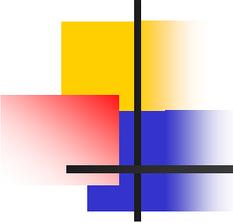


# 帰納・発想論理プログラミングとス キルサイエンス

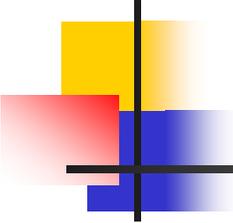
慶應義塾大学  
古川康一



# 身体性、身体スキルへの関心

---

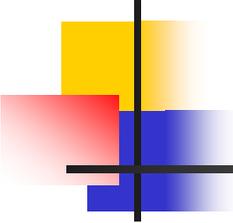
- 身体について、我々はよく知らない。
- 知性は、身体の諸機能の上に宿る。
  - 氷山のよう
- 身体知は、身体性と知性を結びつけるリンクである。
- 身体知の解明は、人工知能、認知科学、スポーツ科学に対する新しい切り口を与える。



# スキルサイエンスで何故帰納・ 発想推論か

---

- 演奏方法から演奏結果を導くのは、演繹的な推論である。
- 演奏結果から演奏方法を求めるのは、逆向きの推論、すなわち帰納・発想推論である。
- 帰納推論は、演奏データからの演奏ルールの抽出を可能にする。
- 発想推論は、与えられた課題をこなすための演奏法の創造を可能にする。

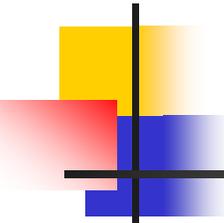


# スキルサイエンスと論理

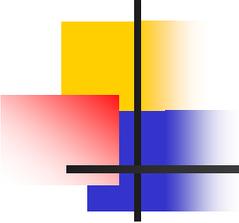
---

- スキルサイエンスの対象は、動きなどの信号データである。
- 信号データを記号化することによって、論理での扱いが可能となる。
- 記号化には、データ処理による手法と、言語化による手法がある。
- 我々は、その両者を採用する。

# Peirceによる演繹、帰納、発想 の定義



- 演繹推論：一般的ルールを特定のケースに当てはめて、結論を得る分析的過程
- 帰納推論：特定のケースと結論から、ルールを推論する合成的過程
- 発想推論：ルールと結論から、特定のケースを推論する、もう1つの合成的過程



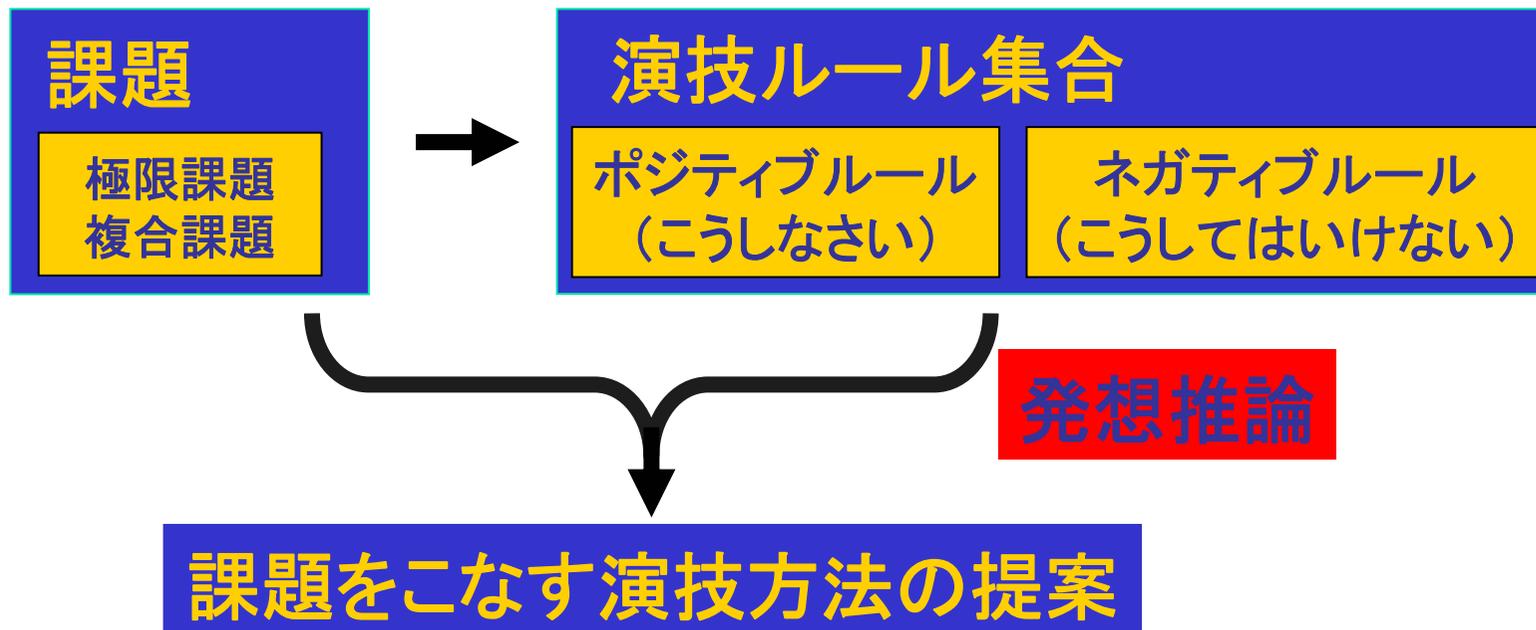
# 演繹推論・発想推論・帰納推論

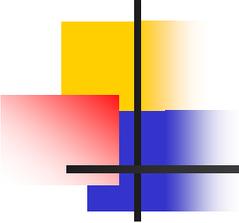
---

- A: すべての人間は死ぬ.                      大前提
  - B: ソクラテスは人間である.                小前提
  - C: ソクラテスは死ぬ.                        帰結
- 
- 演繹推論: AとBからCを導く
  - 発想推論: AとCからBを導く
  - 帰納推論: いくつかのBとCの対からAを導く

