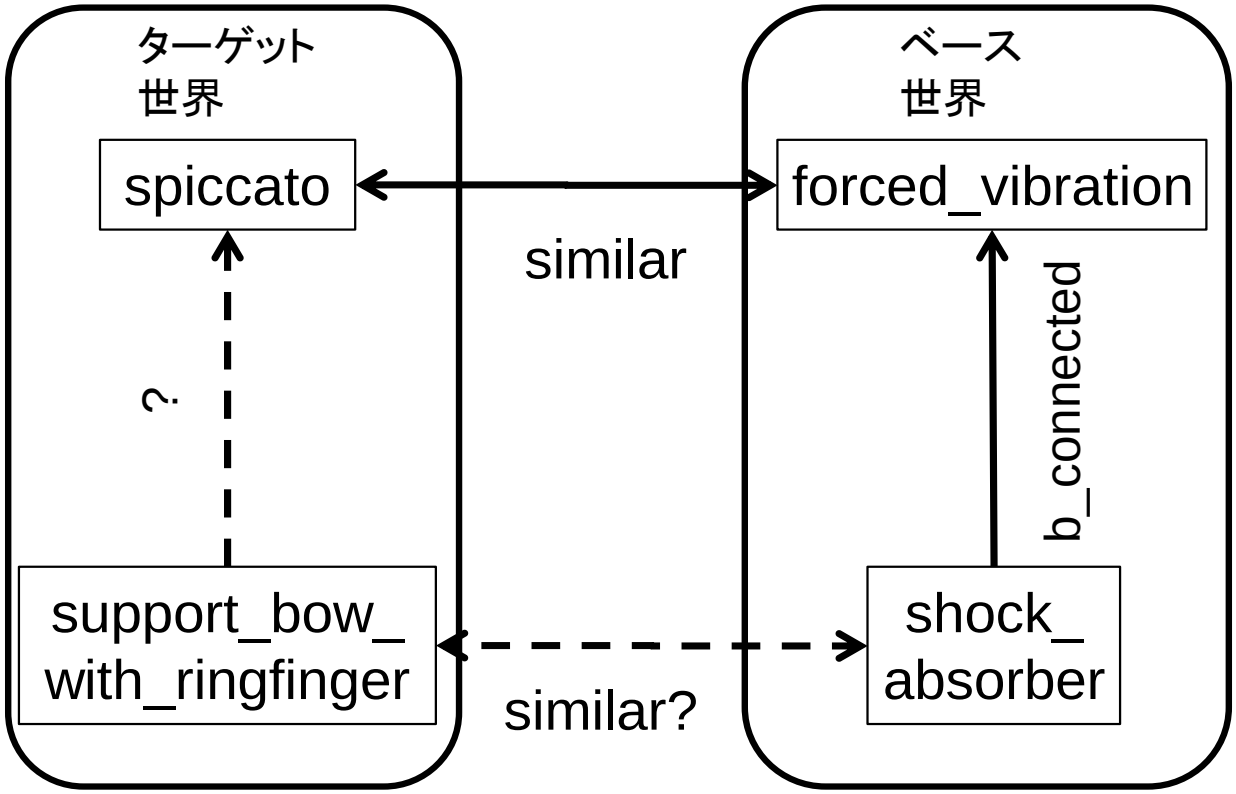
1.originally_connected：直接連結を事実として与えるときに利用.
2.**connected _by_abduction**：直接連結を仮説として導入するときに利用.
3.**connected_by_analogy**：アナロジーを用いて仮説を導入するときに利用.

アナロジカルアブダクションの実行

- **問題**：スピッカート奏法を実現させるために必要とされる「薬指で弓を保持する」ことの理由を，強制振動との類推で説明する問題

■ スピッカート奏法でのアナロジカルアブダクション

■ 破線の部分を仮説として求める.



- **プログラム**:

観測(G): t_caused(spiccato,
support_bow_with_ringfinger).

