認知科学と人工知能　Nr. 10（人工知能　Nr. 2）

小堀　聡

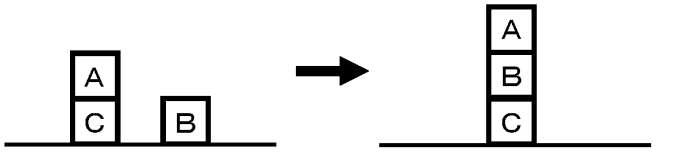
「問題解決と探索」

1．問題の表現

状態空間による表現

状態記述

初期状態
目標状態

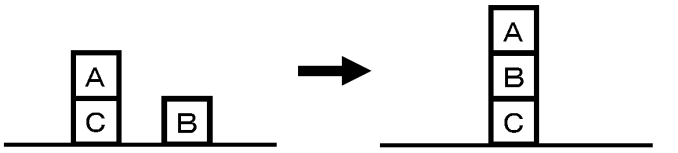


ON(A, C)
ON(B, TABLE)
ON(C, TABLE)

ON(A, B)
ON(B, C)
ON(C, TABLE)

オペレータ（作用素）

前提条件：オペレータを適用するための条件
削除リスト：オペレータ適用後に削除された状態記述
追加リスト：オペレータ適用後に追加された状態記述



ON(A, C)
ON(B, TABLE)
ON(C, TABLE)
CLEAR(A)
CLEAR(B)

ON(A, B)
ON(B, C)
ON(C, TABLE)
CLEAR(A)

拘束条件

目標を達成する際に守らなければならない条件

問題解決

初期状態から出発して，適用可能なオペレータを次々に作用させて，
拘束条件を守りながら目標状態に変換すること

グラフ表現

グラフ

節点，枝，道
有向グラフ，無向グラフ
親，子，先祖，子孫，根
閉路
木（ツリー）：閉路のない連結グラフ
出発節点：初期状態に対応する節点
目標節点：目標状態に対応する節点

2．基本的な探索

試行錯誤の探索（シラミつぶし）

適当な節点を選ぶ

open リスト：調べなければならない節点のリスト

procedure *search*

1　初期節点を *open* に入れる。

2　*LOOP*：**if** *open*＝空 **then** *exit* (*fail*)　；すべての節点を調べ終わったならば探索は失敗である。

3　*n*：＝ *first* (*open*)　；*open* の最初の要素を *n* とする。

4　**if** *goal* (*n*) **then** *exit* (*n*)　；*n* が目標節点ならば *n* を返す。

5　*remove* (*n*, *open*)　；*open* から *n* を取り除く。

6　*open* の中味を更新する。

7　**goto** *LOOP*　；ステップ 2 の *LOOP* へ戻る。

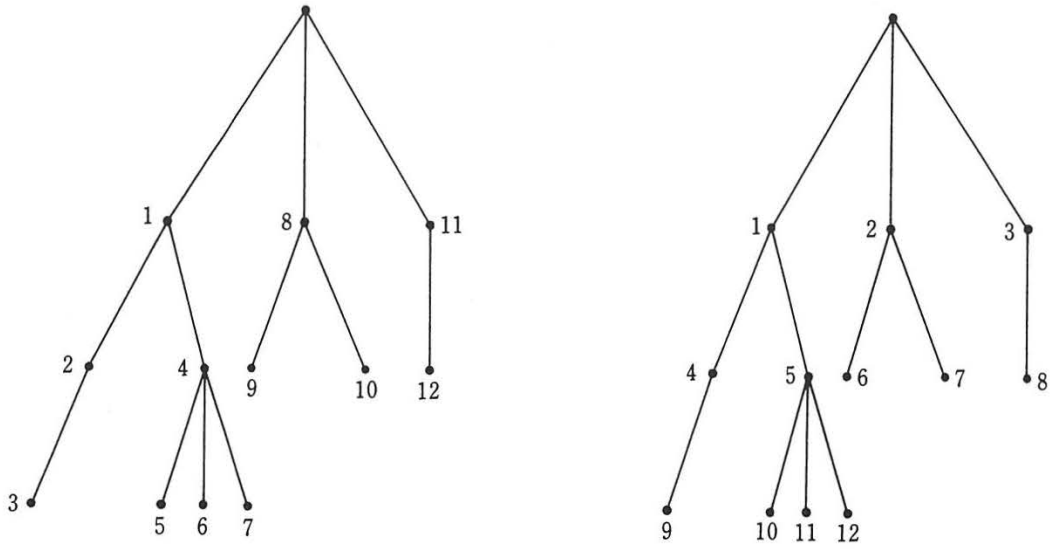
縦型探索（深さ優先探索）

探索順序：木の深い節点を先に調べる

横型探索（幅優先探索）

探索順序：木の浅い節点を先に調べる

縦型探索と横型探索の比較



（a） 縦型探索　　（b） 横型探索

縦型探索と横型探索の例

procedure *depth-first-search*

5　　*remove* (*n, open*)

　　add (*n, closed*)　　；*n* を *closed* に入れる。

6　　*n* を展開し，すべての子節点を生成する。子節点のうち，*open* あるいは *closed* の中に含まれない節点だけを *open* の先頭に入れ，そのおのおのから *n* へのポインタをつける。

procedure *breadth-first-search*

5　　*remove* (*n, open*)

　　add (*n, closed*)

