

第8章 差異の解析による学習

8.1 問題の表現方法

8.2 帰納的発見的手法

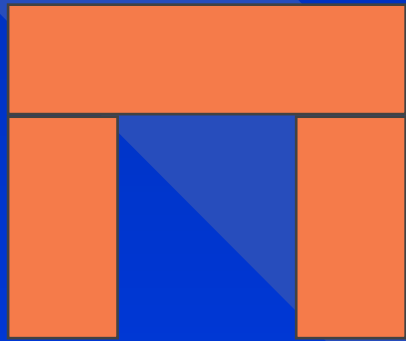
8.3 学習例

差異の解析による学習

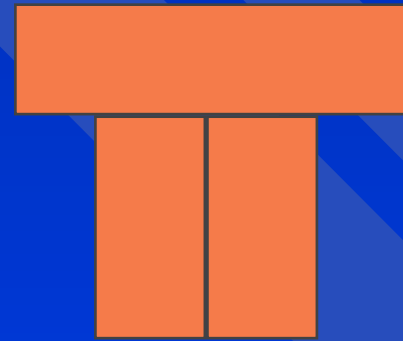
正例と負例 (nearmiss) を
与えることにより概念を
学習する。

与える正例と負例

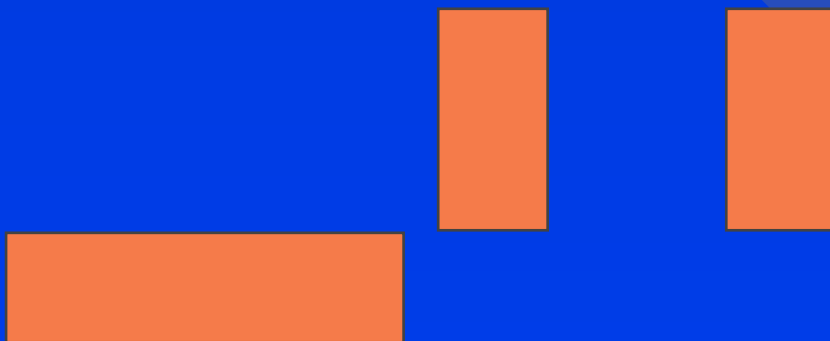
1 . Arch



