第4回　論理型AI
～安全安心な人工知能を目指して～

東京理科大学理工学部情報科学科
滝本　宗宏

2023/11/22

東京理科大学　創域理工学部
情報計算科学科
INFORMATION SCIENCES

目次

構造化データ

- 非構造化データ：整理されていない生のデータ
  副次的に生成 ・・ 画像データ，音声データ，テキストなど

- 構造化データ：整理されたデータ
  深層学習（Deep Learning）や機械学習によって，非構造化データから生成 ・・ 表データ，XMLデータなど

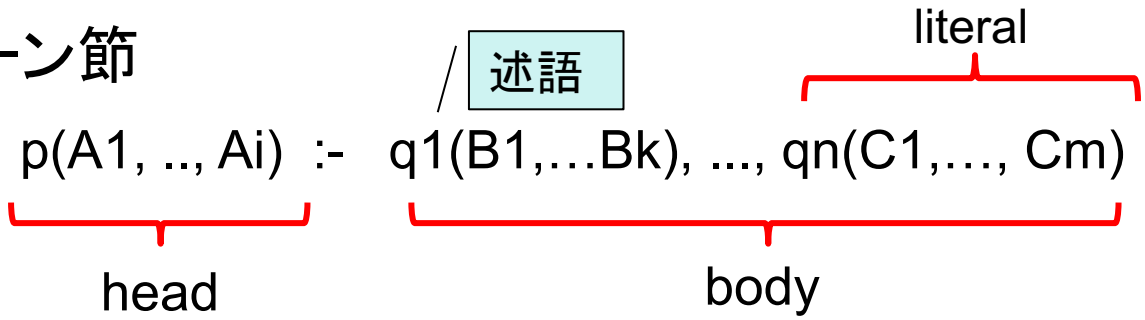
- 深層学習や機械学習の流行
  → 多くの構造化データ

- 構造化データ間の知見（関係）の重要性

帰納論理プログラミング（説明可能AI，XAI）
… XApe（XAI Parallel Engine, Progolの拡張）を例に解説

知識の表現と論理型プログラム

- ホーン節



literal**1** かつ literal**2** かつ … かつ literal**n** なら head

- ルール

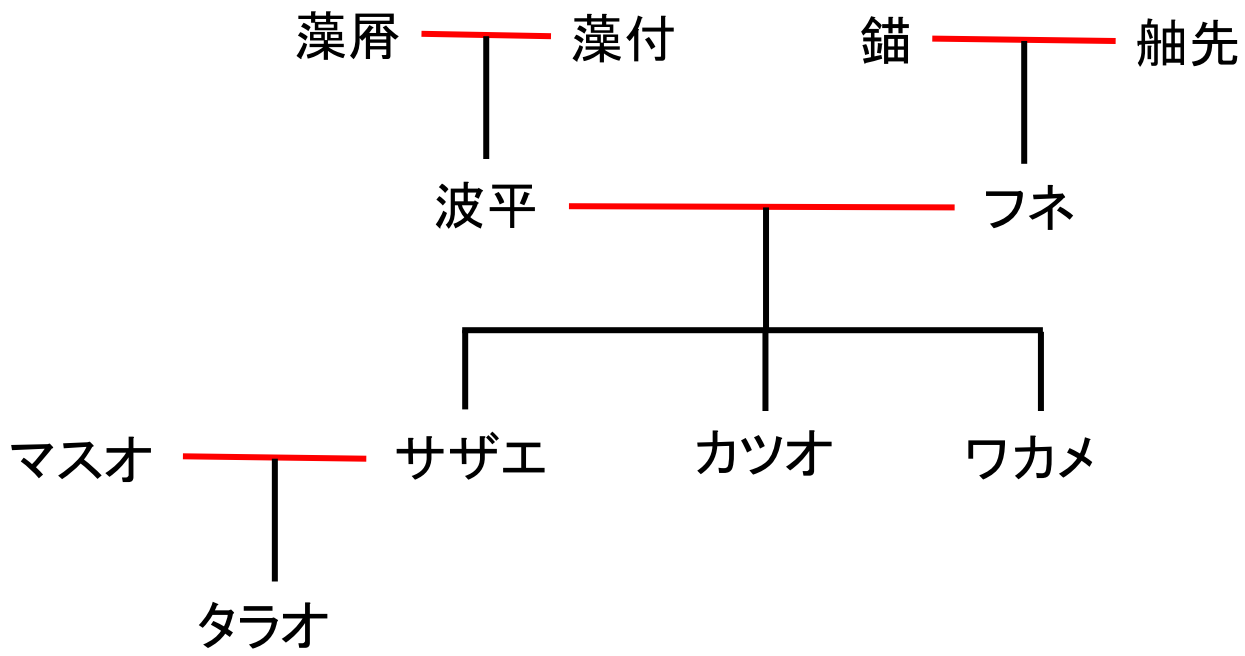
head :- literal**1**, literal**2**, … , literal**n**

- 事実（bodyがない節）

head

知識の表現と論理型プログラム

- 例：磯野家



知識の表現と論理型プログラム

事実

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

質問と実行

? parent (サザエ, タラオ)

知識の表現と論理型プログラム

事実

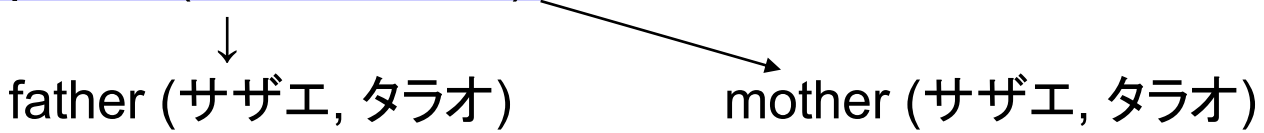
father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

質問と実行

? parent (サザエ, タラオ)



知識の表現と論理型プログラム

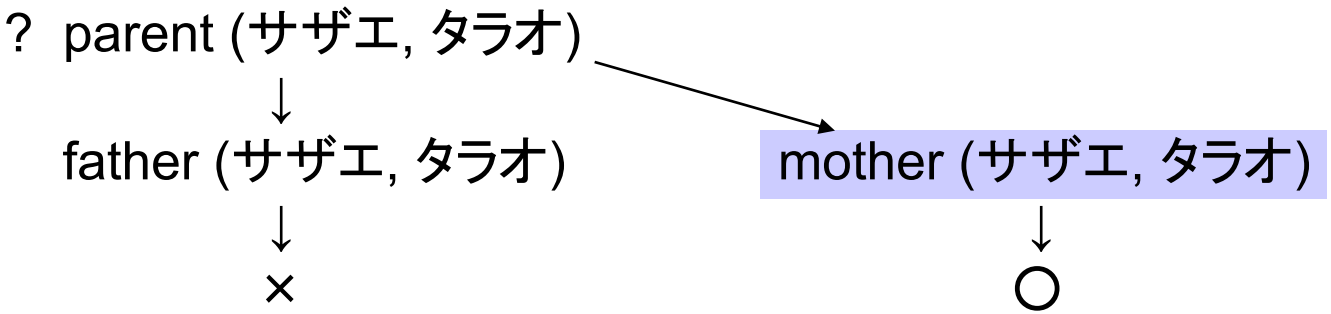
事実

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

質問と実行



知識の表現と論理型プログラム

事実

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

質問と実行

? parent (サザエ, タラオ)
Yes

知識の表現と論理型プログラム

事実

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

質問と実行（変数を含む場合）

? parent (波平, B)

知識の表現と論理型プログラム

事実

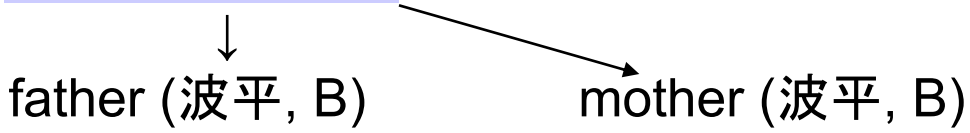
father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

質問と実行

? parent (波平, B)



知識の表現と論理型プログラム

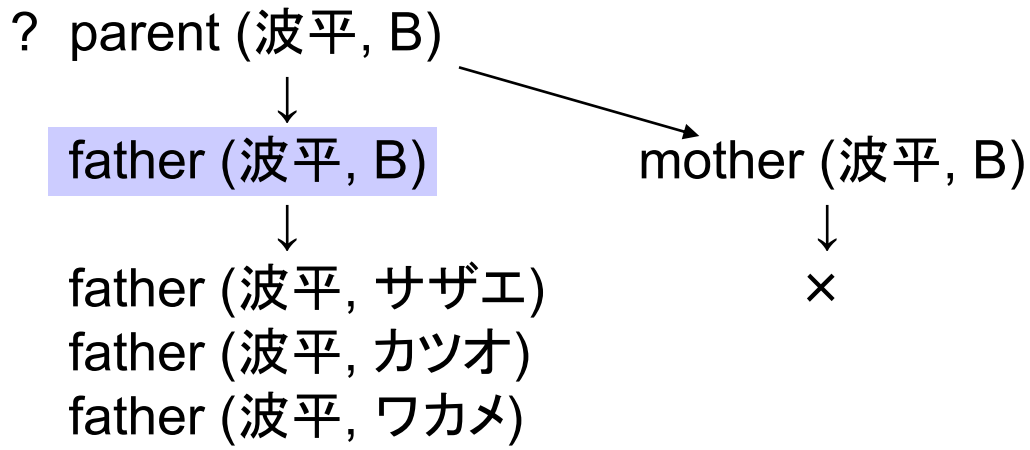
事実

father (波平, サザエ)
father (波平, カツオ)
father (波平, ワカメ)
father (藻屑, 波平)
mother (サザエ, タラオ)

ルール

```
parent (A, B)  :-  father (A, B)
parent (A, B)  :-  mother (A, B)
```

質問と実行



知識の表現と論理型プログラム

事実

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

質問と実行

? parent (波平, B)
B = サザエ，カツオ，ワカメ

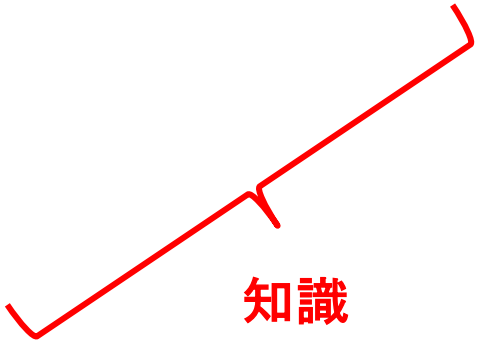
知識の表現と論理型プログラム

事実

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

ルール

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)



質問と実行

? parent (波平, B)
B = サザエ，カツオ，ワカメ

知識の表現と論理型プログラム

演習1

- 準備

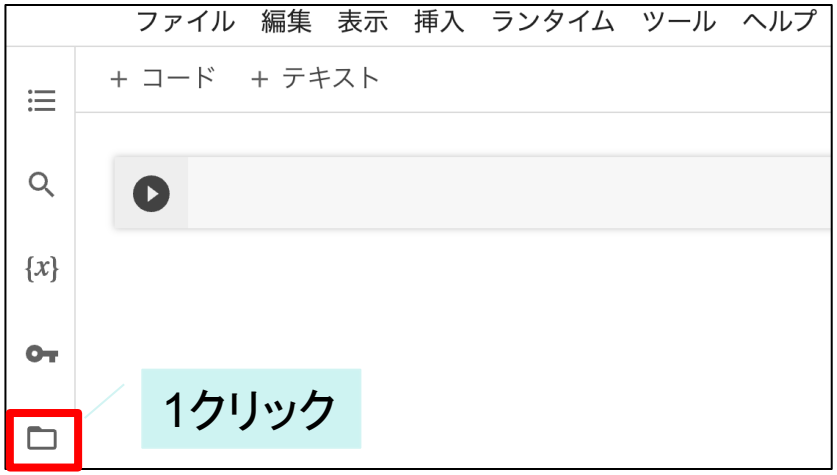
1. 第4回のページ（Colabへのリンクもあります）からファイルをダウンロード
   https://www.rs.tus.ac.jp/mune/nikkei/
   Google Colaboratory（以下，Colab）のページを開く.
   https://colab.research.google.com/
2. 「ノートブックを新規作成」する.