6　　*n* を展開し，すべての子節点を生成する。子節点のうち，*open* あるいは *closed* の中に含まれない節点だけを *open* の最後部に入れ，そのおのおのから *n* へのポインタをつける。

最適解の探索

全コストを最小にする道を求める

例：巡回セールスマン問題（計算量はＮ！になる）

open リスト内のすべての節点をコストの小さい順に並べる

組合せ的爆発

メモリと計算速度の限界

３．コストの予測を用いた探索（※参考）

ヒューリスティクスの利用

山登り法

最も目標に近づくと予測される節点を選びながら目標に達しようとする方法

各節点でのコストを計算し，コストが最小となる節点を次の節点として選ぶ

目標以外の小さな「山」がある場合に失敗する可能性がある

最良優先探索

その時点までに得られているすべての節点の中から，最も目標に近い（コストが小さい）と思われる節点を選ぶ

Ａ*アルゴリズム

すべての節点に対して，目標までのコストの推定値が与えられているとき，

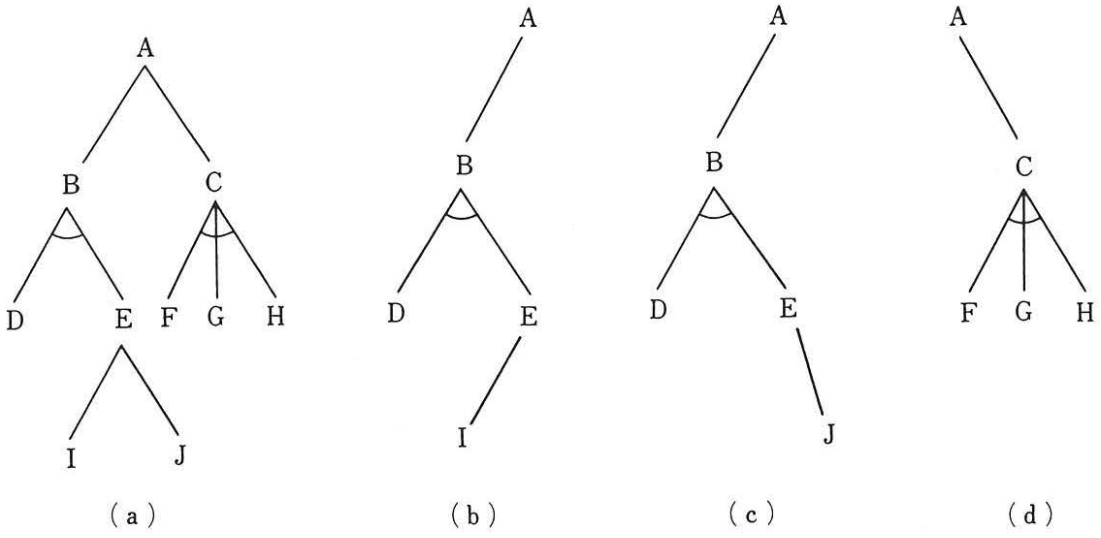
節点ｎを通る最適な道のコストの推定値を評価関数として用いる

ｎから目標までの推定値が真のコストを越えなければ最適解が得られる

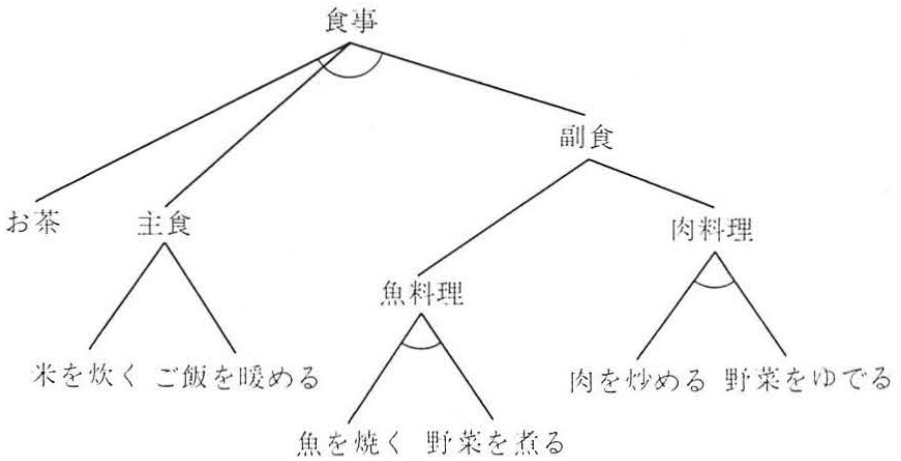
４．AND/OR グラフの探索

AND/OR グラフ

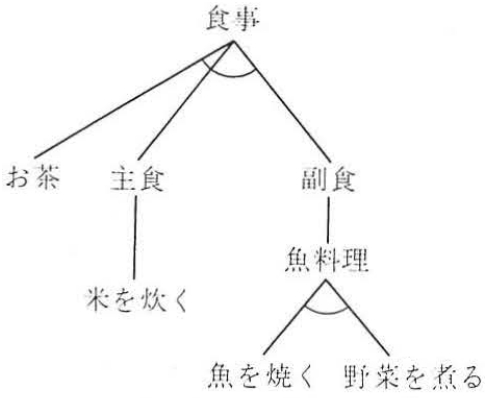