3 . Near miss



2 . Near miss



4 . Arch

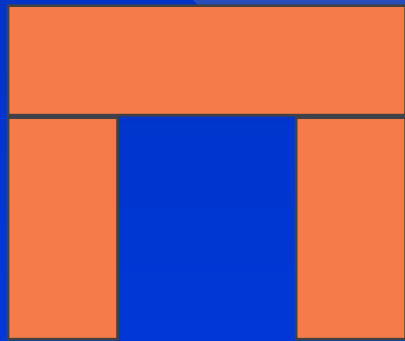


8.1 問題の表現方法

- 部品表現
- 部品と部品の関係
- 分類の階層

図形の表現例（部品）

1 . Arch



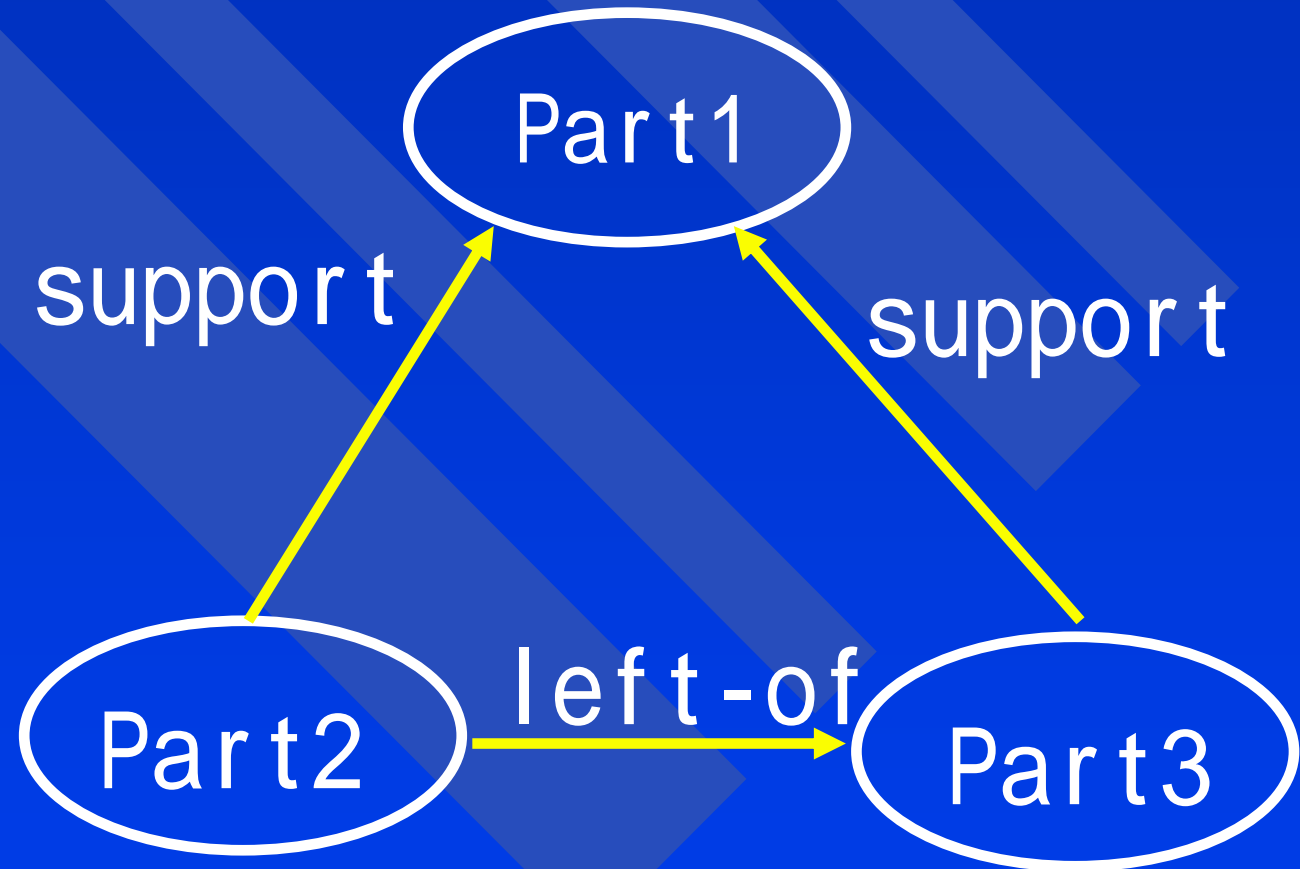
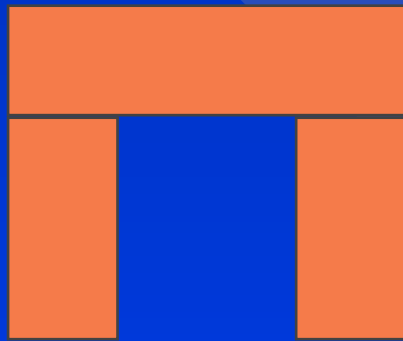
Part1