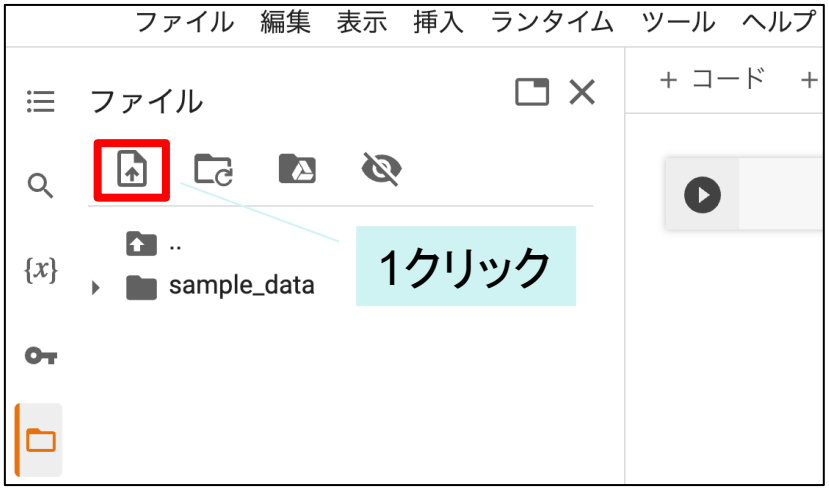
知識の表現と論理型プログラム

演習1

- ノートブックの操作

1. ファイルをアップロード





ファイル（ape.zip）を選択

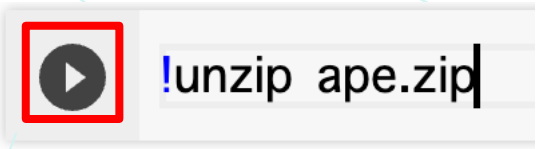
知識の表現と論理型プログラム

演習1

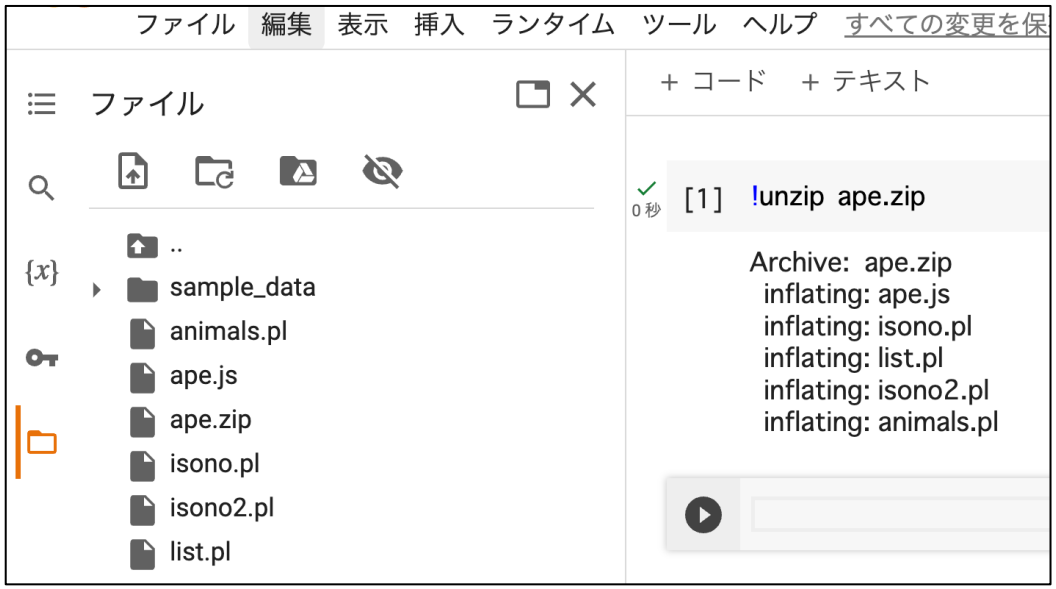
- ノートブックの操作

2. 実行とape.zipの解凍

先頭は「!」　最後は「Shift+Enter」



ここの1クリックでも実行

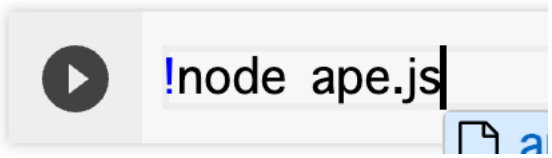


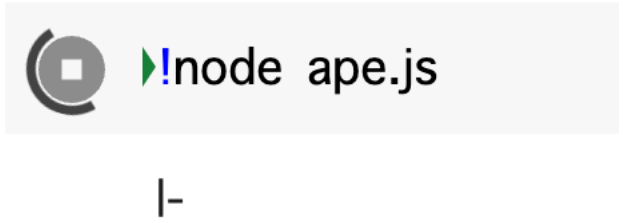
知識の表現と論理型プログラム

演習1

– ノートブックの操作

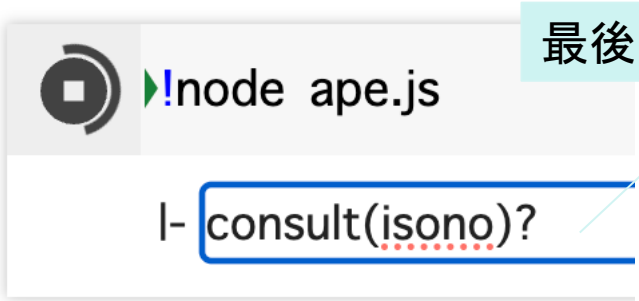
3. XApeの実行



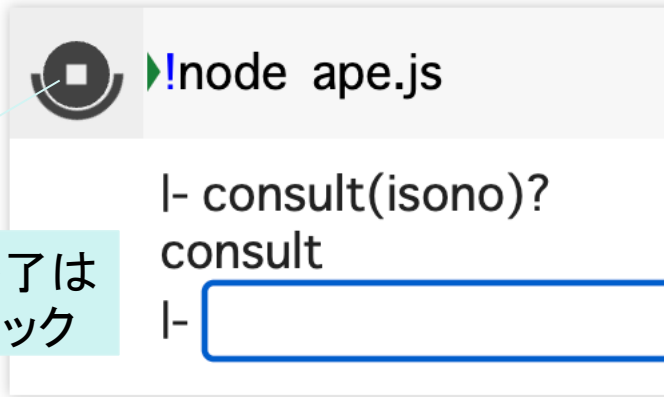


ここをクリックして，
XApeのコマンドを入力

4. XApeのコマンド入力
（ファイルisono.plの読込み）



最後は「Enter」



実行の終了は
ここをクリック

知識の表現と論理型プログラム

演習1

- ノートブックの操作

5. XApeのコマンド入力
（質問の実行）

```
!node  ape.js

|- consult(isono)?
consult
|- mother(X,tarao)?
X = sazae
Yes
```

```
!node  ape.js

|- consult(isono)?
consult
|- mother(X,tarao)?
X = sazae
Yes;
No.
|- |-
```

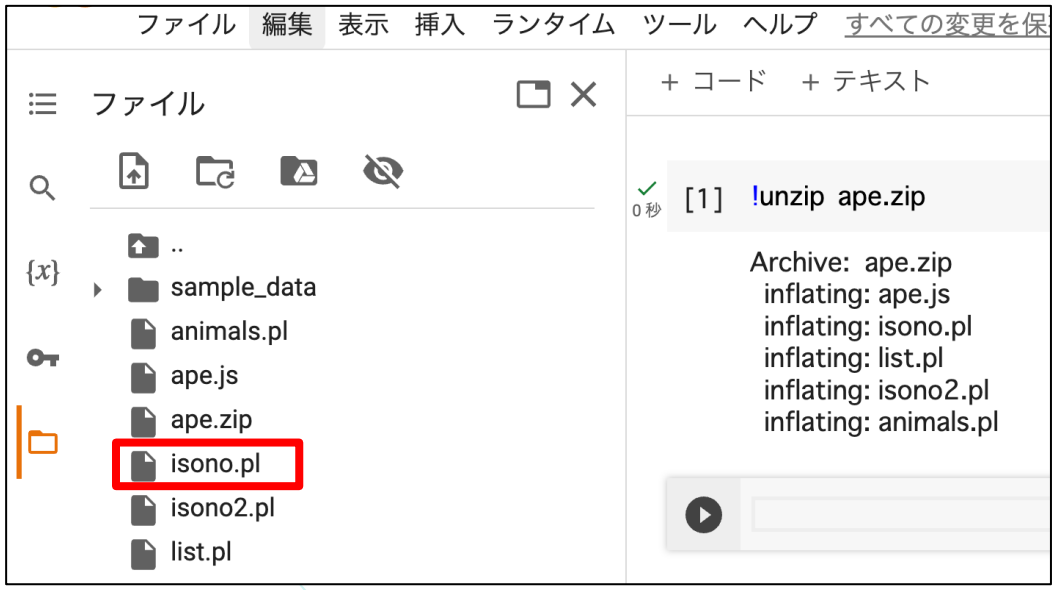
「;」を入力すると次候補を表示.
「.」を入力すると終了.

知識の表現と論理型プログラム

演習1

- ノートブックの操作

6. ファイルを開く/編集する.



ダブルクリック

注：「%」はコメントなので，適宜付けたり消したりしてください.

知識の表現と論理型プログラム

演習1

- ノートブックの操作

7. isono.plに「parent」を加えてみよう.

```
36 mother(fune, sazae).
37 mother(fune, katsuo).
38 mother(fune, wakame).
39
40 parent(X,Y) :- father(X,Y).
41 parent(X,Y) :- mother(X,Y).
```

付加

```
!node ape.js
|- consult(isono)?
consult
|- parent(X, sazae)?
X = fune
Yes;
X = namihei
Yes;
No.
|- |-
```

帰納論理プログラミング

- 背景知識

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

- 事例（grandfather)

[正例]

grandfather (波平, タラオ)

grandfather (藻屑, カツオ)

⋮

[負例]

grandfather (波平, サザエ)

grandfather (藻屑, タラオ)

⋮

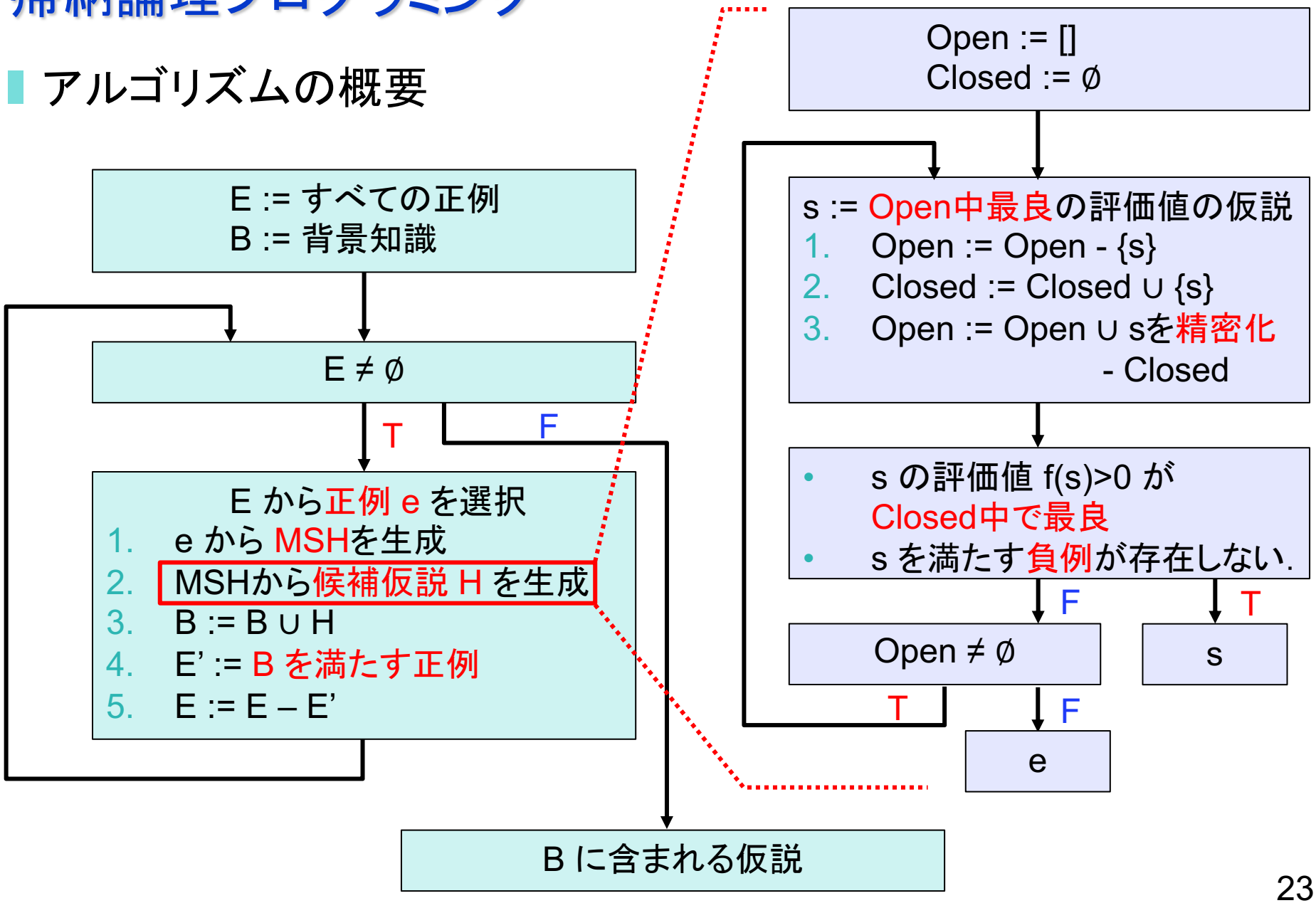
- 仮説（新しいルール）の生成

背景知識と事例のパーツの組合せ → 正例を満たし、負例を満たさないルールを生成

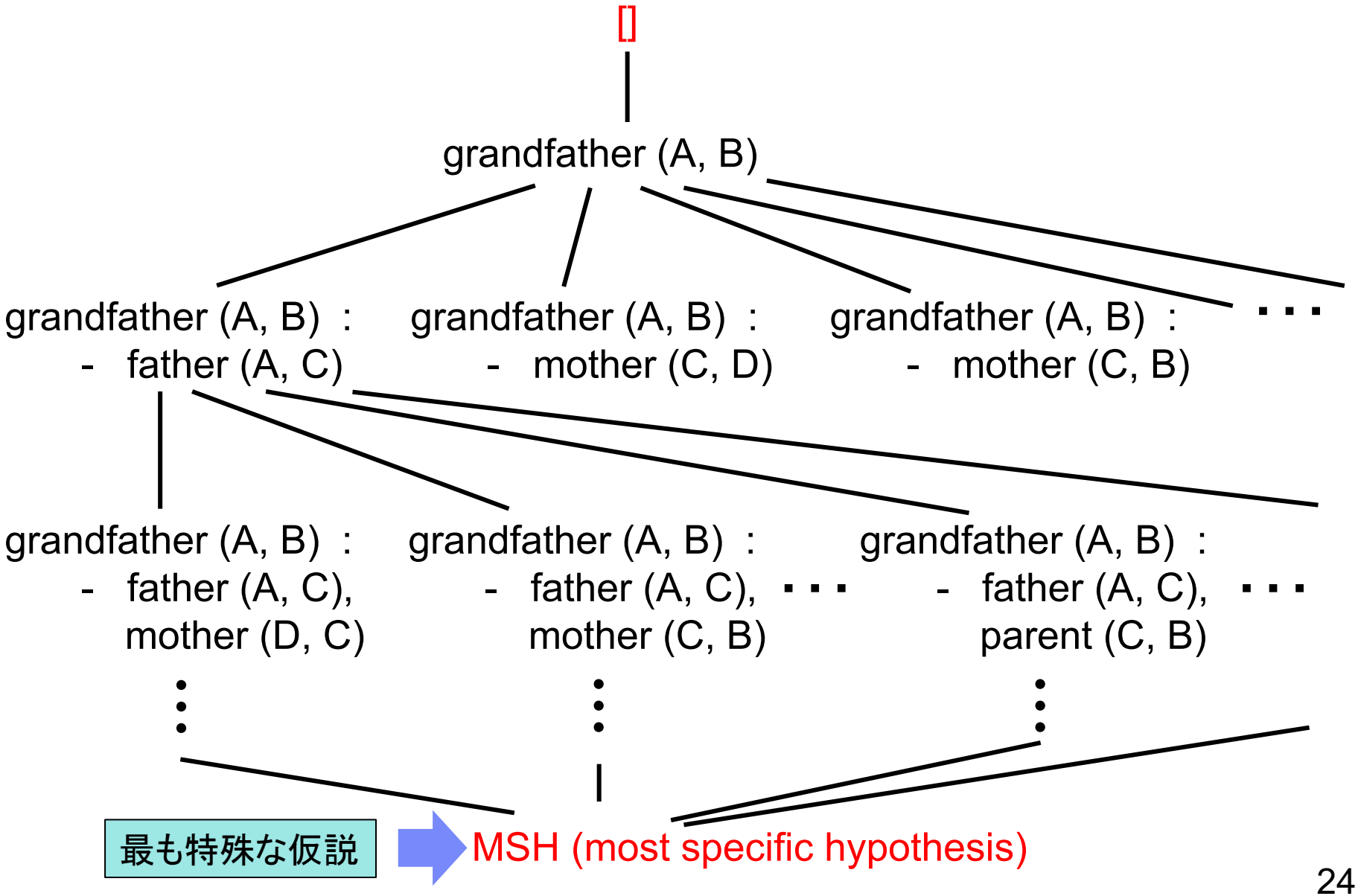
grandfather (A, B) :- father (A, C), parent (C, B)