# 発想推論によるスキル創造支援

- 発想推論システムは、スキル創造支援システムの構築を可能にする。

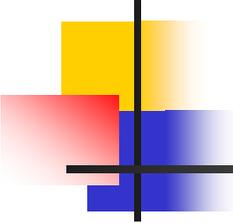




# ACLPLによる複合課題の 奏法探索

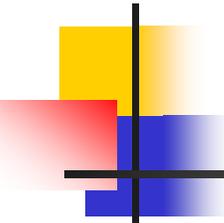
---

- *ACLPL*(Abductive Constraint Logic Programming)は、発想推論を可能にする論理プログラミングである。
- *ACLPL*によって、一貫性制約を満たす複合課題の奏法を探索する。
- *ACLPL*の実用システム *ProLogICA*で実装する。( *ProLogICA*は、変数を含む仮説を許さない)



# ACL<sub>P</sub>の概要

- 発想推論の形式的枠組み:  $\langle T, A, I \rangle$   
 $T$ は与えられた理論(論理プログラム),  
 $A$ は候補仮説集合(abducible),  $I$ は一貫性制約.
- 与えられた理論  $T$ と一貫性制約  $I$ , 観測事実  $G$ に対して, 発想推論は以下の2つの式を満足する  $\Delta \in A$ を求める.
  1.  $T \cup \Delta \models G$  (  $T$ は仮説  $\Delta$ を補うこと  
によって  $G$ を説明できる)
  2.  $T \cup \Delta$ は  $I$ を充足する (  $T \cup \Delta$ は一貫性  
制約  $I$ を満たす)



# 発想推論のためのスキル表現 言語の語彙

---

- 課題の名前

clearSound\_with\_rapidPositionShift, longTone

- 体の各部分の動き

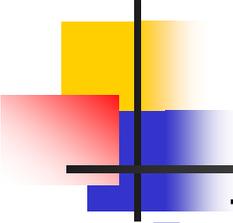
fixShoulder, closeLeg

- 筋肉(群)の活性状態

activeForeMsclsWeak

- 体の(各部分の)力学状態の表現

spineHighImp, bigInertiaMoment



# 正のルールの表現 (1)

---

- 基本課題定義ルール:

- 基本課題を体の動きにより表現する。

rapidPositionShift :- rapidAddAbdOfShoulder,  
addAbdOfElbow.

- 複合課題定義ルール:

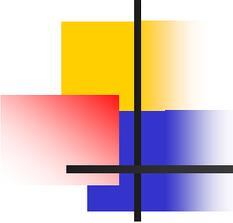
- 複合課題を既存の課題の組み合わせによって定義する。

rapidPositionShift\_with\_vibrato :-  
rapidPositionShift, vibrato.

- 体の動きの定義ルール:

- 体の動きを筋肉の活性状態で定義する。

flexKnuckleLR :- activeForeMsclsWeak,  
activeUpperMsclsWeak.



## 正のルール表現 (2)

---

- 前進推論のためのタクソノミールール:

- 既知の述語の効果を定義する。

bigInertiaMoment :- addAbdOfShoulder.

- 後退推論のためのタクソノミールール:

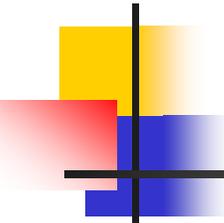
- 抽象的な動きを定義する。

turnWithFlexHandLR :- flexKnuckleLR.

turnWithFlexHandLR :- flexWristLR.

- 動きの伝播ルール:

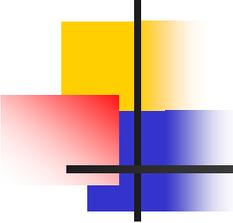
bowingParallelToString :- rightArmShrink.



# 負のルールの表現– Motion Integrity Constraints (MIC)

---

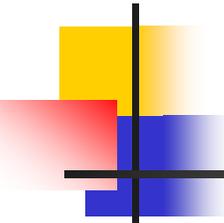
- MIC: 与えられた動きの間中満足すべき一貫性制約。
  - 熟達者は、そのようなMICを守っている。
  - MICは、間違っただメンタルモデルによる無駄な動きを回避するのに役立つ。
- MICの類別
  - 力学的MIC
    - 慣性モーメントの調節に関する制約
    - 振り子運動での支点の固定に関する制約
  - 筋骨格MIC
    - インピーダンス調節に関する制約
    - 協調動作の調節に関する制約
  - 時空間MIC
    - 空間制約
    - 時間制約



# 一貫性制約違反

---

- 制約違反は、極限課題、あるいは、複合課題の実行中に発生する。
- 極限課題での制約違反の例
  - 弓の先での弓の返しでは、腕を十分に伸ばすことと、振り子の支点となる肩を固定することを同時に達成しなければならない。この二つの要件は、肩筋肉のインピーダンス制約を引き起こす恐れがある。
- 複合課題での制約違反の例
  - 一方の課題のある奏法が上腕筋の強活性化を要求し、他方が強活性化の抑制を要求している。
  - 一方の課題のある奏法が大きな慣性モーメントを生成し、他方の課題が早い動きを必要とする。