Part2

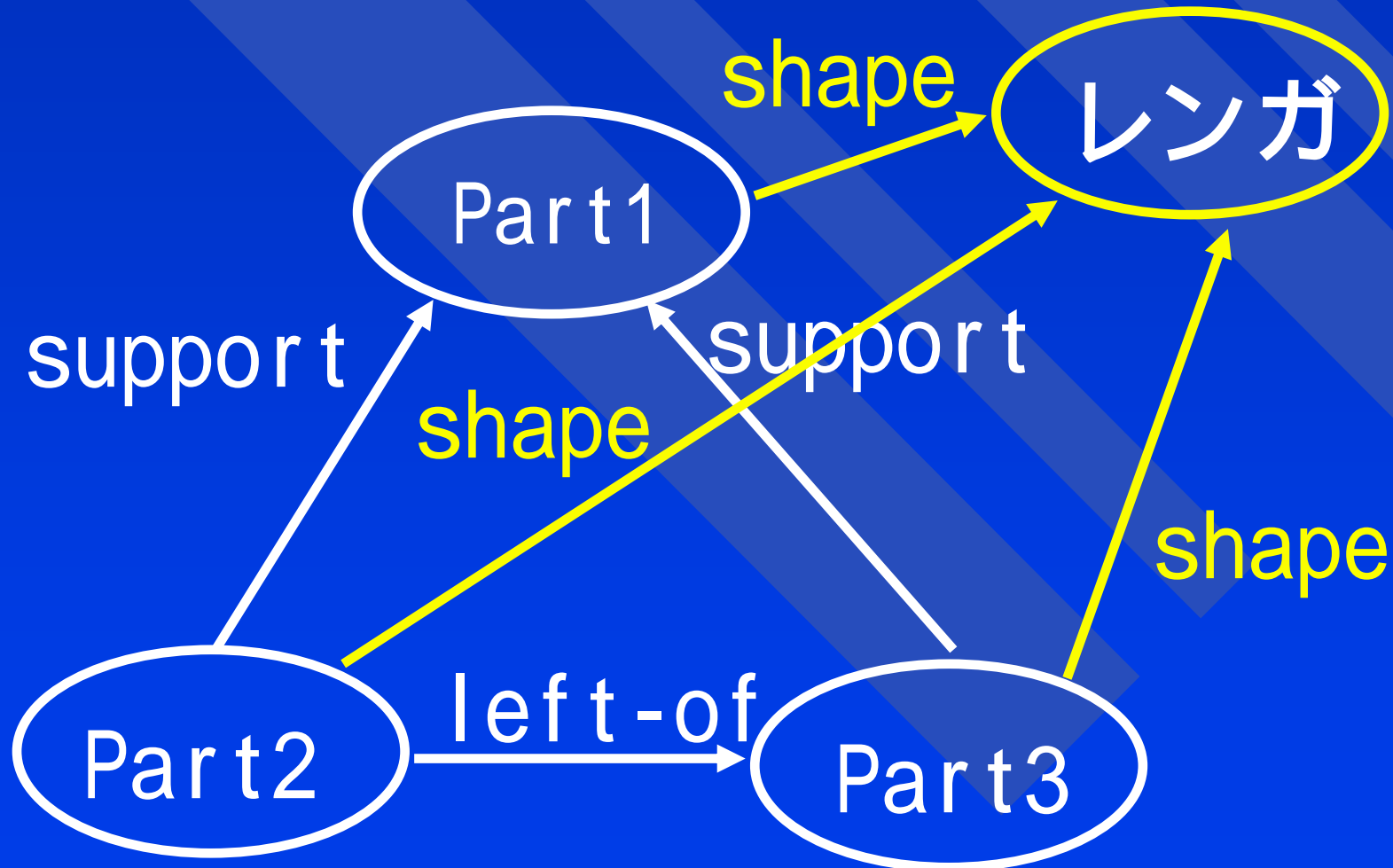
Part3

図形の表現例 (部品と部品の関係)