帰納論理プログラミング

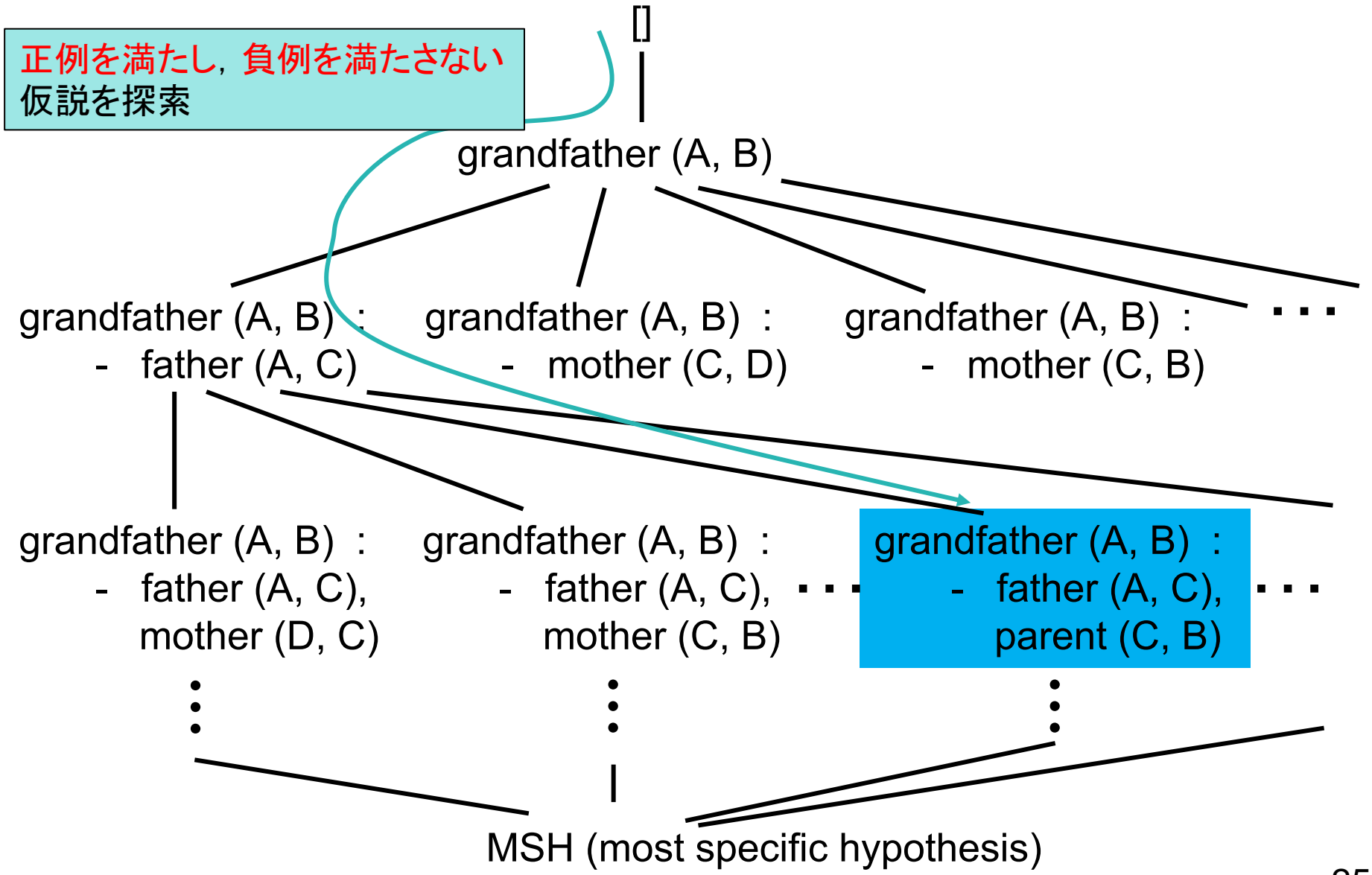
アルゴリズムの概要



帰納論理プログラミング

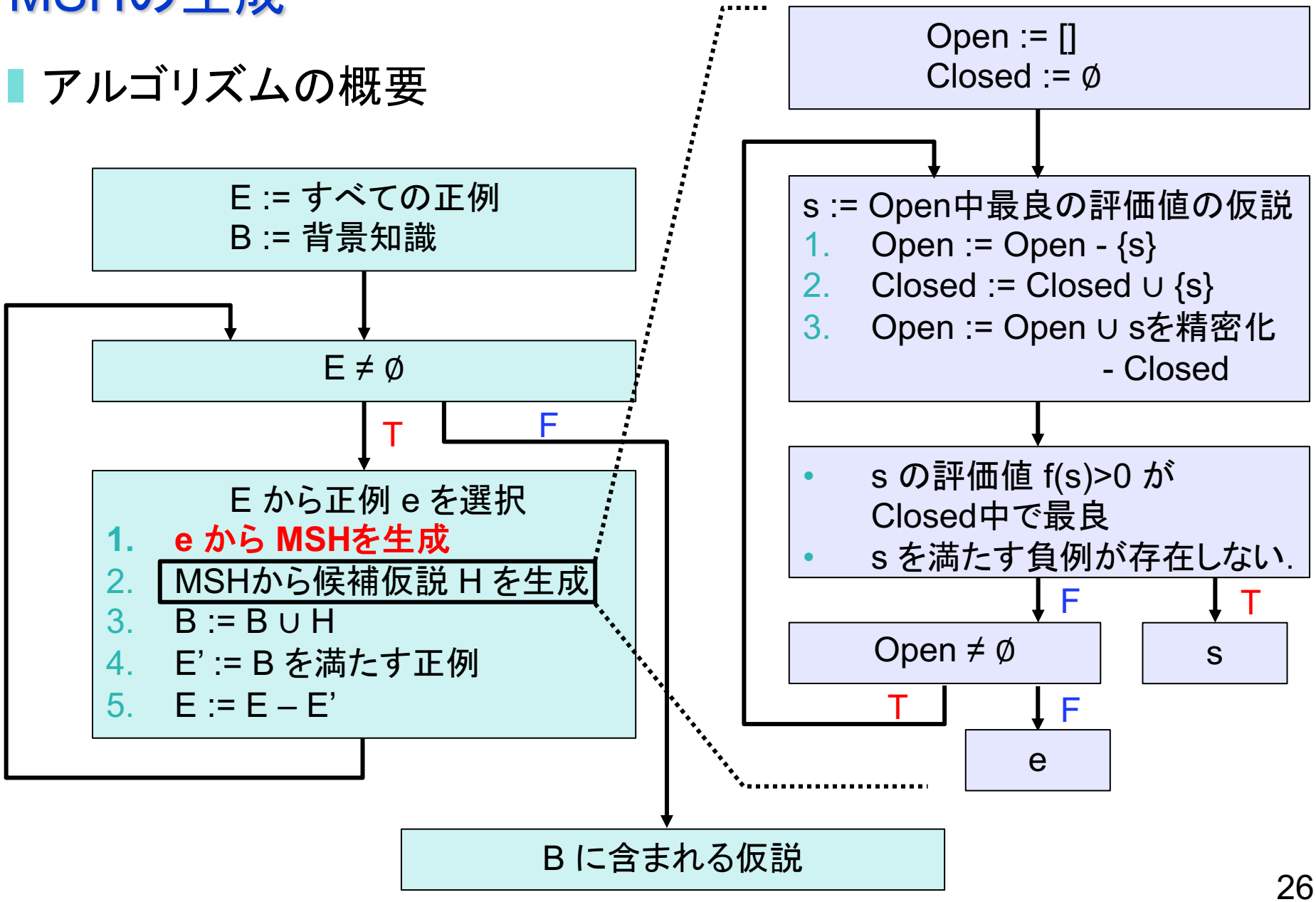


帰納論理プログラミング



MSHの生成

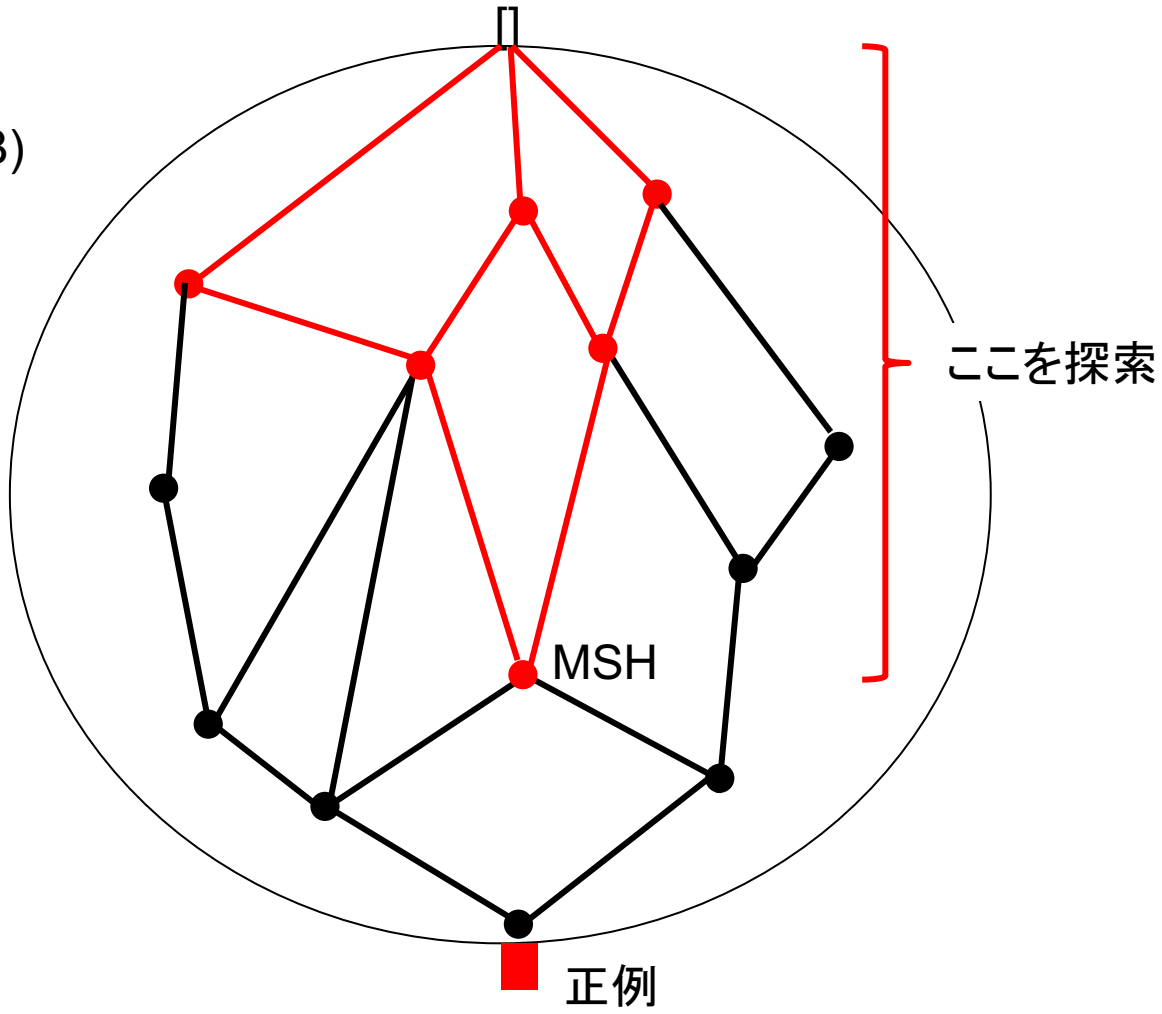
アルゴリズムの概要



MSHの生成

- MSH（Most Specific Hypothesis）：最弱仮説
  最も特殊な仮説

一般的　　特殊
p(A,B) > p(A,A)
p(A,B) > p(A,B) :- q(A,B)
p(A,B) > p(波平, B)



MSHの生成

- 各正例とモード宣言から生成
  - 例：grandfather(波平, タラオ)
    1. \+ ← 出現した定数
    2. \- ← 背景知識を満たす定数
    3. 各定数を変数で置換.