# 発想推論によるプログラム例

---

## 1. 創造支援の例

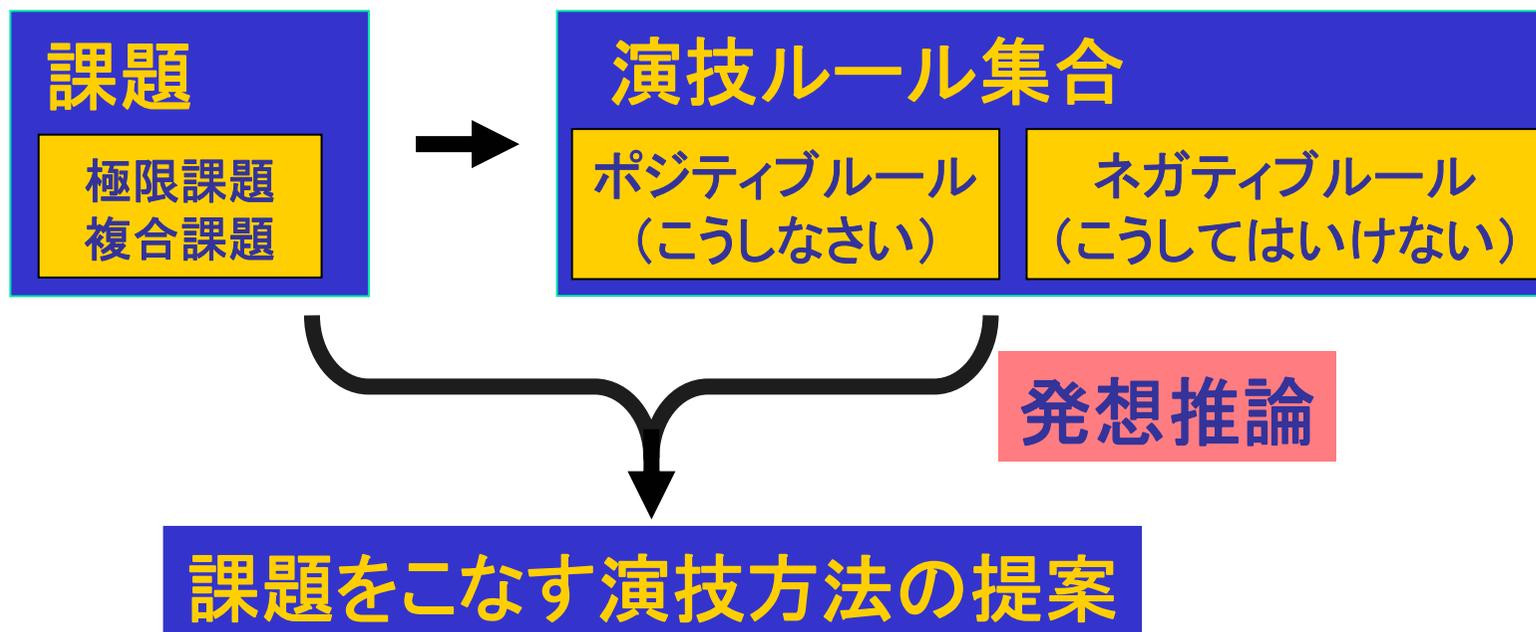
1. 複合課題の実現(ポジション移動とビブラート)
2. 極限課題の実現(発達段階のシミュレーション)

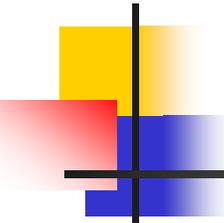
## 2. 診断支援の例

1. 右手と左手の相互干渉の例

# 発想推論によるスキル創造支援

- 発想推論システムは、スキル創造支援システムの構築を可能にする。





# 発想推論による創造支援の例

---

- 課題:

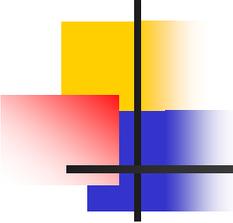
- ビブラートをかけながらの高速ポジション移動

- 課題の困難性:

- 高速ポジション移動は左腕の速い動きを必要とし、そのため筋肉の高活性化を伴う。
- 一方、ビブラートは、左腕の筋肉の活性化を嫌う。

- 解:

- 我々は、両課題に対して、複数の演奏法を用意し、問題を回避できるような組み合わせを求める。



# 発想推論による形式化

- 動作一貫性制約：
  1. 「上腕の位置の固定」と「肩関節の内外転」は矛盾する。
  2. 「高速運動」と「大きな慣性モーメント」は矛盾する。
- abducible:
  - 肩の高速内外転、左上腕筋の強活性化、上腕の内外回、上腕の固定、前腕の内外回、肘の微小内外転
- 矛盾のない解：
  - 「前腕の振り出し振り戻しによってポジション移動を行い、前腕の微小捻り運動によってビブレードを行う」

# PrologICAによるプログラミング

rapid\_position\_shift\_with\_vibrato :- rapid\_position\_shift,vibrato.  
(Compound Task Def.)

rapid\_position\_shift :- rapid\_add\_abdOfShoulder,add\_abdOfElbow.

rapid\_position\_shift :- in\_exCycloOfUpperarm,  
pronosupinationOfForearm. (Basic Task Def.)

add\_abdOfShoulder :- rapid\_add\_abdOfShoulder. (Forward Taxonomy)

rapid\_movement :- rapid\_position\_shift. (Forward Taxonomy)

big\_inertia\_moment :- add\_abdOfShoulder. (Forward Taxonomy)

vibrato :- fixUpperarm,micro\_add\_abdOfElbow. (Basic Task Def.)

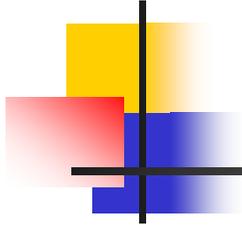
vibrato :- pronosupinationOfForearm. (Basic Task Def.)

add\_abdOfElbow :- active\_left\_upperarm\_mscls\_strong. (Movement Def.)

ic :- fixUpperarm, add\_abdOfShoulder. (due to spatio-temporal cond.)

ic :- rapid\_movement, big\_inertia\_moment. (due to dynamics)

ic :- pronosupinationOfForearm, active\_left\_upperarm\_mscls\_strong.  
(due to neuromuscular systems)



## ■ Definition of Abducibles

modeh(\*,rapid\_add\_abdOfShoulder).

modeh(\*,in\_exCycloOfUpperarm).

