

# 知識の論理的表現

# コンピュータと人工知能の歴史

---

## ➤ 1970年代～1990年代

### ➤ 方法論的アプローチのフェーズ1：早期型人工知能

- コンピュータの性能が低い
- 人間の知的能力をどのような手順やアルゴリズムでコンピュータに再現させるかを研究
- 比較的表現しやすい知識(データ)を利用
- エキスパートシステム(特定分野に特化した知識ベースを元に、専門家に近い判断を下すシステム)

## ➤ 1990年代～2000年代

### ➤ 方法論的アプローチのフェーズ2：知識情報処理

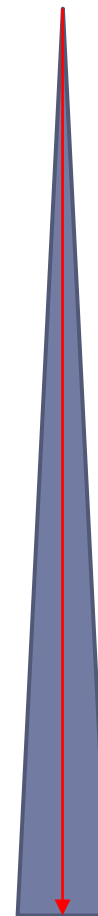
- コンピュータの性能が高まりつつある
- 様々なデータを知識として捉え、知識をもとに人間の知的能力を研究
- 様々な知識に基づく知能情報処理システム(自然言語処理など)

## ➤ 1990年代後半～

### ➤ 方法論的アプローチのフェーズ3：データ駆動型人工知能

- コンピュータの性能がかなり向上
- 大量のデータから知識を自動出すデータ駆動型の研究が盛んに
- 機械学習、ビッグデータ、深層学習

知識の重要性



# 講義計画

1 ガイダンス(授業のテーマや到達目標及び授業の方法について説明する)

**2 論理型知識の表現1(意味ネットワーク・フレーム・ルール)**

3 論理型知識の表現2(述語論理)

4 論理型のプログラミング言語Prolog入門

5 論理型のプログラミング言語Prolog演習1(基礎編)

6 論理型のプログラミング言語Prolog演習2(実践編)

7 Prolog総合演習

8 機械学習1(線形回帰)

9 機械学習2(ロジスティック回帰)

10 機械学習3(単純ベイズ分類器)

11 機械学習4(決定木)

12 機械学習5(サポートベクタマシン)

13 機械学習6(深層学習)

14 第1回目から第13回目までの講義内容について質疑応答を行う

15 理解度テストと解説

論理的  
知識

機械  
学習

# 知識の論理的表現

---

- 意味ネットワーク

オブジェクト(概念・値)間の関係についての知識を表現する

- フレーム

オブジェクトを中心とする知識を表現する

- プロダクションルール

if-then型(どういう条件のもとで何をすべきか) の知識を表現する。

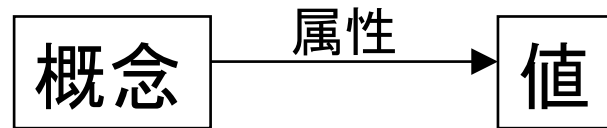
- 述語論理(次回以降)