モード宣言

```
:-modeh(*, grandfather(+person,-person))?
:-modeb(*, father(+person,-person))?
:-modeb(*, mother(+person,-person))?
:-modeb(*, parent(+person,-person))?
```

+father

grandfather(波平, タラオ) :- father(波平, ワカメ),
father(波平, カツオ), father(波平, サザエ),

+mother

grandfather(波平, タラオ) :- father(波平, ワカメ),
father(波平, カツオ), father(波平, サザエ),
~~mother(波平, ？),~~ ~~mother(ワカメ, ？),~~
~~mother(カツオ, ？)~~, mother(サザエ, タラオ)

+parent

grandfather(波平, タラオ) :- father(波平, ワカメ),
father(波平, カツオ), father(波平, サザエ),
mother(サザエ, タラオ),
parent(波平, ワカメ), parent(波平, カツオ),
parent(波平, サザエ), ~~parent(ワカメ, ？),~~
~~parent(カツオ, ？),~~ parent(サザエ, タラオ), ~~parent(タラオ, ？)~~

MSHの生成

- 各正例とモード宣言から生成
  - 例：grandfather(波平, タラオ)
    1. + ← 出現した定数
    2. - ← 背景知識を満たす定数
    3. 各定数を変数で置換.

モード宣言

```
:-modeh(*, grandfather(+person,-person))?
:-modeb(*, father(+person,-person))?
:-modeb(*, mother(+person,-person))?
:-modeb(*, parent(+person,-person))?
```

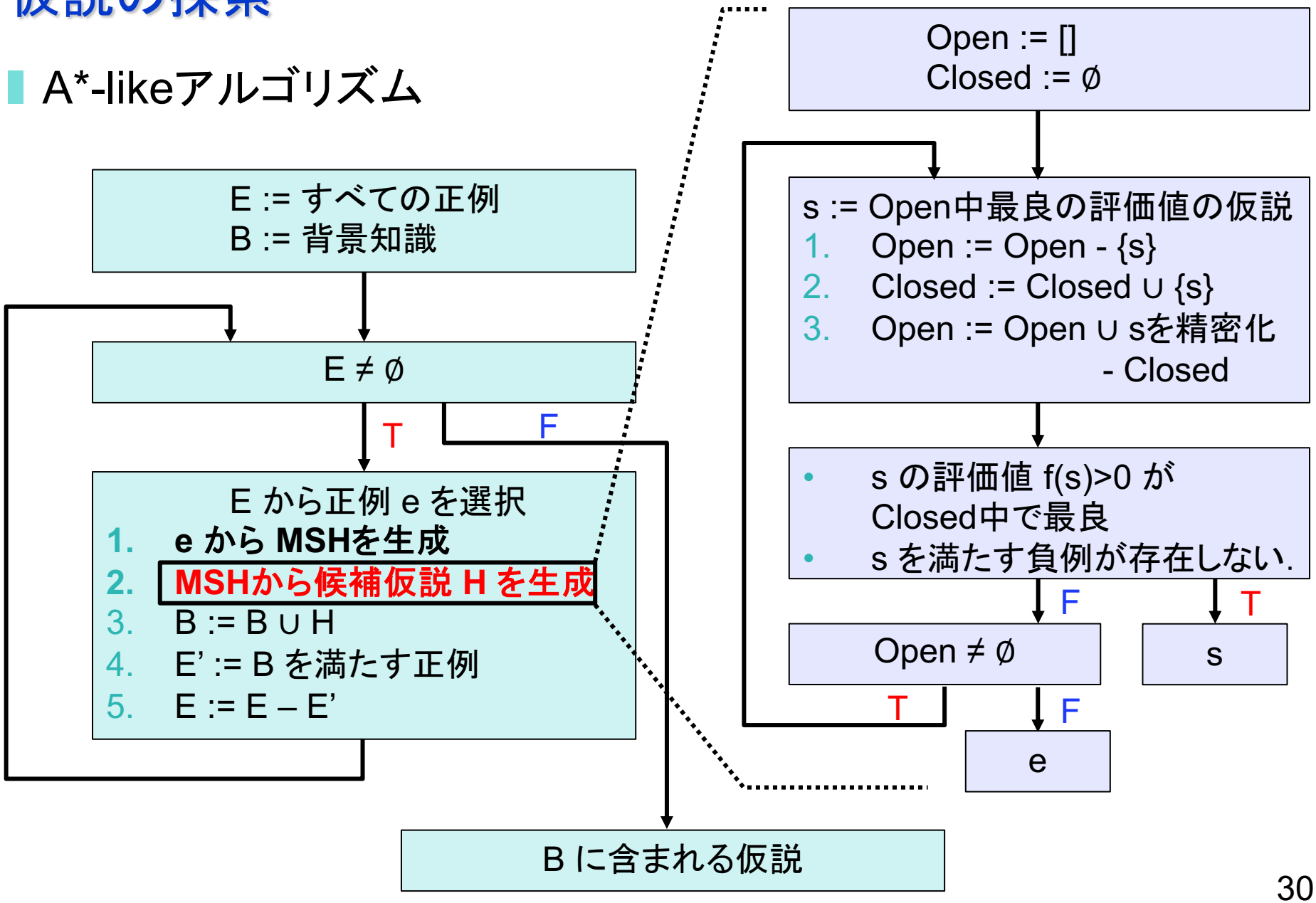
```
grandfather(波平, タラオ) :- father(波平, ワカメ),
        father(波平, カツオ), father(波平, サザエ),
        mother(サザエ, タラオ),
        parent(波平, ワカメ), parent(波平, カツオ),
        parent(波平, サザエ), parent(サザエ, タラオ)
```

波平 → A, タラオ → B, ワカメ → C, カツオ → D, サザエ → E

```
grandfather(A, B) :- father(A, C),
        father(A, D), father(A, E),
        mother(E, B),
        parent(A, C), parent(A, D),
        parent(A, E), parent(E, B)
```

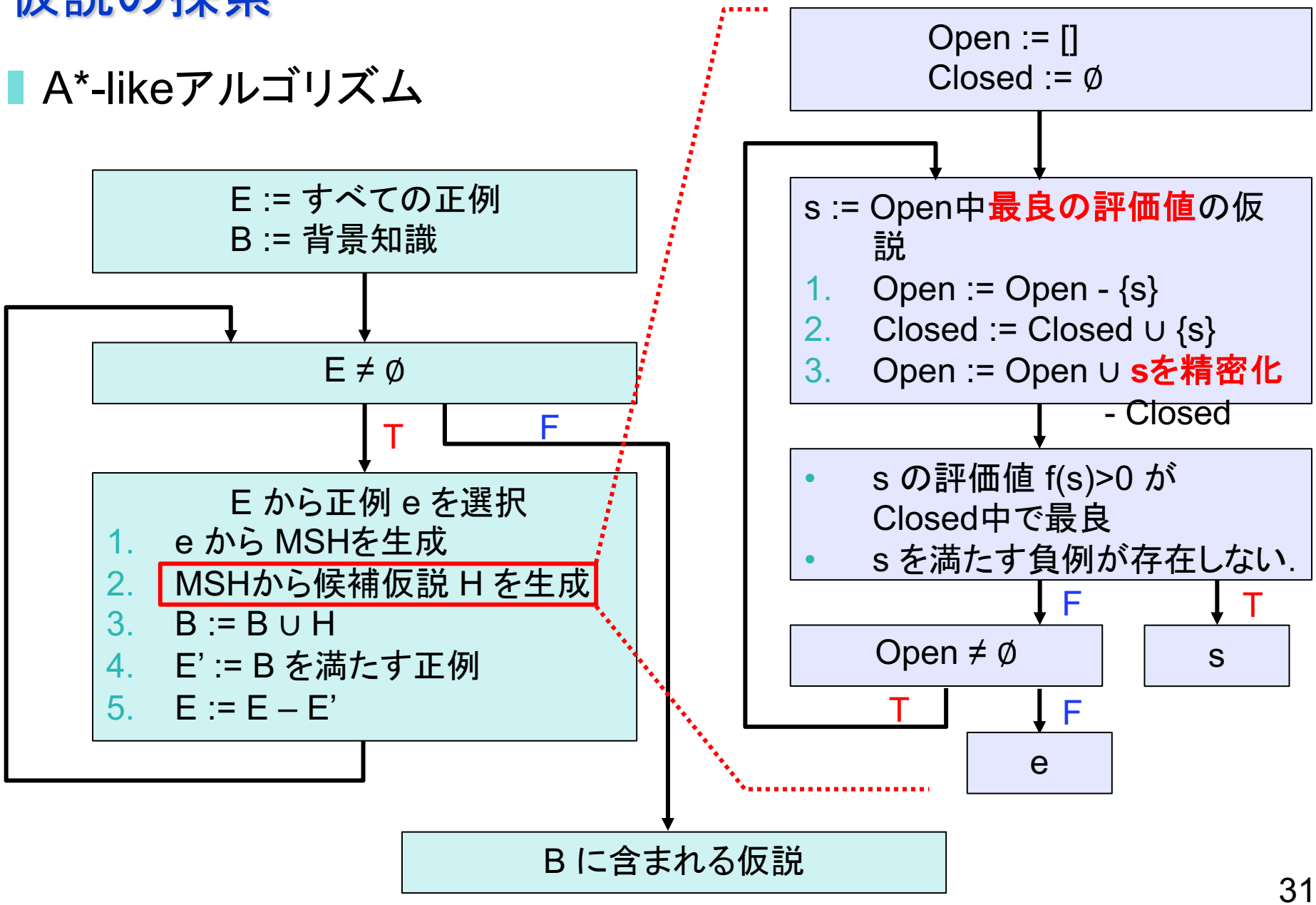
仮説の探索

- A*-likeアルゴリズム



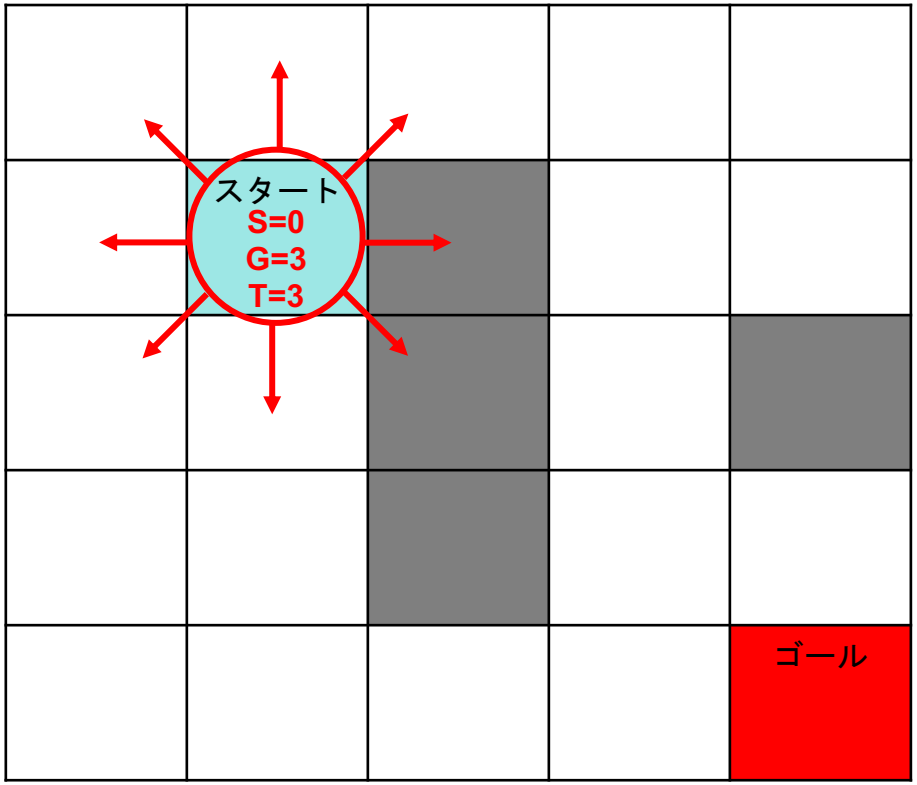
仮説の探索

- A*-likeアルゴリズム



仮説の探索

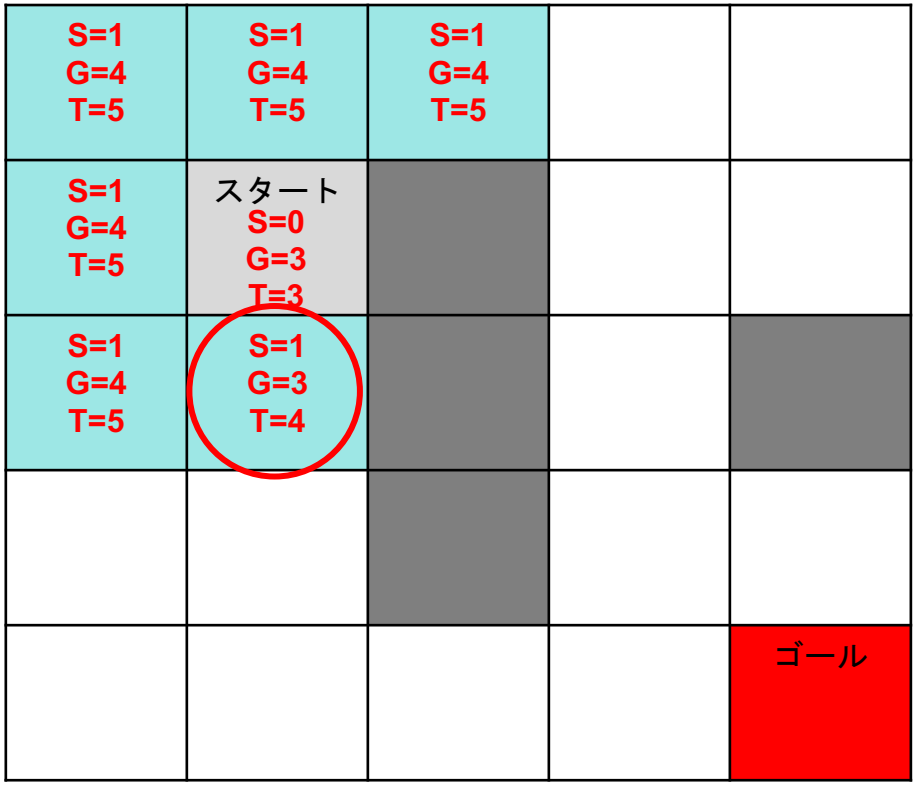
- A*アルゴリズム
  - **S**: スタートからの実距離，**G**:ゴールまでの予想距離，**T**:トータル（S+G）