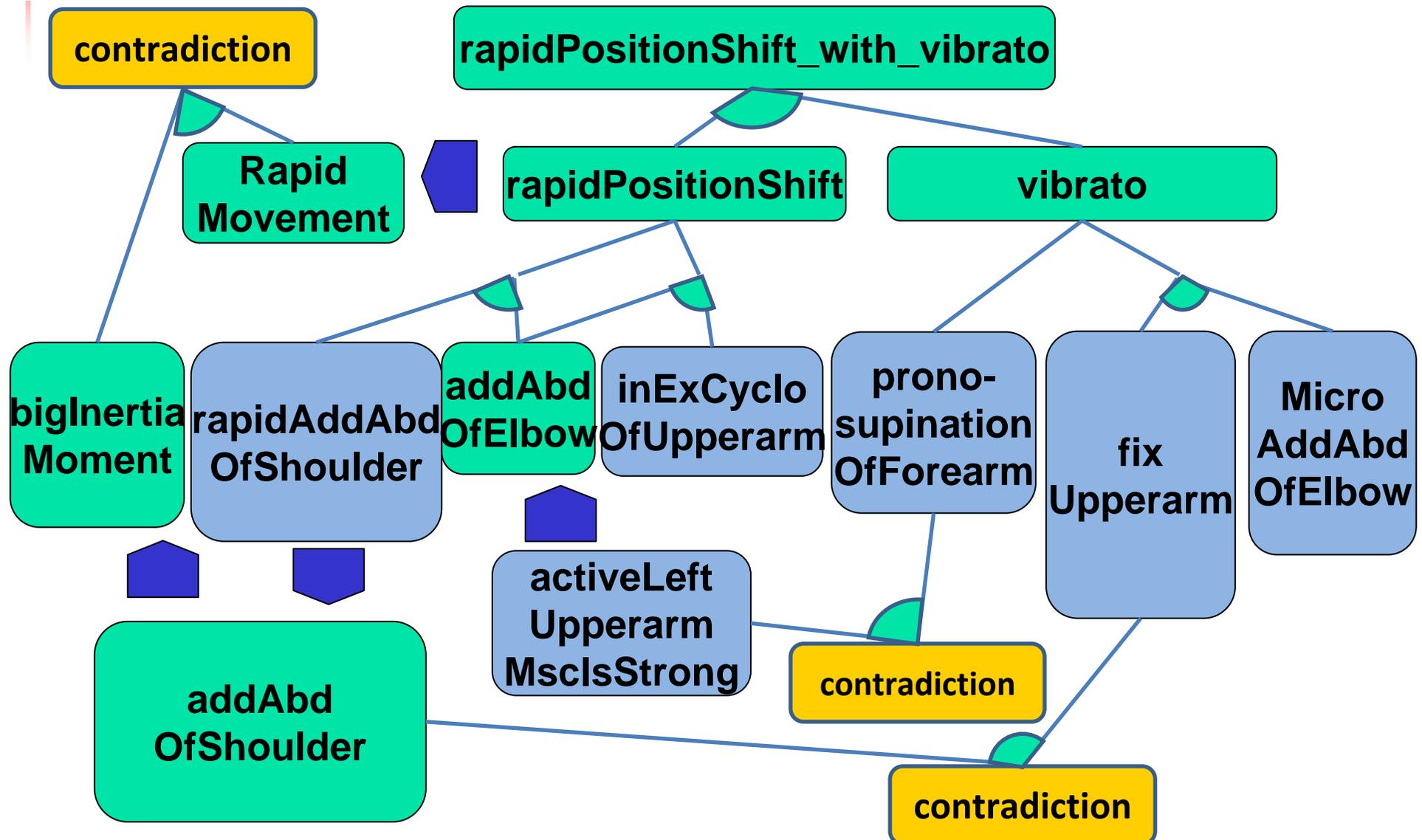
modeh(\*,fixUpperarm).

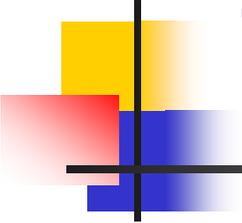
modeh(\*,pronosupinationOfForearm).

modeh(\*,micro\_add\_abdOfElbow).

modeh(\*,active\_left\_upperarm\_mscls\_strong).

# Scheme of Example Program





# 実行結果

---

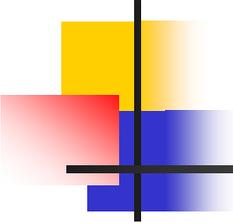
| ?-

```
demo([rapidPositionShift_with_vibrato],  
[],D).
```

```
D = [micro_add_abdOfElbow,  
fixUpperarm,pronosupinationOfForearm,  
in_exCycloOfUpperarm] ? ;
```

```
D = [pronosupinationOfForearm,  
in_exCycloOfUpperarm] ? ;
```

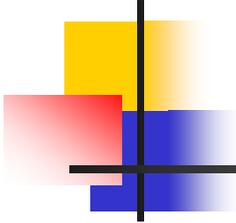
no



# 実行結果の解釈

---

1.  $D = [\text{pronosupinationOfForearm}, \text{in\_exCycloOfUpperarm}]$ 
  - この結果は、前腕の振り出し振り戻しによるポジション移動とビブラートを示唆している。(予想した結果)
2.  $D = [\text{micro\_add\_abdOfElbow}, \text{fixUpperarm}, \text{pronosupinationOfForearm}, \text{in\_exCycloOfUpperarm}]$ 
  - この結果は、ビブラートのかけ方としては微小なポジション移動でもかまわないことを示している(予想しなかった結果)



# 発想推論によるプログラム例

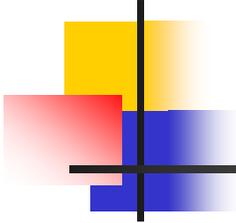
---

## 1. 創造支援の例

1. 複合課題の実現(ポジション移動とビブラート)
2. 極限課題の実現(発達段階のシミュレーション)

## 2. 診断支援の例

1. 右手と左手の相互干渉の例



# 発達段階とデフォルト推論

- 発達段階に応じて通常の課題と極限課題に分ける。
  - 通常の課題: デフォルトルールで表現する。
  - 極限課題: 例外的な高度なスキルとして表現する。
- 例
  - 通常の弓の返し技法がデフォルトに相当し、極限的な状況での弓の返し技法を例外として対処する。
  - 弓の先での弓の返しでは、腕を十分に伸ばすことと、振り子の支点となる肩を固定することを同時に達成したければならない。この二つの要件は、肩筋肉のインピーダンス制約を引き起こす恐れがある。
  - 例外的なスキルでは、肩を固定するために左足を閉じる奏法を採用する。左足を閉じると、床との摩擦で足が開きにくくなり、その結果背筋が伸びて、肩が固定される。

# 発達段階とデフォルト推論 (つづき)

tipTurnQuick :- longTone, turnQuick, not extremeTask.

tipTurnQuick :- extremeTask, extremeLongTone,  
extremeTurnQuick.