# 意味ネットワーク

---

- 概念とその値を属性により結び付けたものである



- 概念自身を値として使うこともできる。
- 属性の**継承**

IS-A属性により、上位概念の持つ属性は基本的に下位概念にも受け継がれる。

【例】太郎 is a 人間 （太郎はその上位概念である人間の属性を継承）

【例】日大 is a 大学 （日大はその上位概念である大学の属性を継承）

【例】知識情報処理 is a 講義科目 （知識情報処理はその上位概念である講義科目の属性を継承）



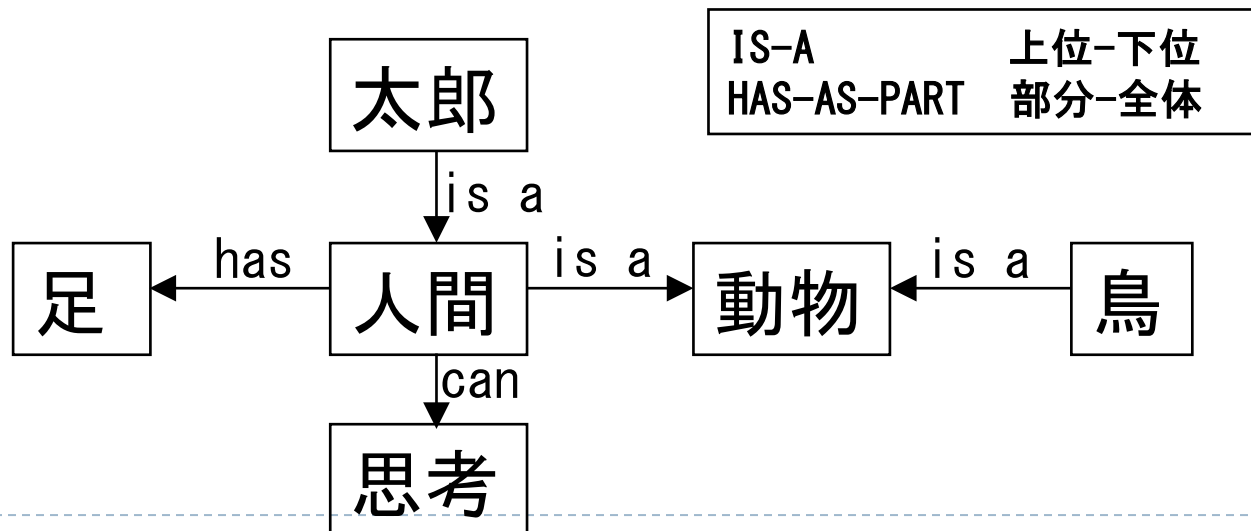
# 意味ネットワークの例

【例】以下は「人間、動物、太郎と鳥」に関する知識である  
人間は動物である。人間は思考できる。  
人間には足がある。太郎は人間である。  
鳥は動物である。

意味ネットワークに展開



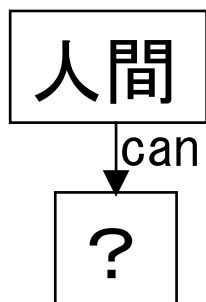
XXはXXである⇒is a 属性



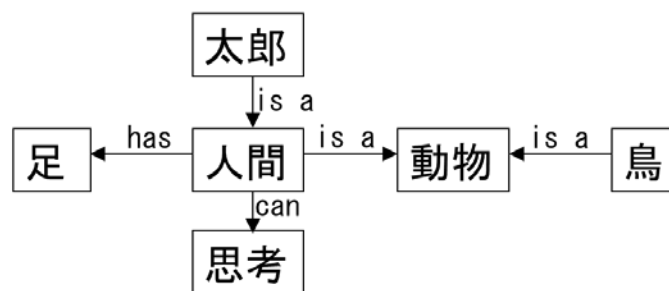
# 意味ネットワークにおける直接推論

前頁の意味ネットワークをもとに、次の質問がなされたとする。

【Q1】人間は何ができるか？

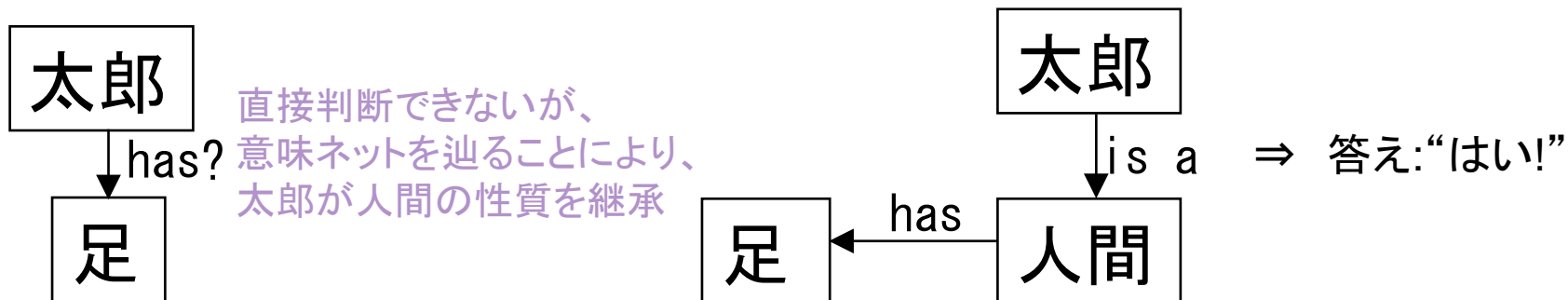


答え：？＝“思考”