  - スタートをオープン:S，G，T（評価値）を計算する．

仮説の探索

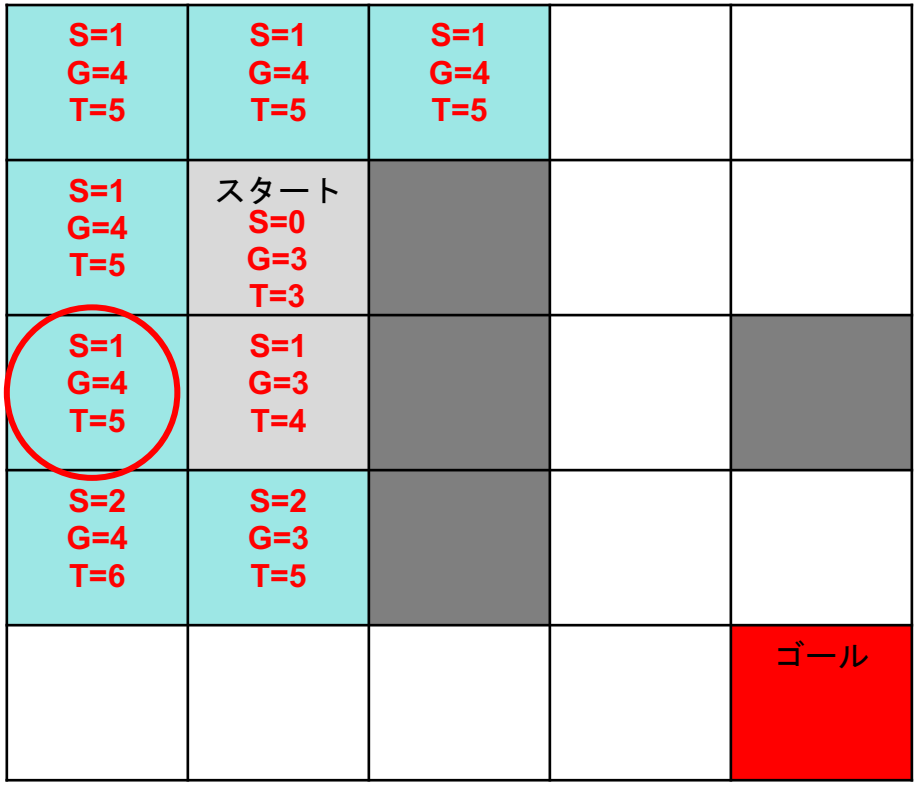
- A*アルゴリズム
  - スタートの隣接マスをオープン



  - スタートをクローズし，評価値（T）が最小のオープンマスを選択．

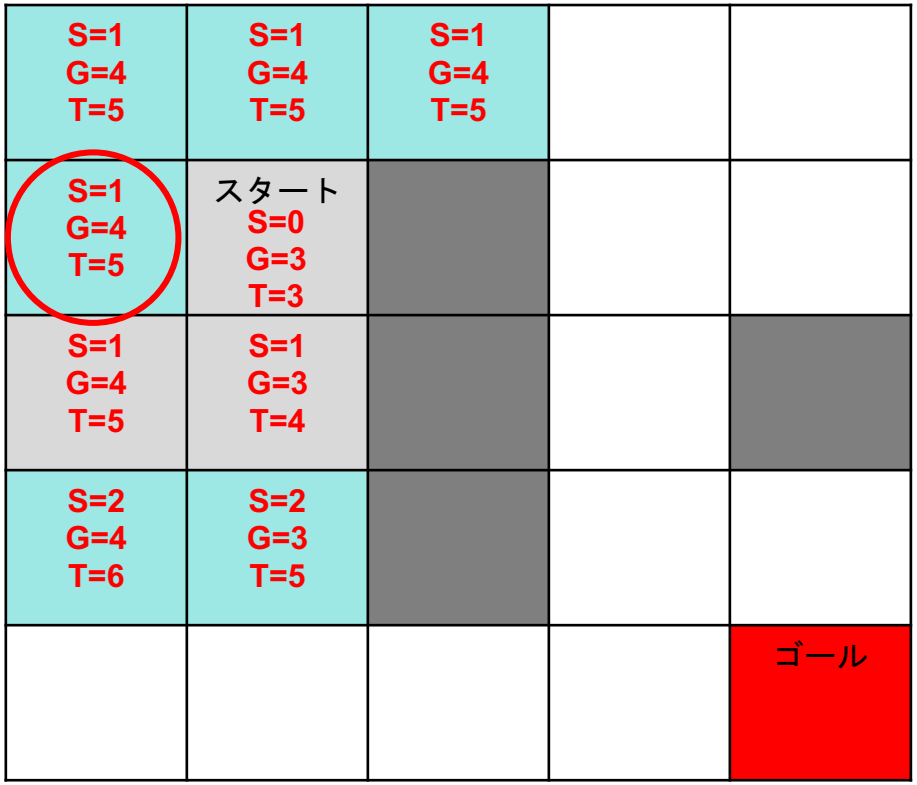
仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ



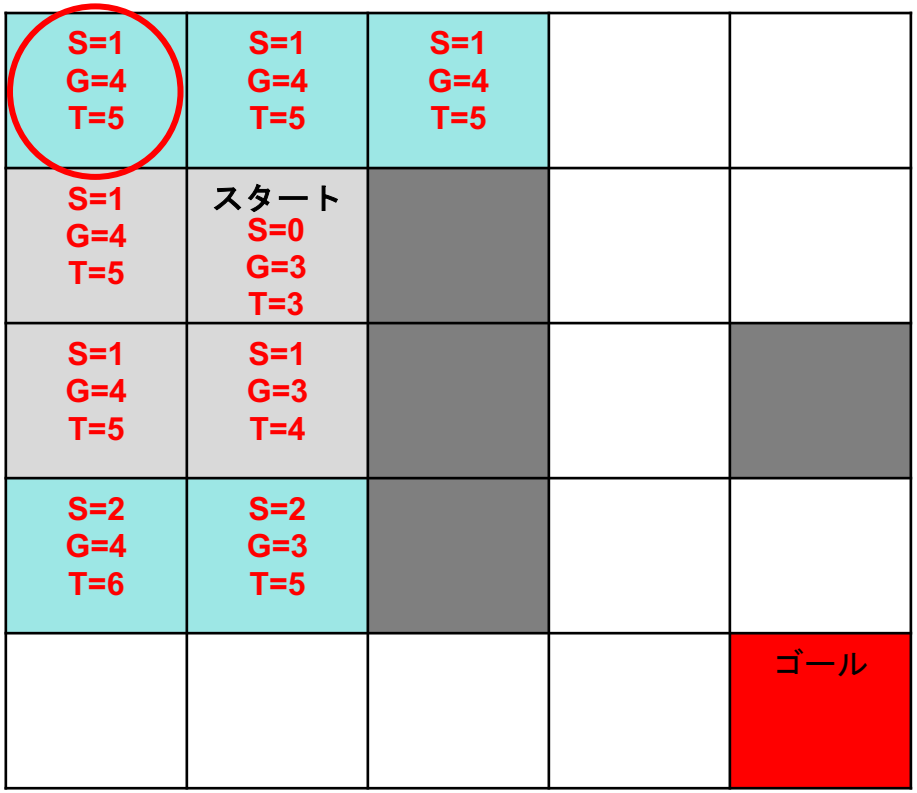
仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ



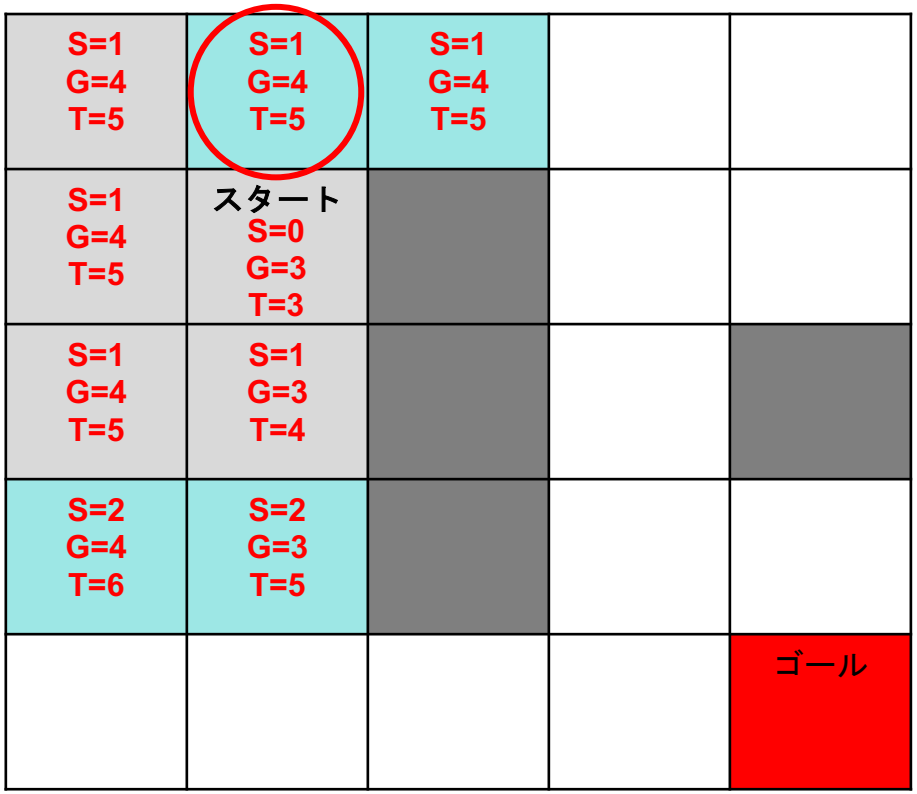
仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ



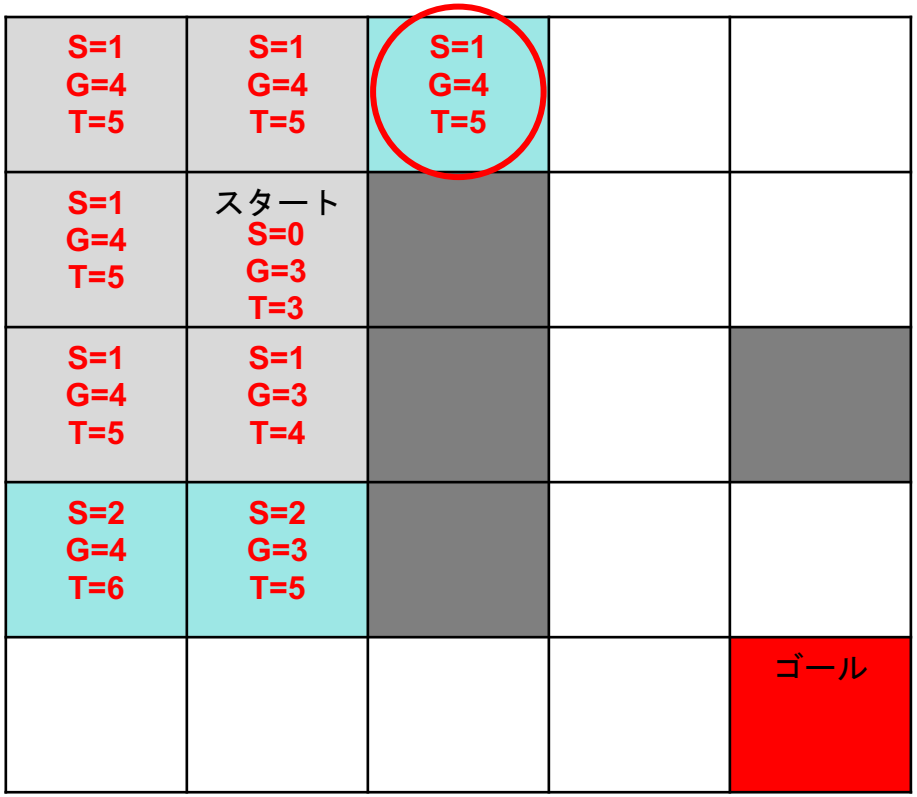
仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ



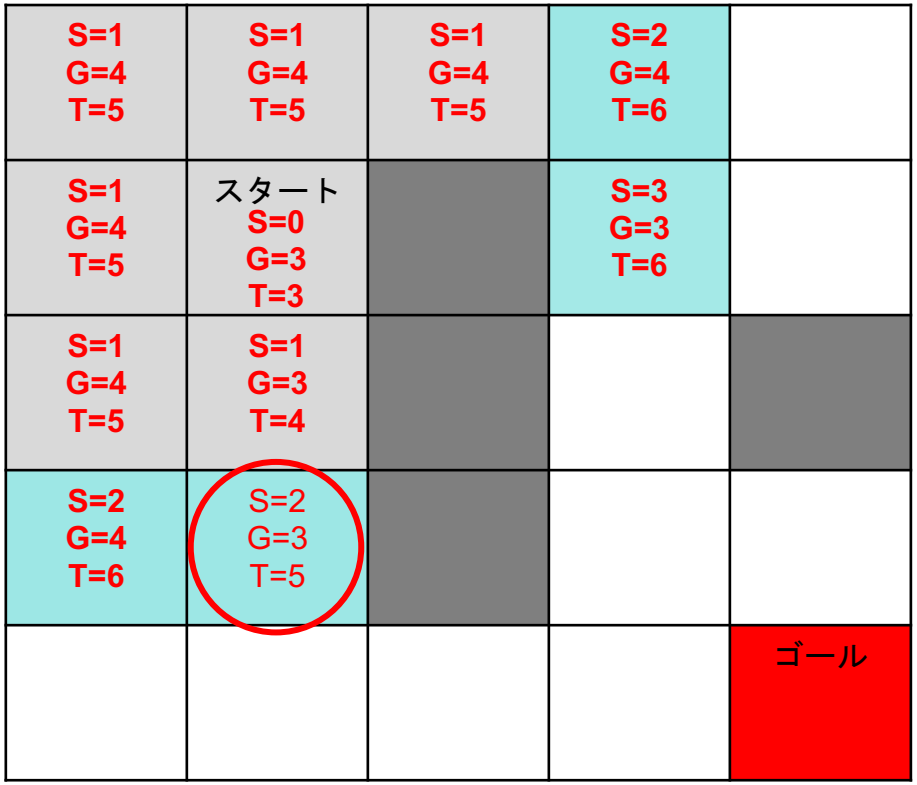
仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ



仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ



仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ

S=1 G=4 T=5	S=1 G=4 T=5	S=1 G=4 T=5	S=2 G=4 T=6	
S=1 G=4 T=5	スタート S=0 G=3 T=3		S=2 G=3 T=5	
S=1 G=4 T=5	S=1 G=3 T=4			
S=2 G=4 T=6	S=2 G=3 T=5			
S=3 G=4 T=7	S=3 G=3 T=5	S=3 G=2 T=5		ゴール

仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ

S=1 G=4 T=5	S=1 G=4 T=5	S=1 G=4 T=5	S=2 G=4 T=6	
S=1 G=4 T=5	スタート S=0 G=3 T=3		S=2 G=3 T=5	
S=1 G=4 T=5	S=1 G=3 T=4			
S=2 G=4 T=6	S=2 G=3 T=5			
S=3 G=4 T=7	S=3 G=3 T=5	S=3 G=2 T=5		ゴール

仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ

S=1 G=4 T=5	S=1 G=4 T=5	S=1 G=4 T=5	S=2 G=4 T=6	
S=1 G=4 T=5	スタート S=0 G=3 T=3		S=2 G=3 T=5	
S=1 G=4 T=5	S=1 G=3 T=4			
S=2 G=4 T=6	S=2 G=3 T=5		S=4 G=1 T=5	
S=3 G=4 T=7	S=3 G=3 T=5	S=3 G=2 T=5	S=4 G=1 T=5	ゴール

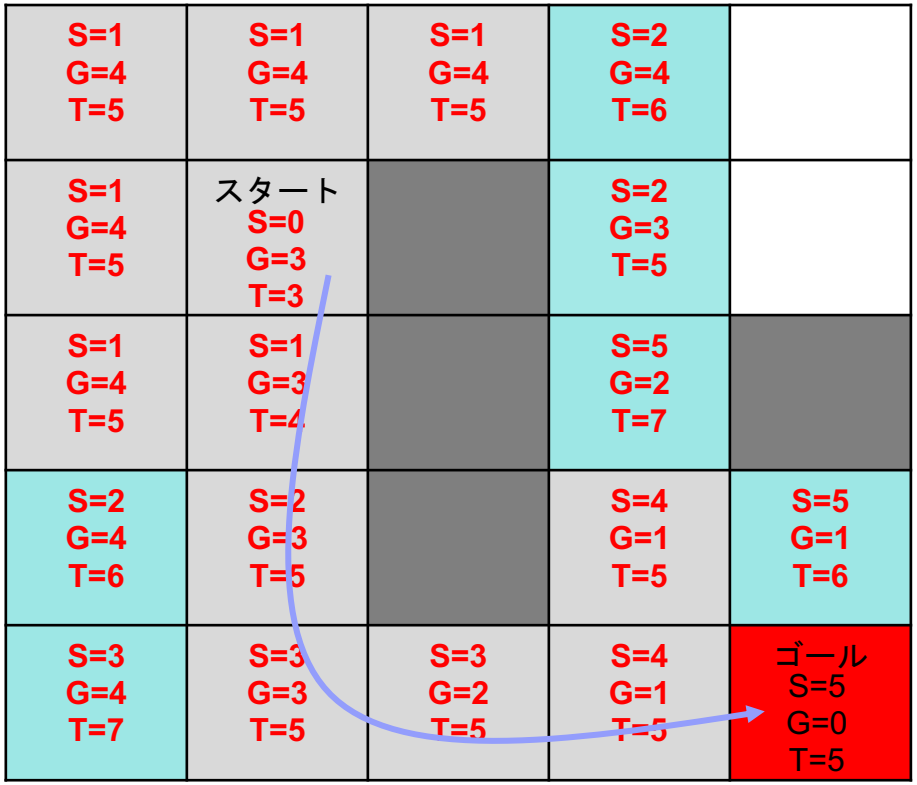
仮説の探索

- A*アルゴリズム
  - 評価値最小のマスを選択 → 隣接マスをオープン → 現在のマスをクローズ

S=1 G=4 T=5	**S=1 G=4 T=5**	**S=1 G=4 T=5**	**S=2 G=4 T=6**	
S=1 G=4 T=5	スタート S=0 G=3 T=3		**S=2 G=3 T=5**	
S=1 G=4 T=5	S=1 G=3 T=4		**S=5 G=2 T=7**	
S=2 G=4 T=6	S=2 G=3 T=5		**S=4 G=1 T=5**	**S=5 G=1 T=6**
S=3 G=4 T=7	S=3 G=3 T=5	**S=3 G=2 T=5**	S=4 G=1 T=5	ゴール

仮説の探索

- A*アルゴリズム

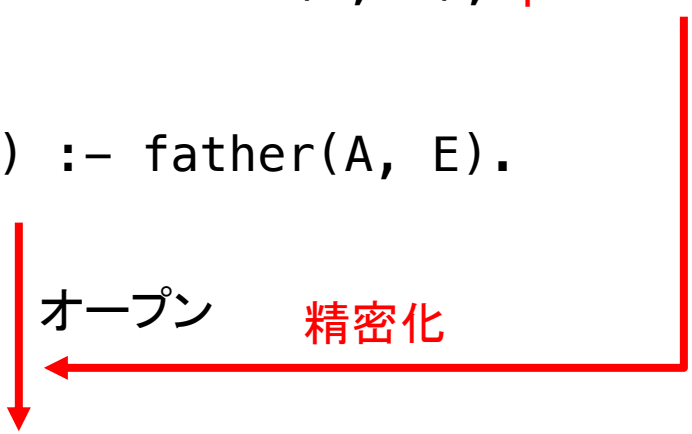


仮説の探索

- A*-likeアルゴリズム
  - 精密化（オープン）：現仮説候補にMSHからリテラルを加える

```
MSH: grandfather(A, B) :-
                father(A, C), father(A, D), father(A, E),
                parent(A, C), parent(A, D), parent(A, E),
                              mother(E, B), parent(E, B).
```

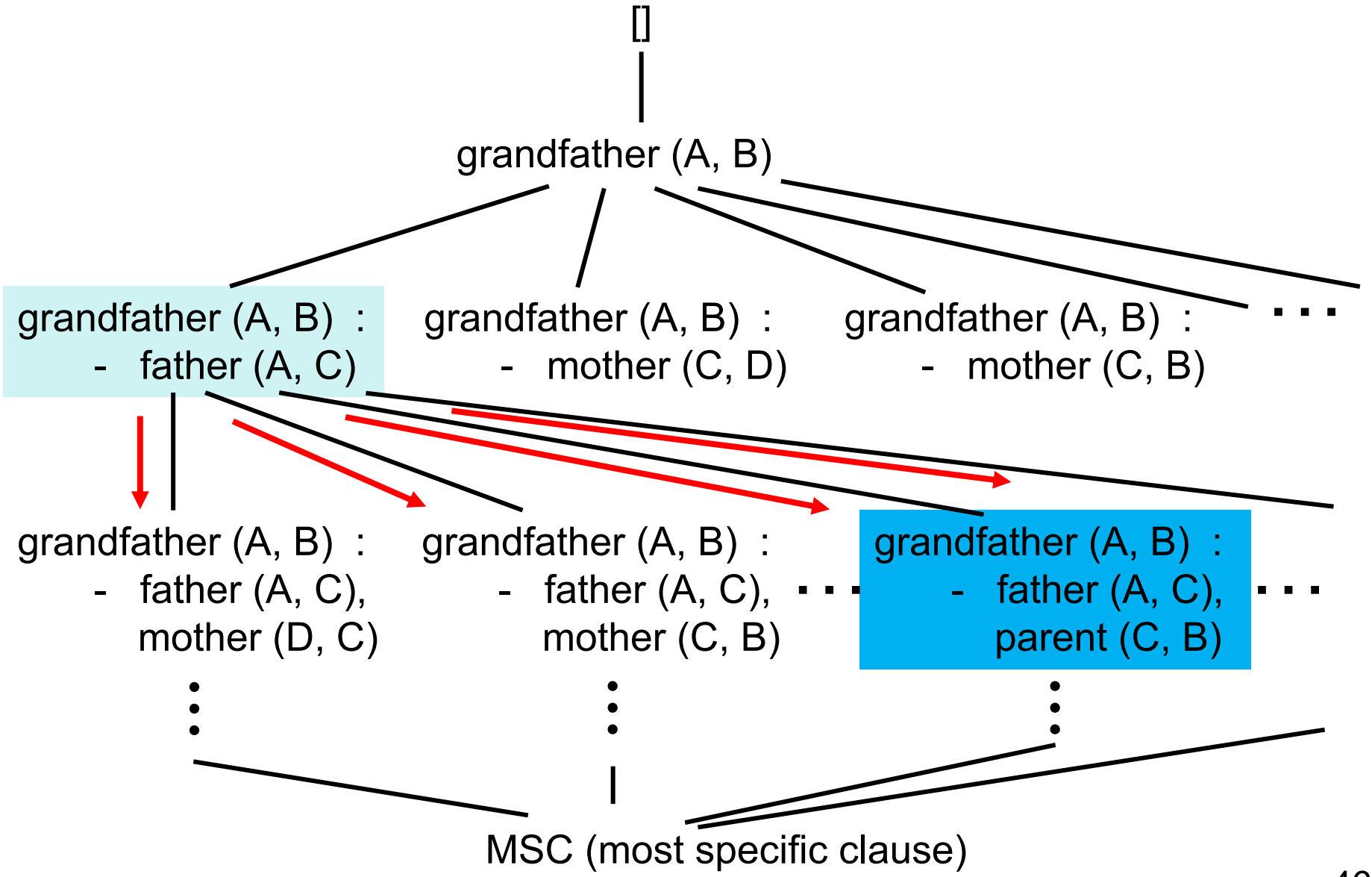
現仮説候補：`grandfather(A, B) :- father(A, E).`



新仮説候補：`grandfather(A, B) :- father(A, E), parent(E,B).`

評価値の計算

仮説の探索


[]
grandfather (A, B)
grandfather (A, B) :
- father (A, C)
grandfather (A, B) :
- mother (C, D)
grandfather (A, B) :
- mother (C, B)
grandfather (A, B) :
- father (A, C),
mother (D, C)
grandfather (A, B) :
- father (A, C),
mother (C, B)
grandfather (A, B) :
- father (A, C),
parent (C, B)
MSC (most specific clause)

仮説の探索

A*-likeアルゴリズム

- 被覆：正例や負例が仮説を満たすとき，その仮説はその正例や負例を被覆する．
- 評価関数 f:　評価値を計算（スコアが大きいほうがよい）

$$f = p - (n + c + h)$$

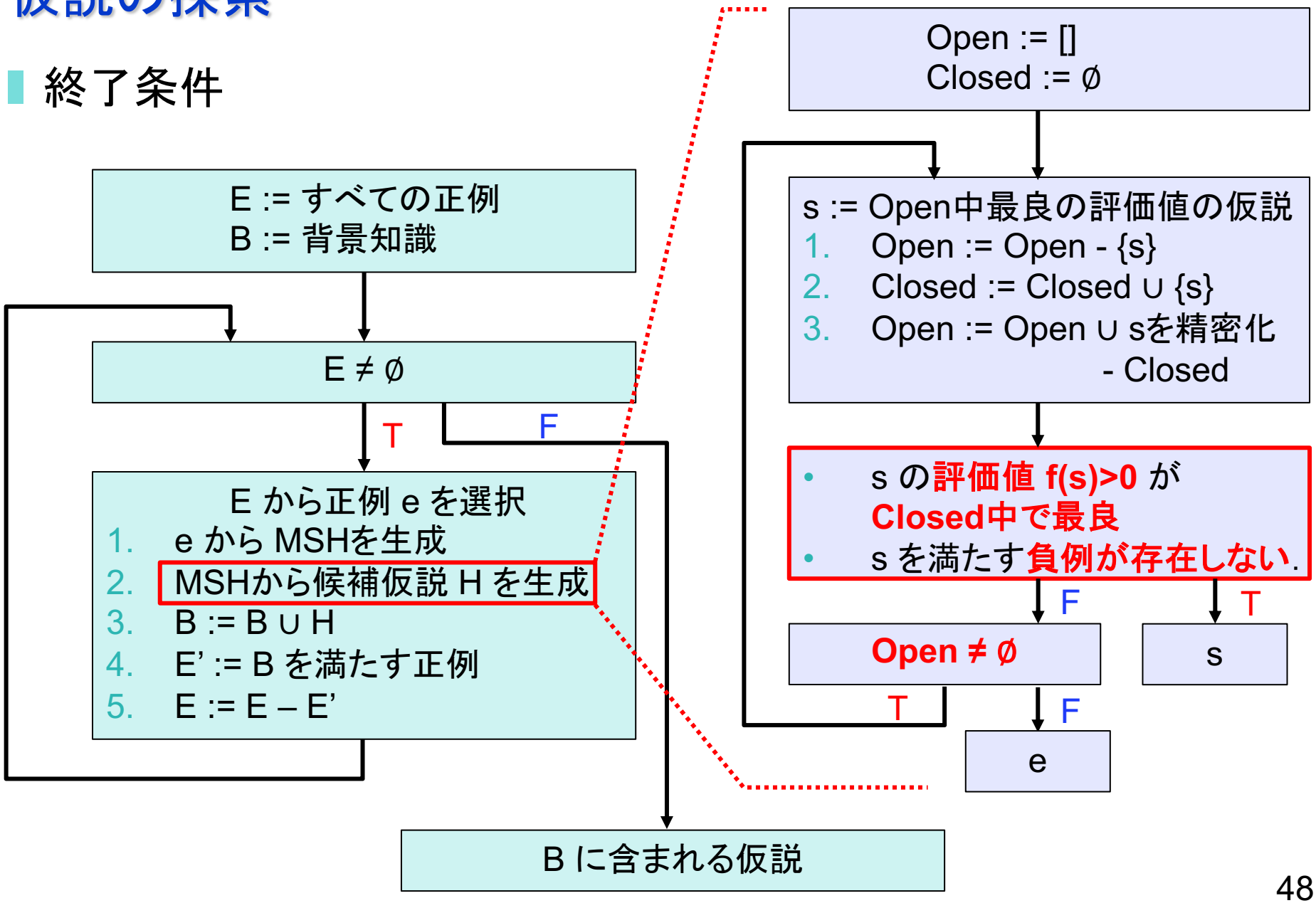
p	n	c	h
被覆する**正例の数**	被覆する**負例の数**	仮説のリテラル数	残りの予測リテラル数