- デフォルトルール

turnQuick :- turnWithFlexHandLR.

turnWithFlexHandLR :- flexKnuckleLR.

turnWithFlexHandLR :- flexWristLR.

flexKnuckleLR :- activeForeMsclsWeak, activeUpperMsclsWeak.

flexWristLR :- activeUpperMsclsStrong.

- 例外ルール

extremeLongTone :- stretchArm.

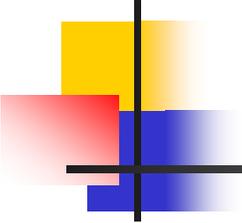
extremeTurnQuick :- shoulderFix, turnQuick.

shoulderFix :- closeLeg.

shoulderFix :- spineHighImp, upperArmHighImp.

spineHighImp :- activeWaistMscls.

upperArmHighImp :- activeShoulderMscls.



# 発達段階とデフォルト推論 (つづき)

---

- 一貫性制約

ic :- stretchArm, activeShoulderMscls.(腕の伸ばしと肩筋の活性化)

- 仮説候補

abducible\_predicate(extremeTask).

abducible\_predicate(longTone).

abducible\_predicate(stretchArm).

abducible\_predicate(activeUpperMsclsStrong).

abducible\_predicate(activeForeMsclsWeak).

abducible\_predicate(activeUpperMsclsWeak).

abducible\_predicate(closeLeg).

abducible\_predicate(activeWeistMscls).

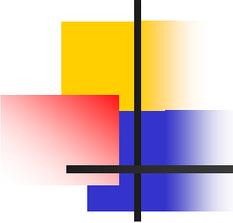
abducible\_predicate(activeShoulderMscls).

abducible\_predicate(flexKnuckleLR).

abducible\_predicate(flexWristLR).

# 発達段階とデフォルト推論 (つづき)

```
| ?- demo([tipTurnQuick],[],D).  
D = [flexKnuckleLR,longTone] ? ;  
D = [flexWristLR,longTone] ? ;  
D =  
  [flexKnuckleLR,closeLeg,stretchArm,extremeTask] ? ;  
D = [flexWristLR,closeLeg,stretchArm,extremeTask] ? ;  
no
```



# 発想推論によるプログラム例

---

## 1. 創造支援の例

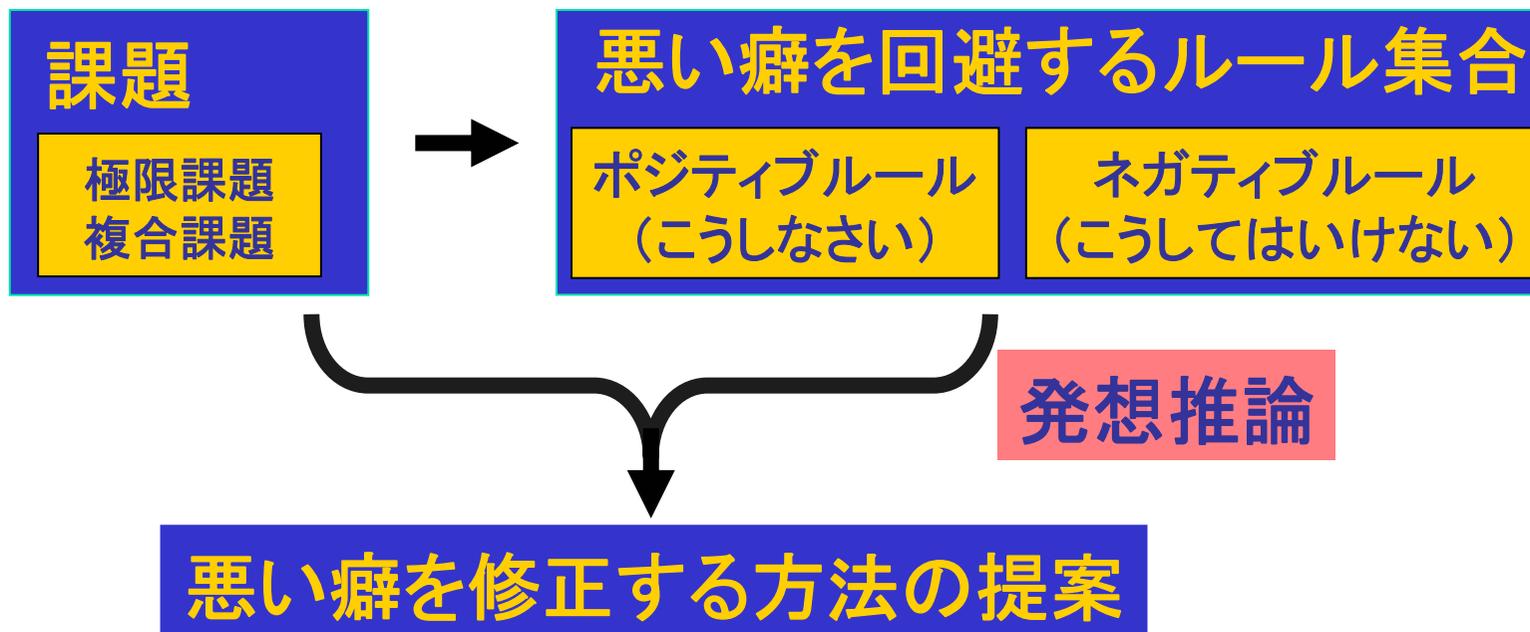
1. 複合課題の実現(ポジション移動とビブラート)
2. 極限課題の実現(発達段階のシミュレーション)