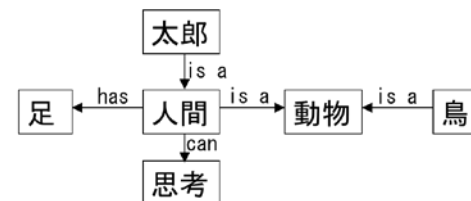
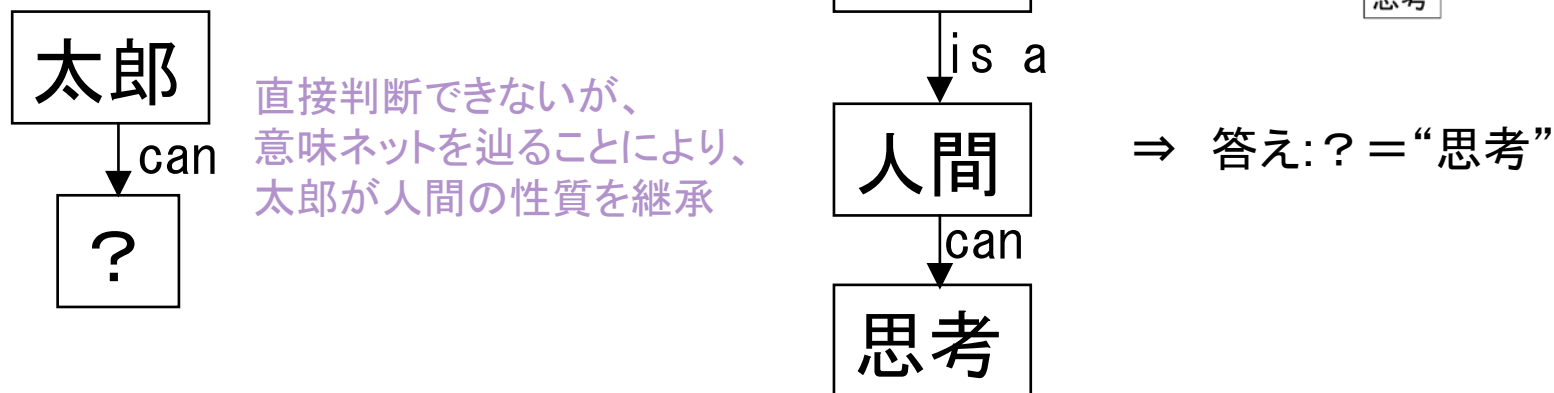