1. 被覆する正例が多く，
2. 被覆する負例が少なく，
3. リテラル数が少ない（単純）

良い仮説

仮説の探索

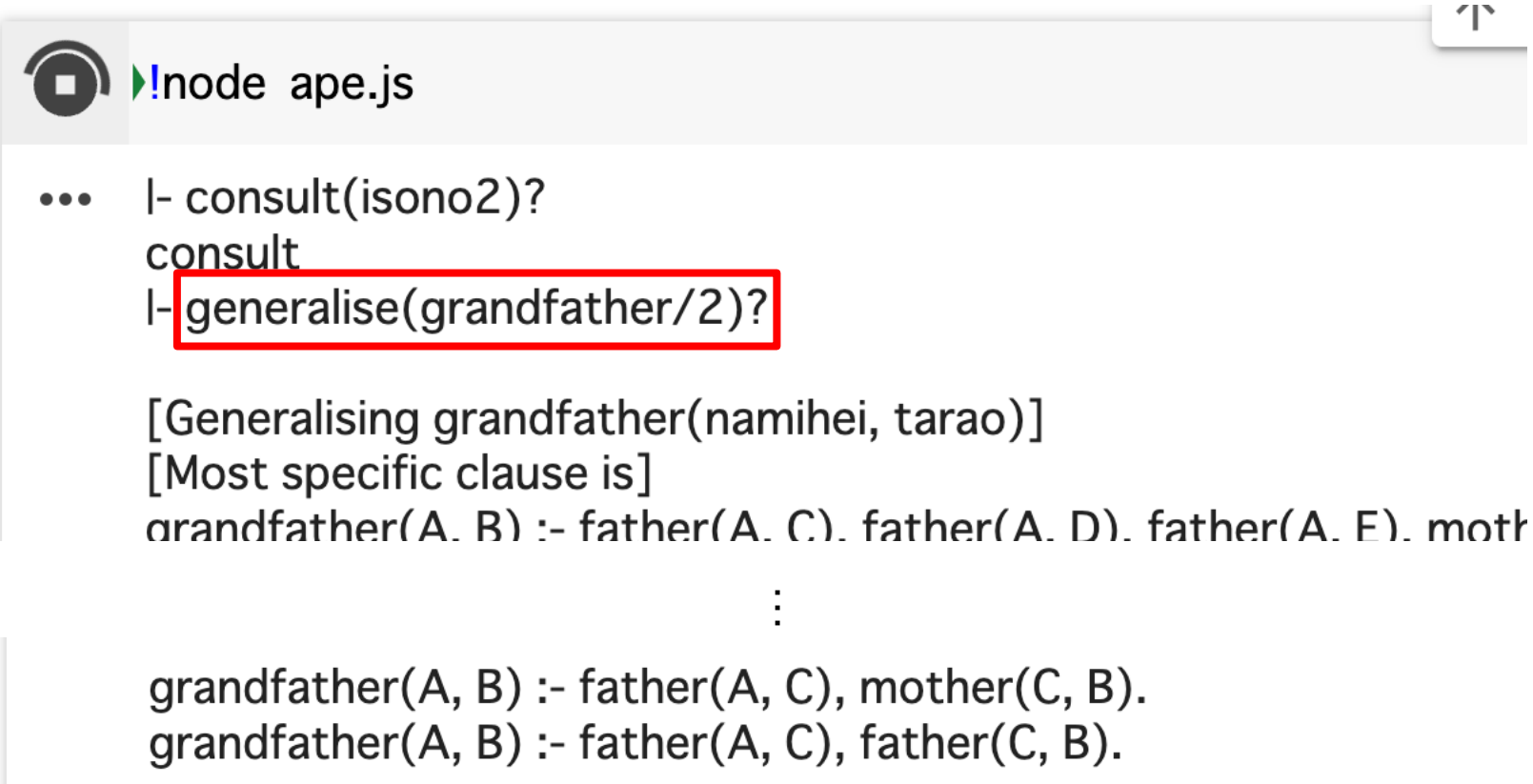
終了条件


Open := []
Closed := ∅
E := すべての正例
B := 背景知識
s := Open中最良の評価値の仮説
1. Open := Open - {s}
2. Closed := Closed ∪ {s}
3. Open := Open ∪ sを精密化
- Closed
E ≠ ∅
T
F
• s の評価値 f(s)>0 が Closed中で最良
• s を満たす負例が存在しない.
E から正例 e を選択
1. e から MSHを生成
2. MSHから候補仮説 H を生成
3. B := B ∪ H
4. E' := B を満たす正例
5. E := E – E'
F
T
Open ≠ ∅
s
T
F
e
B に含まれる仮説

帰納論理プログラミング

演習2

- 「grandfather」の学習
  1. 「isono2.pl」を読み込む.
  2. 「generalise(grandfather/2)?」を実行.



帰納論理プログラミング

演習2

- 背景知識とmodeの付加
  1. 「isono2.pl」に次のmode宣言を付加：
     :-modeb(*, parent(+person,-person))?
  2. 「isono2.pl」に次の背景知識を付加：
     parent(X,Y) :- father(X,Y).
     parent(X,Y) :- mother(X,Y).
  3. 「isono2.pl」の読込み.
  4. 「generalise(grandfather/2)?」の実行.

- 「animals」の学習
  1. 「animals.pl」の読込み.
  2. 「generalise(class/2)?」の実行.

正例からの学習

- 確率論理プログラミング（SLP, stochastic logic programming）
  - 事例空間全体の確率分布 ＝ 正例の確率分布

  - SLPによって，確率的に正例の生成を試みる.

ほとんどが負例

  - 例：

モード宣言：	:- modeh(*,p(+list,-elem))?
背景知識：	elem(a). elem(b). elem(c).
	list([]).
	list([H|T]) :- elem(H), list(T).
正例：	p([b,c],c).
	p([a,b,c],b).
	p([a,a],a).

  - modehから節を生成：‘*p’(A,B) :- list(A), elem(B).

正例からの学習

- 確率論理プログラミング（SLP, stochastic logic programming）
  - 例：　モード宣言：　:- modeh(*,p(+list,-elem))?
    背景知識：　elem(a). elem(b). elem(c).
    list([]).
    list([H|T]) :- elem(H), list(T).
    正例：　p([b,c],c).
    p([a,b,c],b).
    p([a,a],a).

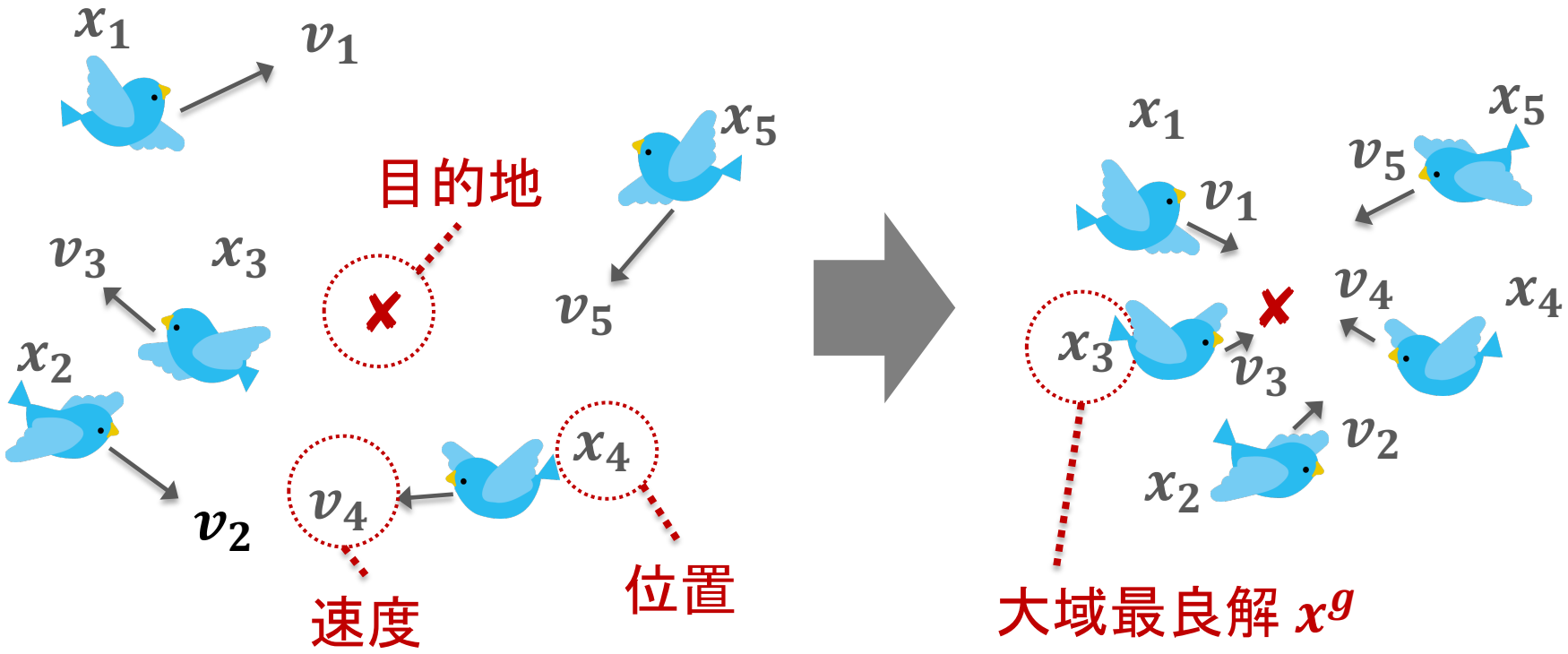
1. modehから節を生成：‘*p’(A,B) :- list(A), elem(B).
2. 確率の付与：すべての正例に対して，prologのゴール節として実行（融合操作数をカウント）．　4:elem(a). 3:elem(b). 3:elem(c).
   3:list([]). 7:list([H|T]) :- elem(H), list(T).
3. サンプリング：’*p’(X,Y). を実行
   listの選択：3/10の確率：list([]). 7/10の確率:list([H|T] :- elem(H), list(T).
   elemの選択：
   4/10の確率：elem(a). 3/10の確率:elem(b). 3/10の確率：elem(c).

帰納論理プログラミング

- 演習3
  - 「isono2.pl」の先頭にある「set(posonly)?」のコメントをはずして実行.
  - 「animals.pl」の先頭にある「set(posonly)?」のコメントをはずして実行.