仮定可能述語(Γ):
[connected_by_abduction, similar,
print_connected_by_analogy]

背景知識(B):

ベース世界:b_connected(forced_vibration,
shock_absorber).

ターゲット世界:

←connected_by_abduction
(spiccato,support_bow_with_ringfinger).

類比:similar(spiccato, forced_vibration).

SOLARの実行結果

- 推論深度上限10，仮説長制限4とした場合，このプログラムの実行により，7つの解.

print_connected_by_analogy(spiccato,
support_bow_with_ringfinger) ∧
similar(support_bow_with_ringfinger, shock_absorber)

⇒**薬指による弓の保持が強制振動でのショックアブソーバの役割を果たしていることが示された。**

アナロジカルアブダクションの比喩表現への応用

- 比喩は、類推の一種である。
- 合奏指導に著しい効果を発揮する比喩表現をアナロジカルアブダクションで形式化した。
- 分かったこと:
  - 類推は、**説明**の役割を果たす。
  - 比喩は、**イメージの増強**の役割を果たす。

sfp（スフォルツァートピアノ）の奏法

- ***sfp***（スフォルツァートピアノ）というアクセントの一種

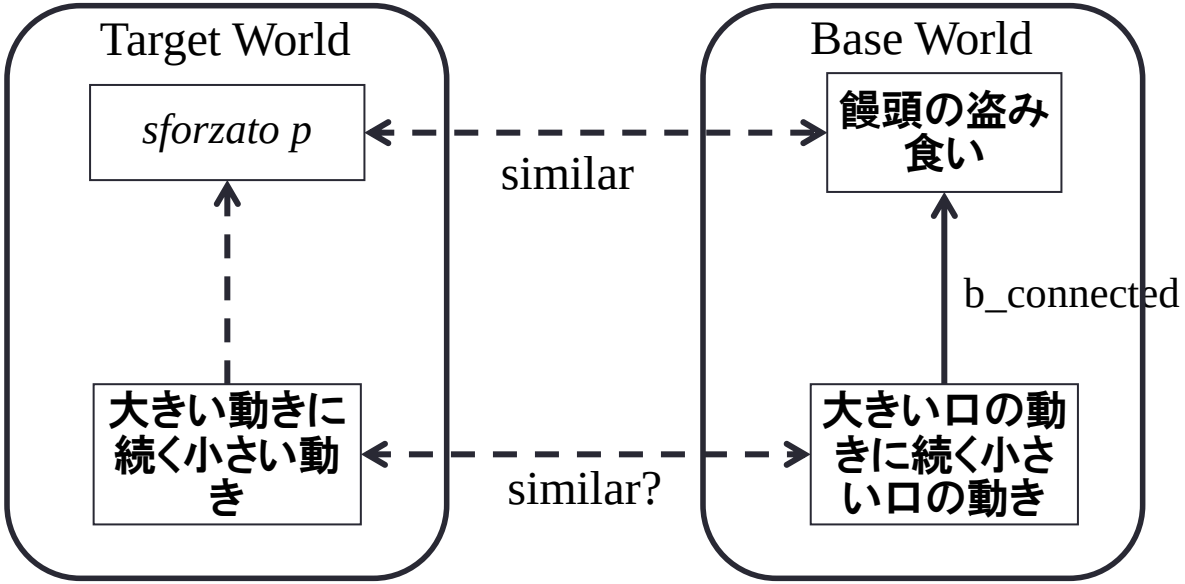
  ある指揮者が「饅頭の盗み食い」と表現したのには的を射ていると納得した。

- **<饅頭の盗み食いのイメージ>**

  誰も来ないのを見計らって饅頭を口に放り込むのが***sf***、人が来たので口を閉じて、アンコの甘みを楽しむのが***p***