1 . Arch



図形の表現例（部品の種類）



フレームによる表現

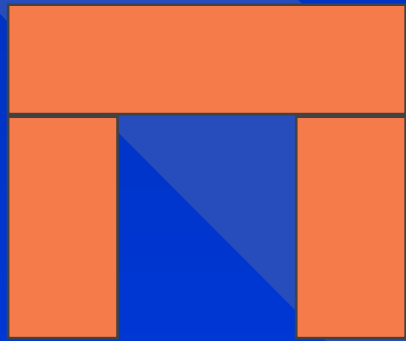
Part 1
shape
値 : レンガ

Part 2
shape
値 : レンガ
support
値 : part 1
left-of
値 : part 3

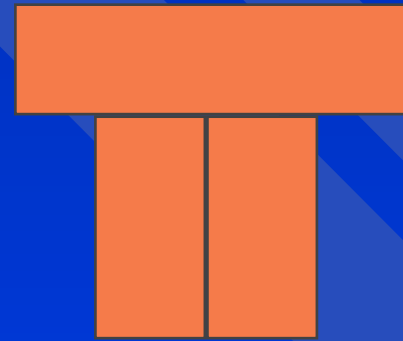
Part 3
shape
値 : レンガ
support
値 : part 1

与える正例と負例

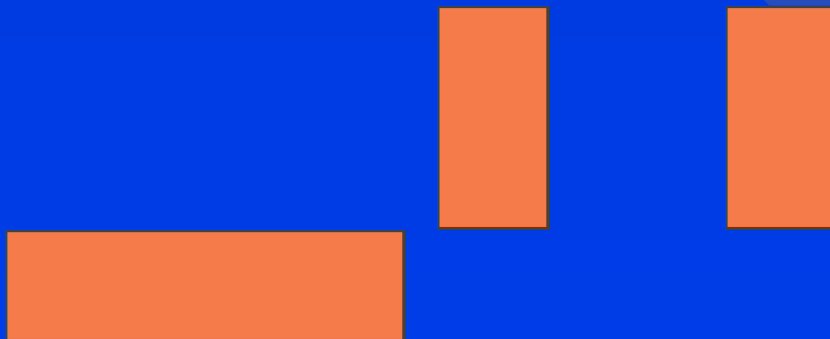
1 . Arch



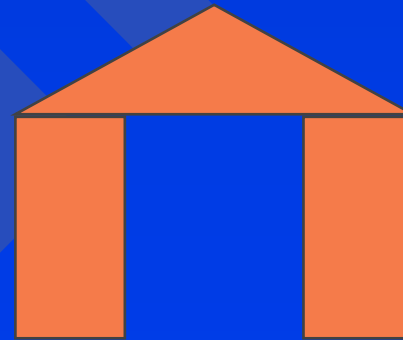
3 . Near miss



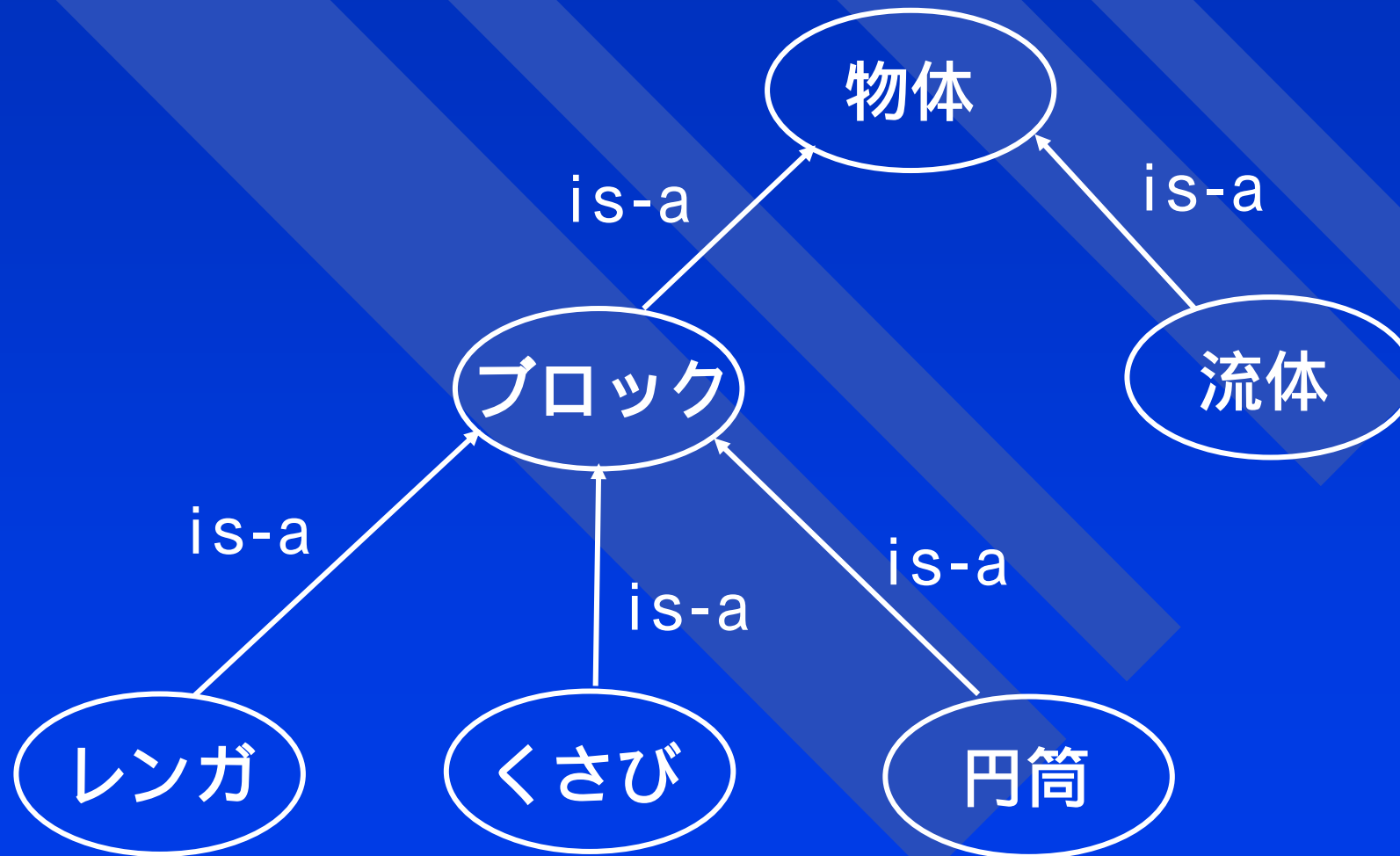
2 . Near miss



4 . Arch



形状に関するクラス階層



8.2 帰納的発見的手法

物体モデルを学習する手続きW

- 1 . 最初の例を記述の初期状態とする .
例は正例である必要がある .
- 2 . 2番目以降の例に対して以下の処理を行う .
 - 2.1 ニアミスであれば , 特殊化手続き
 - 2.2 正例であれば , 一般化手続き

特殊化手続き

1. モデルとニアミスの各部分の対応
2. モデルとニアミスとの間の差異について
 - 2.1 重要な差異が存在すれば,
