

# 第3章 ルールの選択方法

3.1 競合解消

3.2 OPS5

3.3 ルールの記述の考え方

# プロダクションシステムの構造



# 前向き推論（認識行動サイクル）

## Step 1 照合(matching)

作業領域内のデータに満足される条件部をもつルール<sub>の</sub>集合(競合集合)を求める。

## Step 2 競合解消(conflict resolution)

競合集合の中から選択基準に従って、特定のルールを選択する。

## Step 3 実行(action)

選択されたルールの結論部を実行し、作業領域の内容を更新する。

# 競合解消戦略

- 特異性による順序づけ
- ルールによる順序づけ
- データによる順序づけ
- 条件部の大きさによる順序づけ
- 発火記録による順序づけ
- 文脈による制限

# LEX戦略 (Lexicographic sort)

- (1) 既に行われた組を競合集合から削除する。
- (2) より新しいデータをもつ組を選択する。
- (3) ルール条件部の詳細度が大きい組を選択する。
- (4) 任意の組を選択する。

# OPSS5

プロダクションシステムの基本概念の  
高速化に成功したシステムである。



# 知識ベース

## ルール表現

( p <ルール名> <条件部> --> <結論部> )

[例] 体毛をもつ動物はほ乳類である。

( p ルール 1

( 動物 ^ もつ 体毛 )

-->

( make 動物 ^ 種類 ほ乳類 ) )





## ルールの例

countの値を1つ増やす。

(p ルール2

(count ^値 <x>)

-->

(modify 1 count ^値 (compute <x> + 1)))

# ルールの例 ( 1 )

(p rule1

(someone ^situation hungry)

(apple ^size 10)

(eat ^number <count>)

-->

(remove 2)

(modify 3 ^number (compute <count> + 1)))

## ルールの例 ( 2 )

(p rule2

(someone ^situation hungry)

(apple ^size 10)

-->

(make eat ^number 0))

(p rule3

(apple ^color black)

-->

(remove 1))

# 例 1

## 作業領域

- 1 (apple ^color red ^size 10)
- 2 (apple ^color blue ^size 10)
- 3 (someone ^situation hungry)

# 例 1 の動作

各サイクル毎の

競合集合

選択されたルール

ルール実行後の作業領域

は表3.1（教科書p.32）を参照

## 例 2

### 作業領域

- 1 (apple ^color red ^size 10)
- 2 (apple ^color black ^size 10)
- 3 (someone ^situation hungry)

## 例 2 の動作

各サイクル毎の  
競合集合  
選択されたルール  
ルール実行後の作業領域  
は表3.2（教科書p.36）を参照



# 例1と例2の結果が同じ になった理由

例2では黒色りんごが食べられてしまった！

競合解消戦略LEX戦略の

「(3)ルール条件部の詳細度が大きい組を  
選択する。」

により, rule3が選択されなかった。

# 黒色りんごを食べないようにする 方法

1. rule1の条件に食べることの出来る  
りんごの色を記述する。  
食べることの出来る色をすべて  
記述する必要がある。
2. 作業過程を考慮に入れる。

## 作業過程を考慮に入れた場合

(operation ^step check\_apple)

食べられないりんごをチェックする過程

(operation ^step have\_apple)

りんごを食べる過程

# 作業過程を考慮したルール ( 1 )

(p rule1

(operation ^step have\_apple)

(someone ^situation hungry)

(apple ^size 10)

(eat ^number <count>)

-->

(remove 2)

(modify 3 ^number (compute <count> + 1)))

# 作業過程を考慮したルール ( 2 )

(p rule2

(operation ^step have\_apple)

(someone ^situation hungry)

(apple ^size 10)

-->

(make eat ^number 0))

(p rule3

(operation ^step check\_apple)

(apple ^color black)

-->

(remove 1))

# 作業過程を考慮したルール ( 3 )

(p rule4

(operation ^step check\_apple)

-->

(modify 1 ^step have\_apple))

(operation ^step check\_apple)

を

(operation ^step have\_apple))

に書き換える .

**作業過程の切替**

# 例 3

## 作業領域

- 1 (apple ^color red ^size 10)
- 2 (apple ^color black ^size 10)
- 3 (someone ^situation hungry)
- 4 (operation ^step check\_apple)

## 例3の動作

各サイクル毎の

競合集合

選択されたルール

ルール実行後の作業領域

は演習問題3.4の解答（教科書p.130）

を参照