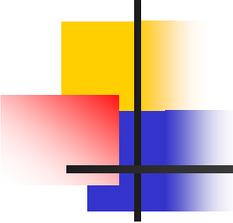
## 2. 診断支援の例

1. 右手と左手の相互干渉の例

# 発想推論による身体スキル診断

- 発想推論によって、身体スキル診断を実現する。





# 身体スキル診断の例

---

- 症状：
  - 左腕による高速ポジション移動は、雑音を生成しがちである。
- 考えられる理由：
  - 左腕と右腕は、注意していないと同時に縮んでしまう。
- 処方：
  - この同時伸縮が発生しがちであることに注意を払うこと
  - によって、この現象の発生を回避する。
- 発想推論は、この処方を導いてくれる。

# Skill Diagnosis Example

clearSound\_with\_rapidPositionShift :- clearSound,  
rapidPositionShift.

rapidPositionShift :- addAbdOfShoulder, addAbdOfElbow.

rapidPositionShift :- inExCycloOfUpperarm, addAbdOfElbow.

bowingParallelToString :- rightArmShrink.

(Movements propagation . Forward rule)

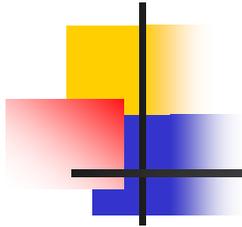
rightArmShrink :- leftArmShrink, not conscious\_on\_armShrink

(Default by unconsciousness. Forward rule)

leftArmShrink :- addAbdOfShoulder, addAbdOfElbow.

(Forward Taxonomy)

ic :- clearSound, bowingParallelToString. (Due to dynamics)



## ■ Definition of Abducibles

modeh(\*, clearSound).

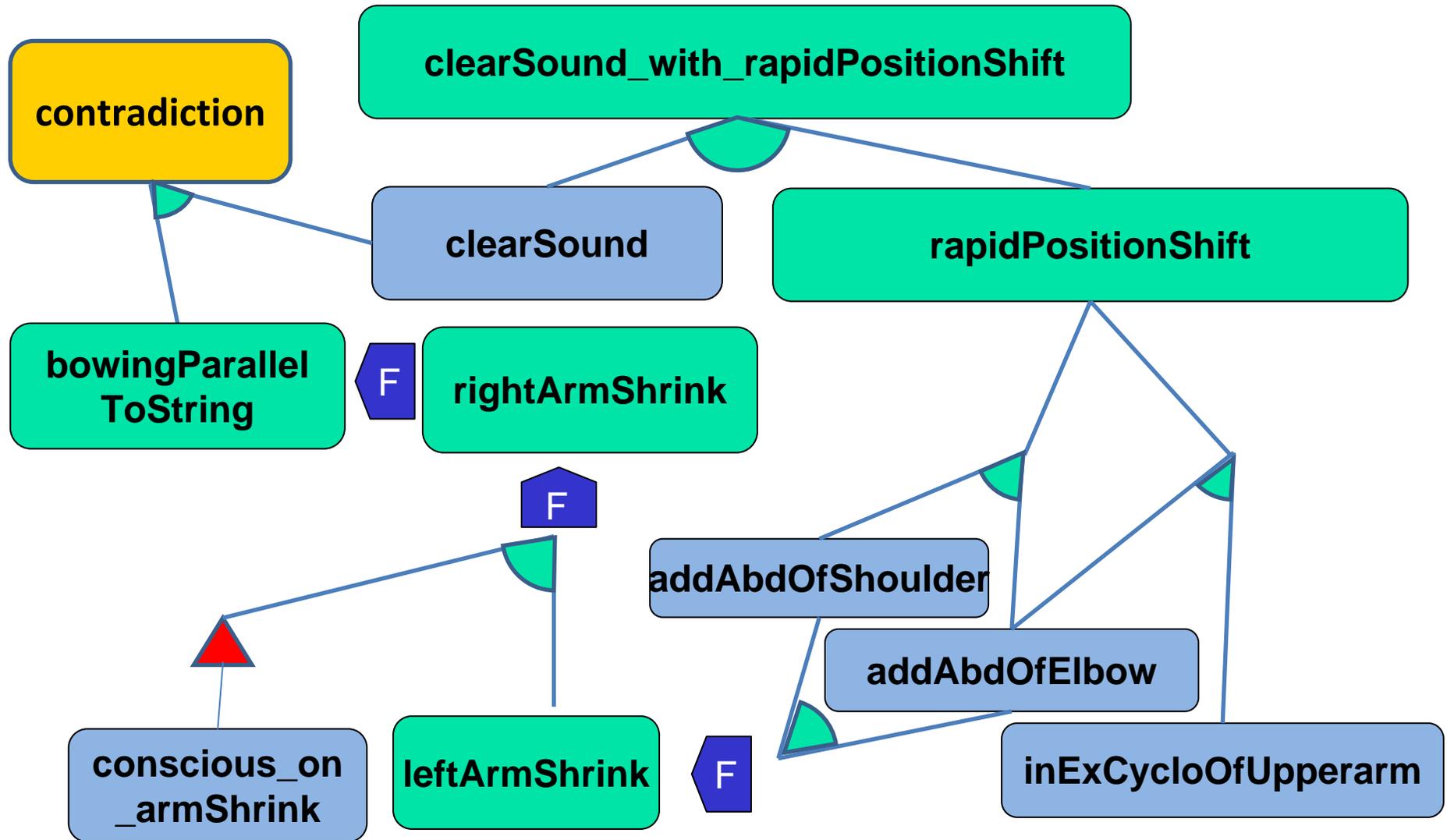
modeh(\*, addAbdOfShoulder).

modeh(\*, addAbdOfElbow).

modeh(\*, inExCycloOfUpperarm).

modeh(\*, conscious\_on\_armShrink).