 - (1) モデルに存在し、ニアミスに存在しない枝が存在すれば、**require-link**処理
 - (2) ニアミスに存在しモデルに存在しない枝が存在すれば、**forbid-link**処理
 - 2.2 重要な差異が存在しなければ、ニアミスは無視

一般化手続き

- 1 . モデルと正例の各部分の対応
- 2.1 クラス階層について
 - (1) 両者のクラスがクラス階層の一部であれば **climb-tree**処理
 - (2) 両者のクラスがすべての要素を示しているなら **drop-link**処理
 - (3) 上記以外るとき **enlarge-set**処理
- 2.2 正例にクラスがなければ **drop-link**処理
- 2.3 差異が数値や領域 , **close-inverval**処理
- 2.4 上記以外は差を無視

処理の種類

- **require-link処理** : must条件に変更する .
- **forbid-link処理** : must-not条件付き枝を追加
- **climb-tree処理** : クラス共通の上位クラスに変更し , must条件にする .
- **enlarge-set処理** : 2つのクラスを結合した新しいクラスを作り , must条件の枝を作る .
- **drop-link処理** : 関係をモデルから削除
- **close-interval処理** : モデルの数値と例の数値で領域を作成する .

8.3 学習例

学習例は，教科書のp.87 ~
p.91を参照