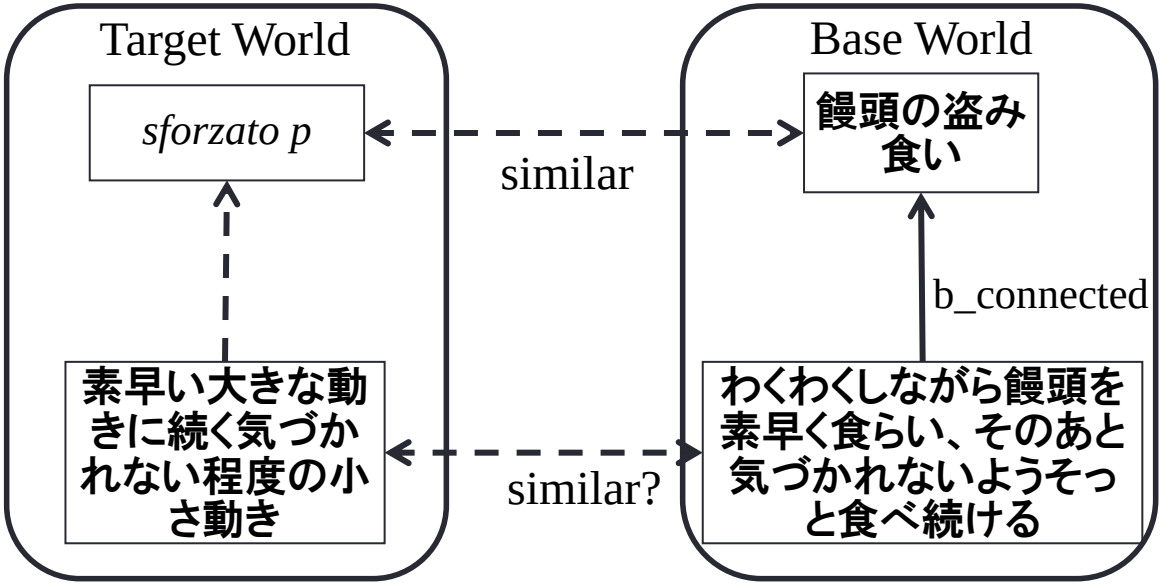
メタファーの有効性を説明する

- ***sfp***の弓の動きを「饅頭の盗み食い」の口の動きでまねる。
- 「饅頭の盗み食い」に伴う**動きの意味**が重要。
- 動きの意味は、プレーヤー間で共有できる。
- アナロジカルアブダクションで形式化できる。
- 動きを伝えるアナロジー:



比喩表現によるイメージの増強

- 比喩表現は、動きだけでなく、その場にふさわしいイメージの増強が図られる。
- 盗み食いするときの素早い行動、そのあとの、ばれないようにするための細心の注意、など。



諏訪正樹氏とのパーソナルコミュニケーション

今後の課題

- コーチング・トレーニングのための比喩表現の創出
- チェロ奏者のための副読本の執筆（升田俊樹氏とともに）
- チェロの腕を磨く
- 老齢化への対処（省エネ奏法の追究）
- 人と室内楽ができるバイオリンロボットの開発！

第五世代コンピュータプロジェクトとスキルサイエンスを振り返って

- 5Gプロジェクトは、多くの独創的な研究成果を上げることができた。
- 優秀な人材を数多く輩出した。
- 我が国のAI研究を海外に飛躍させることに貢献できた。
- 私自身、第五世代コンピュータプロジェクトからスキルサイエンスへ大きな舵を切ったが、背後には論理プログラミングがあった。
- 5G時代のメタプログラミング・帰納推論・発想推論の取り組みがスキルサイエンスに生かされた。

謝辞

- 人工知能学会における5Gプロジェクトおよびスキルサイエンスへの支援・評価に深謝する。
- 多くの指導者、共同研究者、関係者に謝意を表する。

ご清聴ありがとうございました。

ヒトは技を磨く動物である（ホモスキルズ）

演奏曲目

Max Bruch (1838-1920)

Kol Nidrei (典礼歌)

ピアノ: 藤波努、チェロ: 古川康一