# Scheme of program



# 実行結果

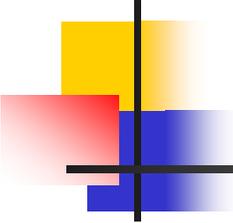
demo([clearSound\_with\_rapidPositionShift],[[],  
D]).

D = [addAbdOfElbow, inExCycloOfUpperarm,  
clearSound] ? ;

D = [addAbdOfElbow, addAbdOfShoulder,  
conscious\_on\_armShrink, clearSound] ? ;

D = [addAbdOfElbow, inExCycloOfUpperarm,  
conscious\_on\_armShrink, clearSound] ? ;

no



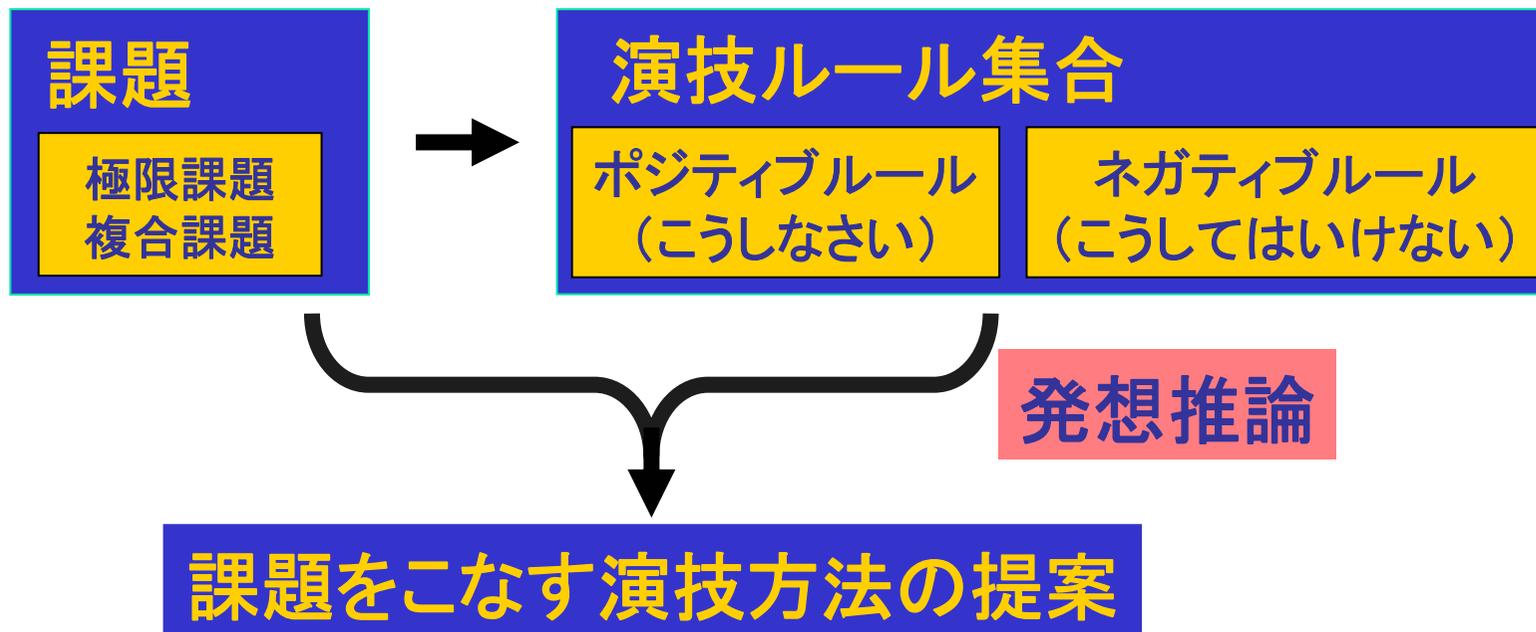
## 診断結果の解釈

---

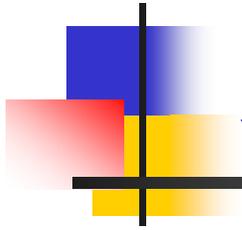
1. D = [addAbdOfElbow, addAbdOfShoulder, conscious\_on\_armShrink, clearSound]
  - 両腕が同時に縮んでしまうことに注意を払う (予想した結果).
2. D = [addAbdOfElbow, inExCycloOfUpperarm, clearSound]
  - 高速ポジションシフトを行うのに、左腕の収縮を行わない。その代わりに、前腕の肘の周りの回転によって高速ポジションシフトを行う。(予想しなかった結果)

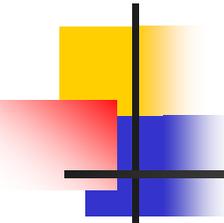
# スキル創造支援システムの課題

- 演技ルール集合をどのように作るのか。  
⇒ データマイニングによる演技ルールの自動抽出
- 候補仮説集合の自動構築
- 課題の選択



# データマイニングによる演技 ルールの自動抽出



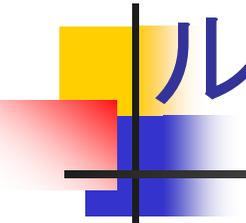


# 筋電データからのポジティブ ルールの抽出(金城)

---

- 筋肉ごとに筋活動の変化をセグメンテーション、クラスタリングにより抽出する。
- クラスタリングの結果に記号を割り当て、記号化された系列データに対して、データマイニング手法を適用する。

# 多チャンネル筋電データからの ルールの抽出



- 各EMGデータを解析し、セグメンテーション、クラスタリングを行って、筋電の活動パターンを抽出する。
- 得られた各クラスターに名前を与え、元の時系列を名前つき区間データ列に変換する。
- その結果、およびAllenの時区間論理をILPに与えて、帰納推論を実行し、活動パターン間に成り立つ関係性を抽出する。