メタヒューリスティクスの利用

- 粒子群最適化（particle swarm optimization, PSO）

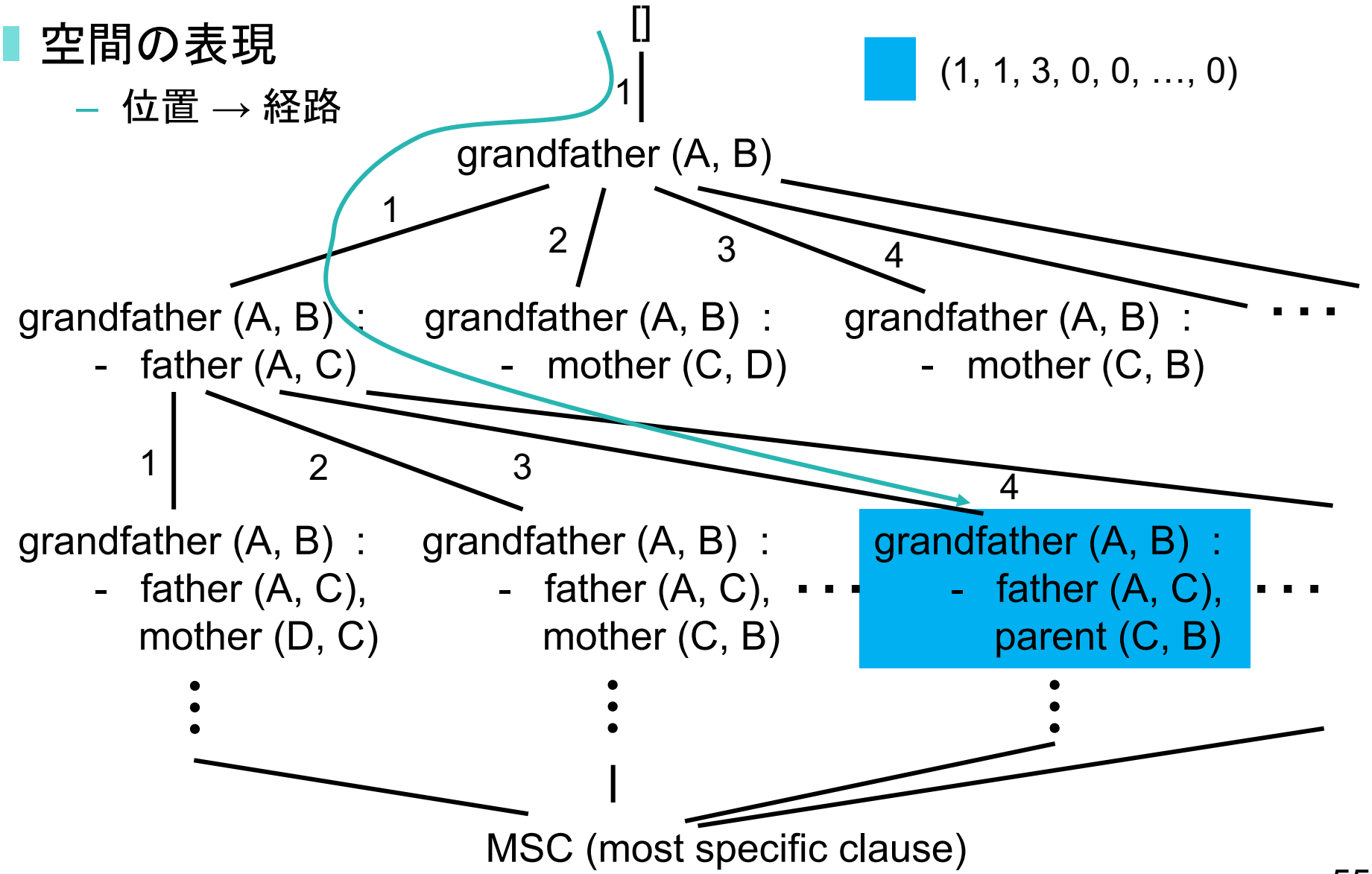


- $v_i = wv_i + c_1 r_1 (x^g - x_i) + c_2 r_2 (x_i^p - x_i)$
- $x_i = x_i + v_i$

w, c_1, c_2：パラメタ，r_1, r_2：ランダム値
x_i^p：粒子 i のこれまでの最良解

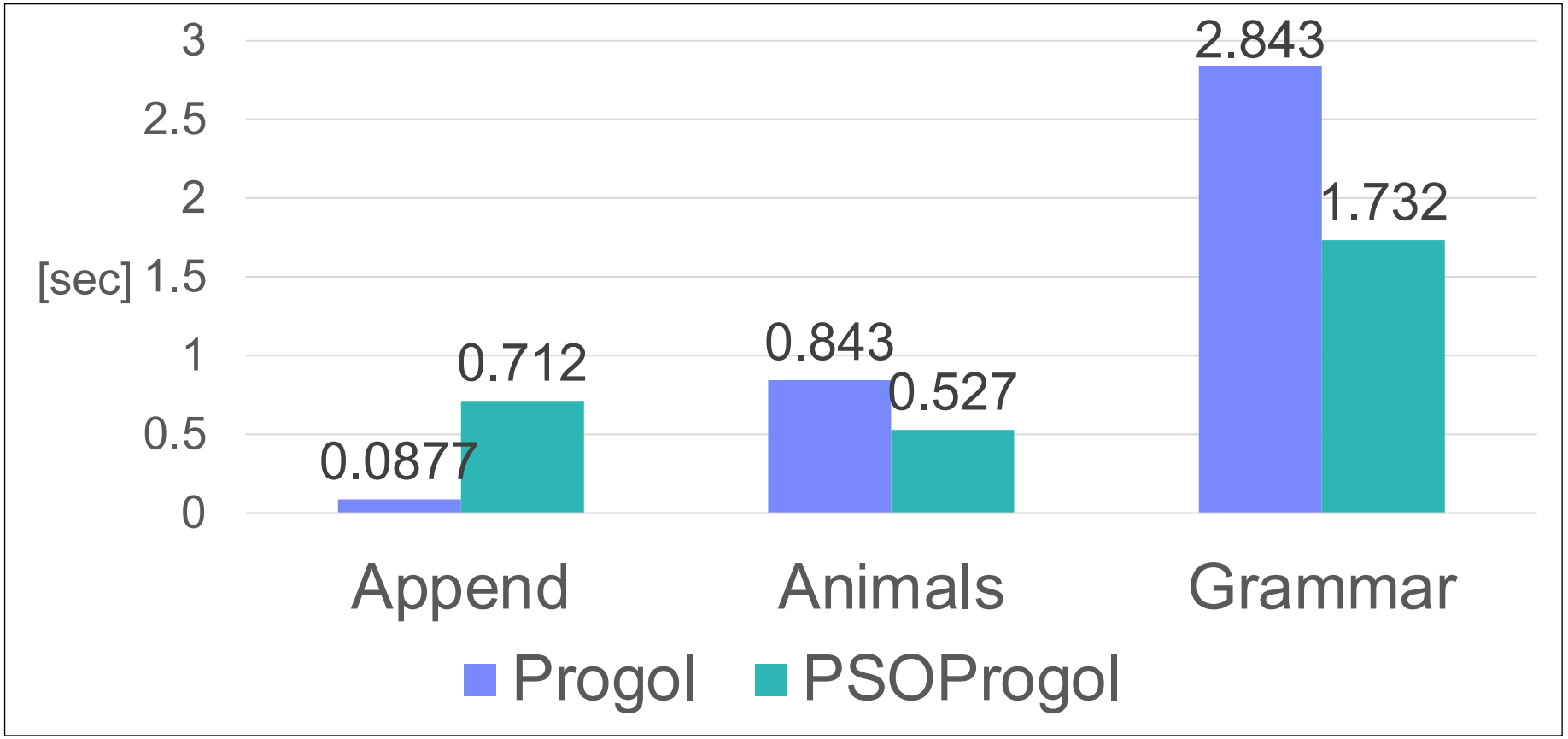
メタヒューリスティクスの利用

- 空間の表現
  - 位置 → 経路



メタヒューリスティクスの利用

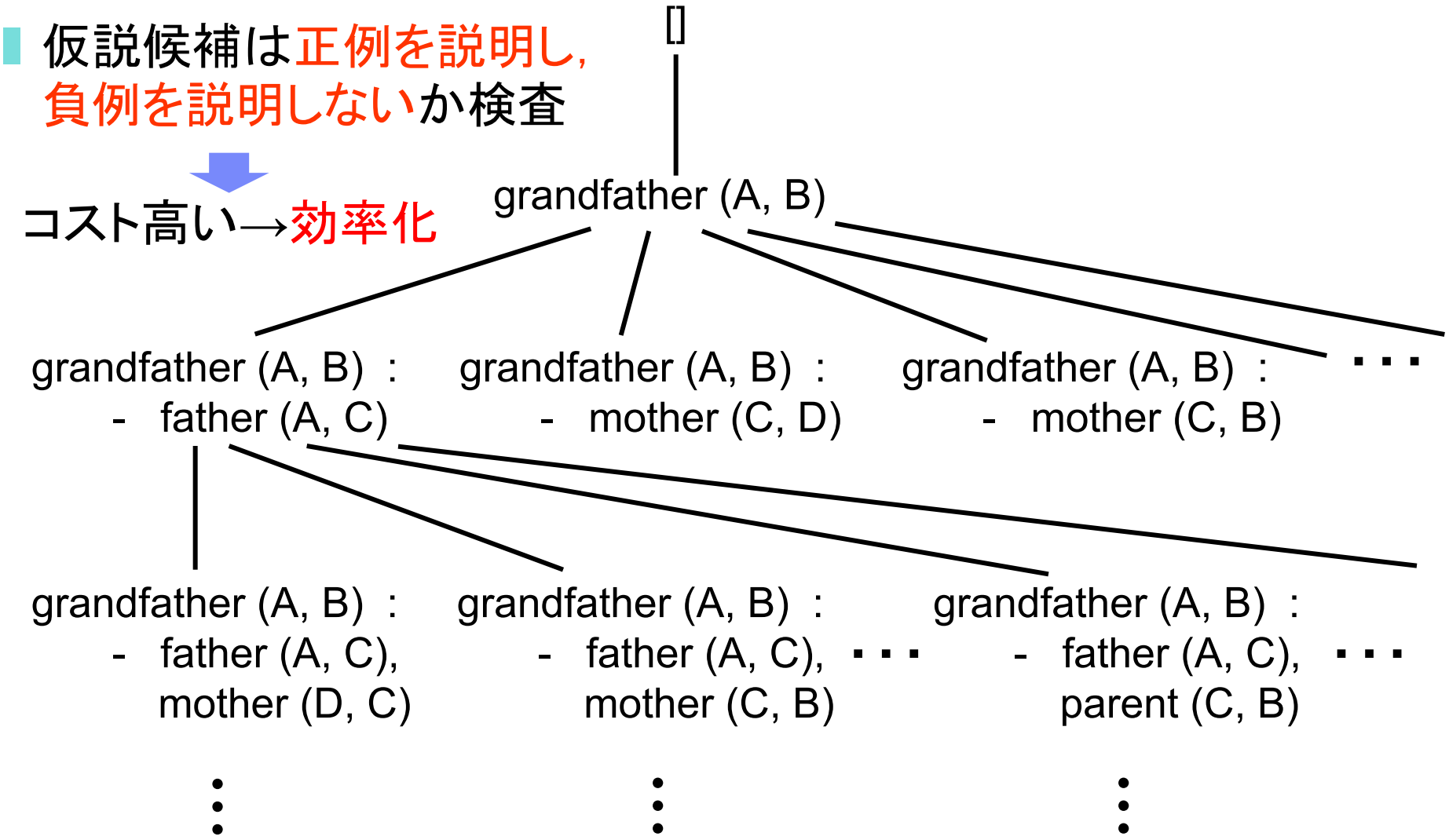
- 評価結果
  - 問題のサイズが大きくなるほど効果的



被覆検査の並列化

- 仮説候補は正例を説明し，負例を説明しないか検査

コスト高い→効率化



被覆検査の並列化

- 背景知識

father (波平, サザエ)　father (波平, カツオ)　father (波平, ワカメ)
father (藻屑, 波平)　mother (サザエ, タラオ)

parent (A, B) :- father (A, B)
parent (A, B) :- mother (A, B)

- 表による表現

father

波平	サザエ
波平	カツオ
波平	ワカメ
藻屑	波平

mother

サザエ	タラオ

father ∪ mother →　parent

波平	サザエ
波平	カツオ
波平	ワカメ
藻屑	波平
サザエ	タラオ

- 表操作（関係操作）

grandfather の表を生成　→　事例は含まれるか？

被覆検査の並列化

grandfather (A, B) :- **father** (A, **C**), **parent** (**C**, B)

father

A	C
波平	サザエ
波平	カツオ
波平	ワカメ
藻屑	波平

×(Cは一致)

parent

C	B
波平	サザエ
波平	カツオ
波平	ワカメ
藻屑	波平
サザエ	タラオ

→

tmptable

A	C	B
波平	サザエ	タラオ
藻屑	波平	サザエ
藻屑	波平	カツオ
藻屑	波平	ワカメ

grandfather (A, B) :- **tmptable** (A, **C**, B)

tmptable

A	C	B
波平	サザエ	タラオ
藻屑	波平	サザエ
藻屑	波平	カツオ
藻屑	波平	ワカメ

AとBだけ残す→

grandfather

A	B
波平	タラオ
藻屑	サザエ
藻屑	カツオ
藻屑	ワカメ

GPU向けSQLを用いた実現

- 被覆検査に必要な演繹をSQLに変換 → GPU向けSQLエンジン

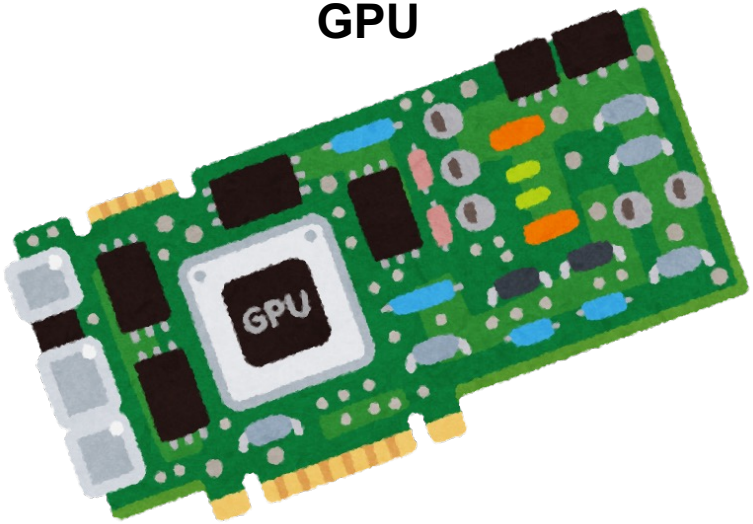
RDBMS

SQLに変換

並列論理学習システム

CPU

GPU



grandfather (A, B) :- father (A, C), parent (C, B)

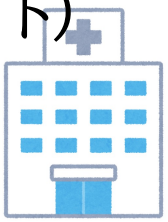
(father as t0 inner join parent as t1 on t0.c1 = t1.c0)
as tmptable

grandfather (A, B) :- tmptable(A,C,B)

select tmptable.c0, tmptable.c2 from

grandfather (A, B)

「がんゲノミクスデータサイエンス医療」(学長特別研究推進プロジェクト)

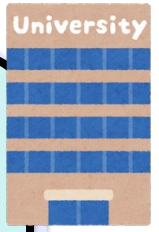


国立がんセンター東病院/NCC-EPOCに蓄積している医療データ

- ◉がんゲノム情報
- ◉がんオミクスデータ
- ◉医療情報
- ◉公共のがんゲノミクスデータ

公共の（がん）ゲノミクスデータ

- ◉ TCGA (The Cancer Genome Atlas)
- ◉ 東北メディカルメガバンク機構
- etc.



データサイエンスにより解決

情報処理
統計解析
機械学習
数理学

- ◉統合プラットフォーム
- ◉層別化
- ◉治療提案
- ◉バイオマーカー/治療標的分子の抽出
- ◉数理モデル etc.

（目標）「がんゲノミクスデータサイエンス医療」の基盤整備(2022年度)

まとめ

- 論理型A.I. は安心安全なA.I. を実現
- 東京理科大学の論理型A.I.は多くの成果を出している.
- 分散型探索と被覆並列化で高速な帰納論理プログラミングが実現できる.
- 論理型A.I.で，A.I.の新たなブレイクスルーを！
- 補完し合う深層学習と論理型A.I.へ.