# 意味ネットワークにおける性質の継承による推論

【Q2】太郎には足があるか？



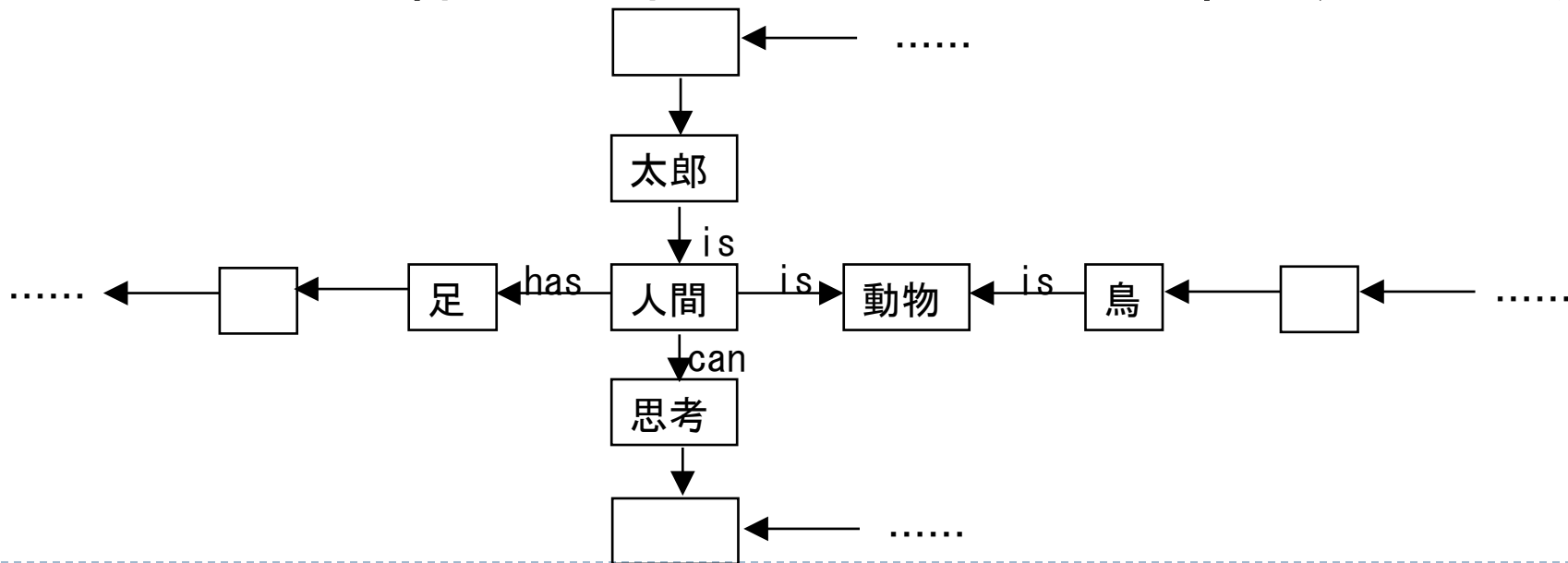
【Q3】太郎は何ができるか？





# 意味ネットワークの特徴

- 対象間関係についての知識を表現するのに最適
- 知識の継承ができ、冗長性を回避できる
- 知識が複雑になるとネットワークが大規模になる(前述の例は全体の意味ネットワークの一部にすぎない)



# フレーム

- フレームは1つ以上のスロットからなる
- 1つのスロットは1つの属性とその値の対である
- 1つのフレームが1つの知識を表す単位である

【例】 以下は「人間」に関する知識を自然言語で表現したものである。

- ① 人間には名前がある。
- ② 人間には性別がある。
- ③ 人間には生年月日があり、生年月日から年齢が計算できる。
- ④ 人間の身長は普通xxxcm前後である。

この知識をフレームで表現すれば下図のようになる。

\$REQUIRE  
: スロット  
値に関する制約  
の記述で  
ある。

人間

名前: \$REQUIRE: (<値>: [文字列])

性別: \$REQUIRE: (<値>: [文字列])

生年月日: \$REQUIRE: (<値>: [整数の3つ組])

(\$IF-INPUT: <生年月日から年齢を計算する手続き>)

年齢: \$REQUIRE: (<値>: [整数])

身長: \$REQUIRE: (<値>: [実数]) (\$DEFAULT: xxx.0)

\$IF-INPUT:  
値が定まった時  
に起動する手続  
きである。

(\$DEFAULT: デフォルト値である)

# フレームにおける推論

前頁の図のフレームに、次の入力を与えられたとする。

(F1) 太郎は人間である。

(F2) 太郎は男性で、2000年1月1生まれである。

ここで、次の質問がなされたとする。

(Q1) 太郎の年齢はいくつか？（本日は2020年10月7日とする）

(Q2) 太郎の身長はどれくらいと思われるか？

(Q3) 太郎は自宅からどうやって会社まで通勤するのか？

- ・すると、まず(F1)より、『人間フレーム』のインスタンス（実体）として『太郎フレーム』が作られる。
- ・次に、(F2)によって、『太郎フレーム』の各スロットが埋められる。
- ・結局、下図に示す『太郎フレーム』が作られる。
- ・そして、(Q1)と(Q2)の各質問に対する答えが返される。  
(Q3)は太郎のフレームのみで回答できない(他の知識が必要となる)