# Extraction of relational rules by Inductive Logic Programming

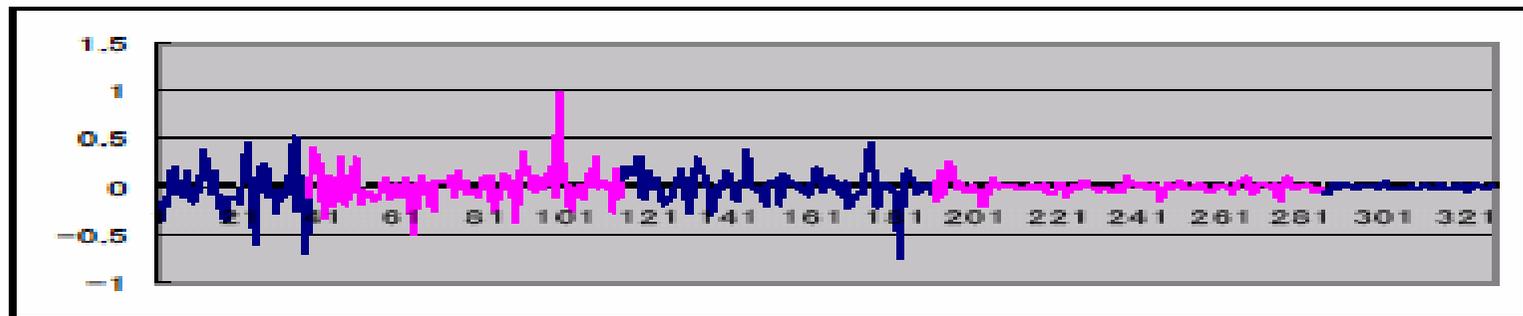
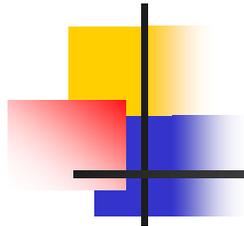
## ■ Experiments

♩ = 120

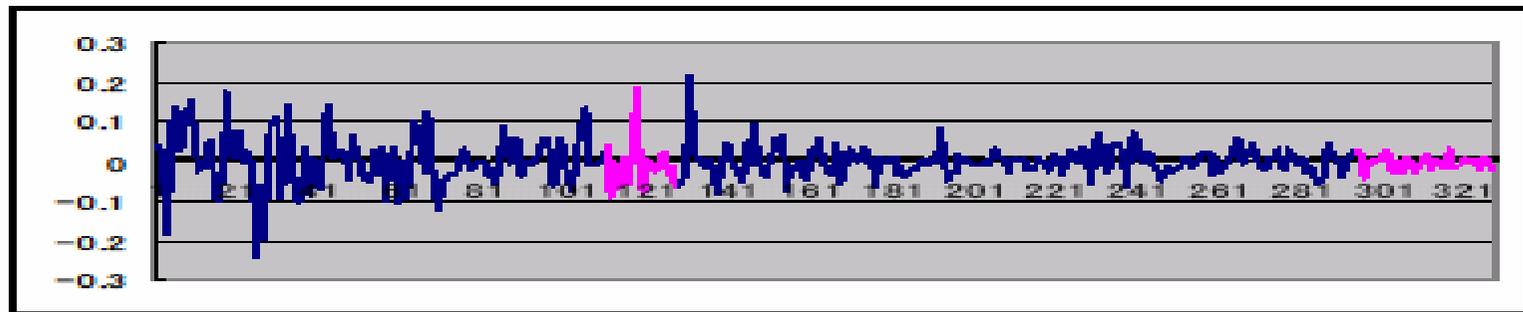
The diagram illustrates musical notation for two staves (bass and treble clef) in 3/4 time. Above the staves, red text labels bowing techniques: '下げ弓' (down-bow) and '上げ弓' (up-bow). The notation is divided into four sections by a vertical line, with red text labels below each section: '1: 全音符' (whole note), '2: 2分音符' (half note), '3: 4分音符' (quarter note), and '4: 8分音符' (eighth note). The first two sections show single notes with bowing directions, while the last two sections show rhythmic patterns of notes with bowing directions.

- Tasks: To extract rules to classify playing with different note lengths.

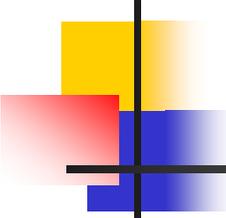
# Results of Segmentation and Clustering



前腕筋



上腕二頭筋



# Extracted Rules

---

## **Rules for 8<sup>th</sup> notes**

beat(A) :-during(A, forearm\_activation,  
triceps\_activation).

[Pos cover = 2 Neg cover = 0]

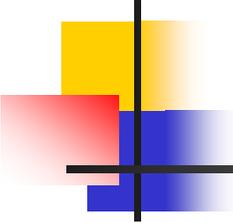
beat(A) :-overlaps(A, forearm\_activation,  
biceps\_non\_activation).

[Pos cover = 2 Neg cover = 2]

## **A rule for half notes**

beat(A):-before(A, forearm\_non\_activation,  
biceps\_non\_activation).

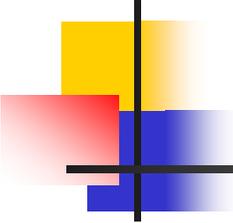
[Pos cover = 2 Neg cover = 0]



## おわりに

---

- スキルサイエンスの論理プログラミングからのアプローチを探った。
- 発想論理プログラミングが、スキル創造支援の構築に適合することを見出した。
- 帰納論理プログラミングは、生体データからのスキルルールの自動抽出に威力を発揮することを見出した。
- 私の中で、スキルサイエンスと第五世代コンピュータプロジェクトが結びついた。



# 今後の課題

---

- 諸アプローチの統合
  - データマイニングによる演技知(プロの演技ルール・動作一貫性制約)の自動獲得
  - 諸知識の統合的活用(生体力学知識、演技知、筋骨格系に関する知識)
  - メタ認知と生体力学の統合
- 支援系の構築
  - スキル創造支援
  - スキル診断支援