太郎

名前： 太郎

性別： 男性

生年月日： 2000, 1, 1

年齢： 20

身長： xxx. 0

生年月日が入  
力されると、  
現在の日付を  
もとに年齢の  
計算が始まる

# フレームの特徴

---

- オブジェクトを中心とする知識表現の手法
- ある概念に関するすべての事実と属性は一箇所に置かれ、データベース内を手間をかけて検索する必要がない（意味ネットワークではネットワーク全体に渡っての検索が必要）。
- 「オブジェクト指向」の考え方に類似
- フレームが互いに結合してフレームのネットワークを構成しないと実用性が低い（意味ネットワークと同じ）



# プロダクションシステム

---

## ➤ プロダクションルール(production rule)

「if **条件** then **行動**」というif-then型(どういう条件のもとで何をすべきか) の知識表現単位。

## ➤ プロダクションシステム (production system)

プロダクション・ルールとその解釈手続きから構成される問題解決システムであり、3つの基本的要素と3種類の動作から構成される。

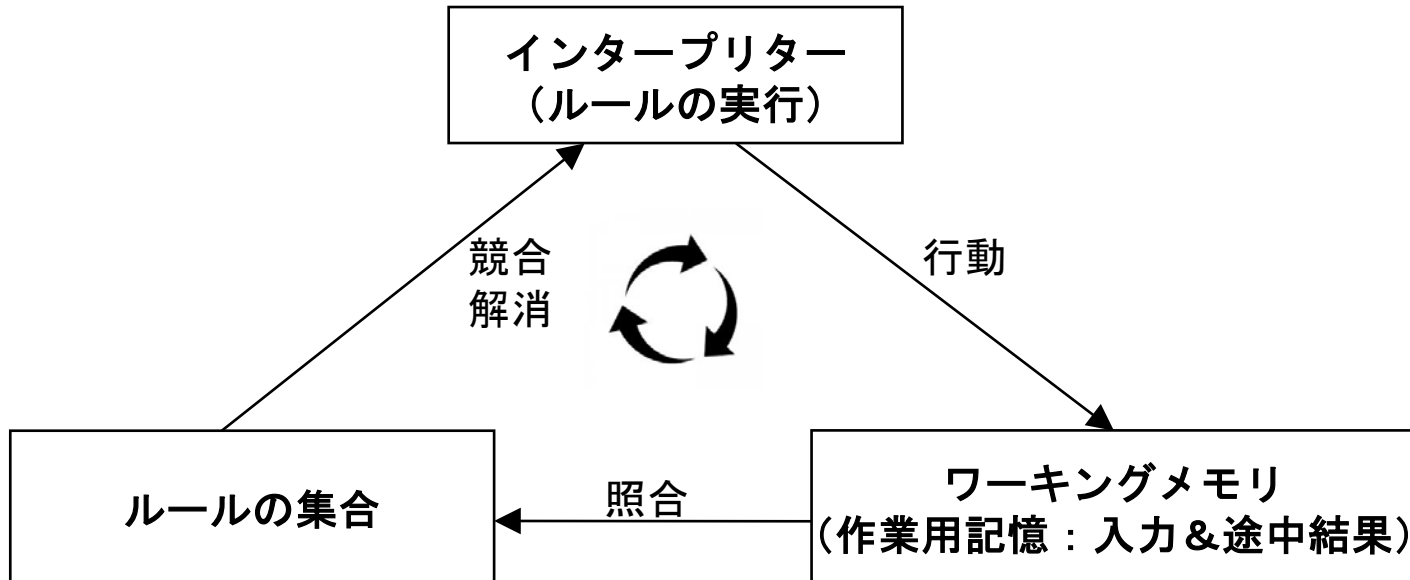
基本的要素

- インタープリター
- ルール集合
- ワーキングメモリ

動作

- 照合
- 競合解消
- 行動

# プロダクションシステムの3要素



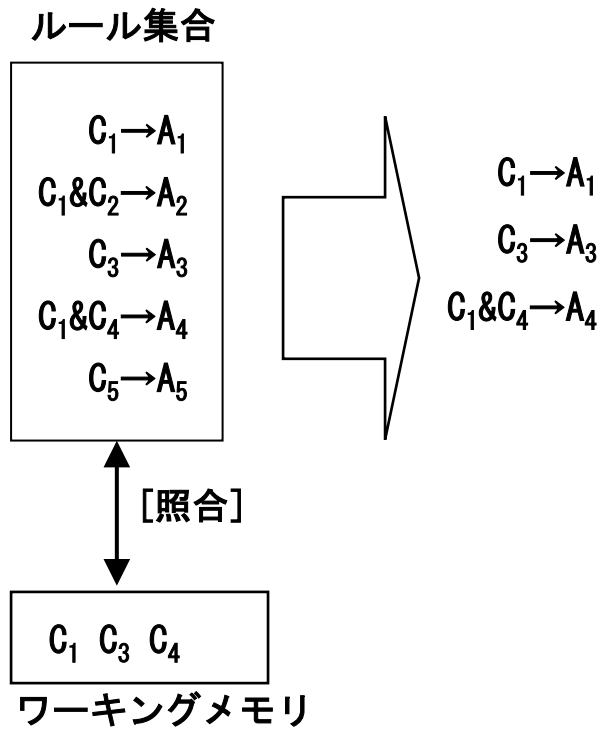
**照 合：** ルールの集合にある各ルールの条件部をワーキングメモリから送られてくる入力か途中結果と照合し、条件が満たされたルールを選び出す。選び出されたルールの集合を競合集合 (conflict set) という。

**競合解消：** 競合集合の中から実際に実行するルールを1つだけ選択する。

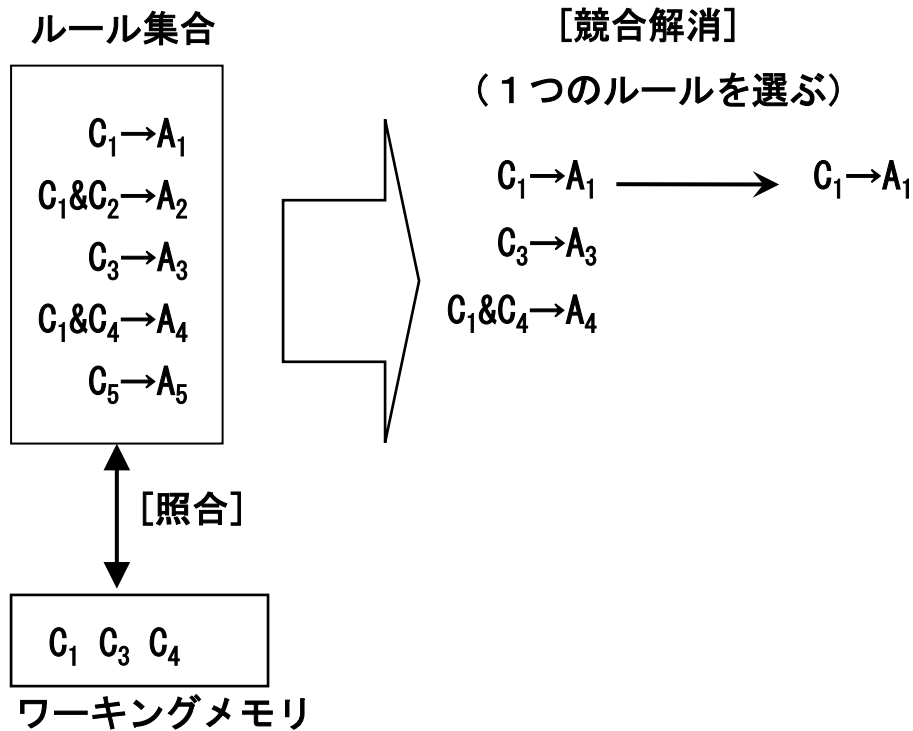
**行 動：** 選択されたルールの行動部を実行することによりワーキングメモリを更新する。

# 認識一行動サイクル

---

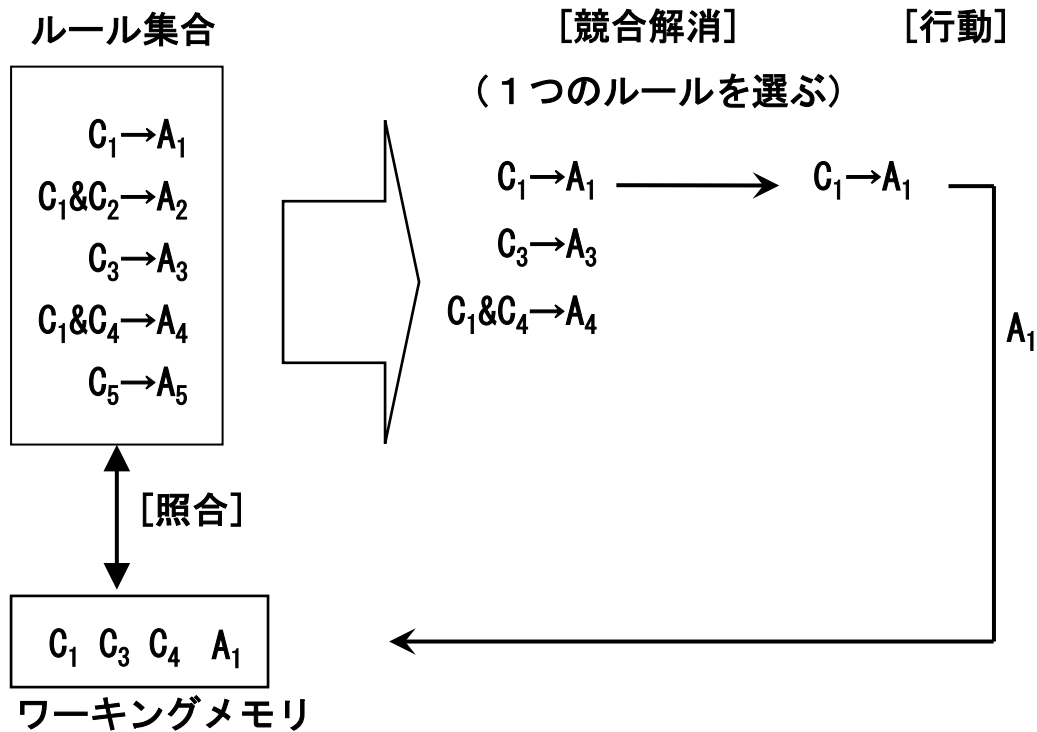


# 認識一行動サイクル





# 認識一行動サイクル



# プロダクションシステムの競合解消の原則

---

- ①一度選択したルールを再び選択しない(同じ処理が無限に繰り返されるのを防ぐ)。
- ②なるべく新しく追加された作業記憶の要素(途中結果)と照合したルールを選択する。
- ③できるだけ条件部の記述が最も詳細なルールを選択する。

※上記3つの原則を必ずしも同時に適用する必要がない



# プロダクションシステムの例

---

【例】ある授業科目の試験に落ちた学生に、その科目を再履修する必要があるかどうか（必修か選択かによる）を、指示するプロダクション・システムを考える。

## ■プロダクション・ルール

P1: if (学生:1年生) then (教養課程基礎コース)

P2: if (教養課程基礎コース)

then (必修科目:数学) & (必修科目:英語) & (選択科目:物理)

P3: if (<科目>の試験:合格) then (<科目>の履修:既済)

P4: if (<科目>の試験:不合格) then (<科目>の履修:未済)

P5: if (<科目>の履修:未済) & (必修科目:<科目>)

then (<科目>の再履修:必要)

P6: if (<科目>の履修:未済) & (選択科目:<科目>)

then (<科目>の再履修:不要)

## ■作業用記憶の初期状態

1 (学生:1年生)

2 (数学の試験:不合格)

## ■認識－行動サイクル

- ・競合解消戦略は前記の原則①、②、③。
- ・作業用記憶の要素（入力か途中結果）の数字が大きいほど最近書き込まれたことになる。
- ・作業用記憶に新たに追加される要素がなければ終了となる。