AND 節点：対応する問題のすべてを解決しなければならない
OR 節点：対応する問題のいずれかを解決すればよい
解グラフ：解を表す部分グラフ　（解は複数ある場合もある）
終端節点：解グラフの端点，解決した問題を表す



AND/OR 木の探索の説明図



食事の AND/OR グラフ



解グラフの例

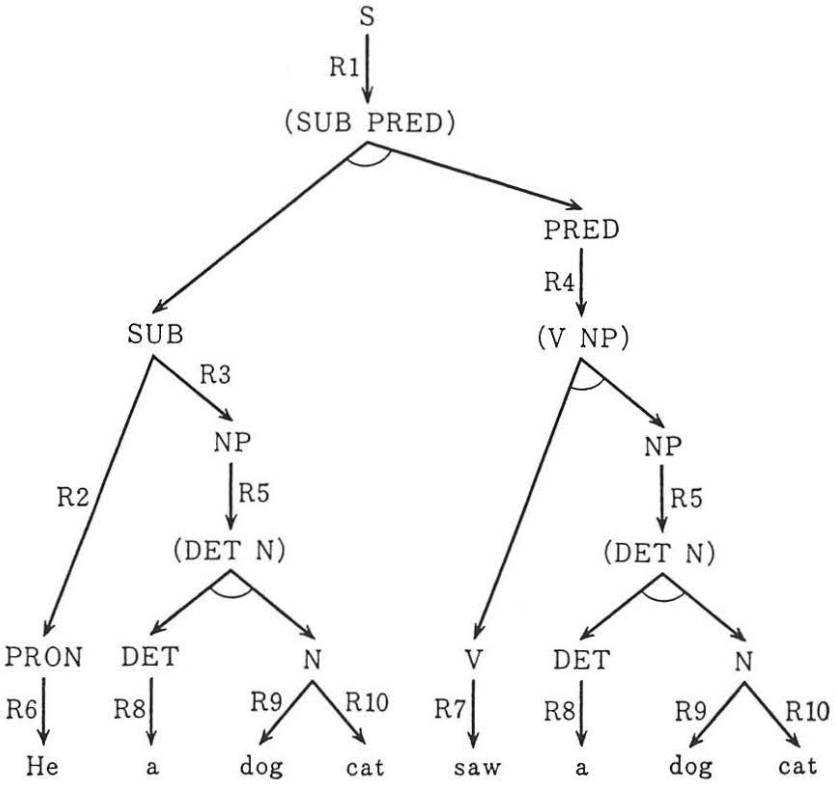
AND/OR グラフの性質

- 終端節点は解決されている節点である.
- ある節点の子節点が AND 節点の場合は，そのすべての AND 節点が解決されていれば，その節点も解決している.
- ある節点の子節点が OR 節点の場合は，その OR 節点のいずれか一つでも解決されていれば，その節点も解決している.
- 解グラフは，問題の AND/OR グラフの一部であり，出発節点を含み，どの節点も OR 節点への枝を２本以上持たず，すべての節点が解決されている AND/OR グラフである.

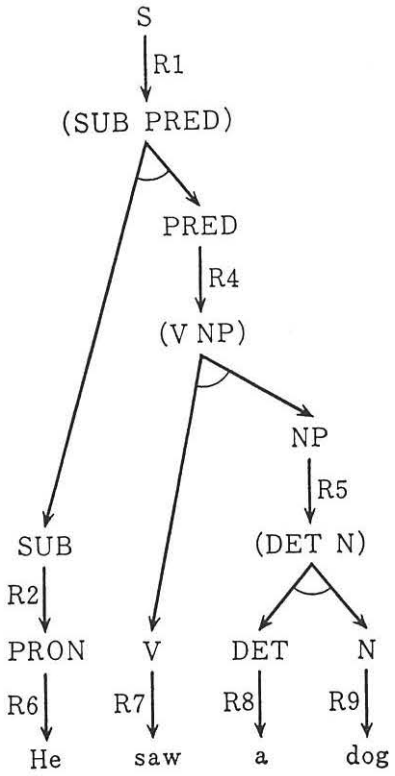
部分解グラフの評価と展開

- 部分解グラフの節点が解決されているか，解決不可能か，それともいずれとも断定できないかを決める．必要があれば節点のコストを求める.
- 部分解グラフを展開して成長させる.

通常のグラフ探索と同様，縦型探索，横型探索，最適探索などがある.



文章生成問題の AND/OR グラフ

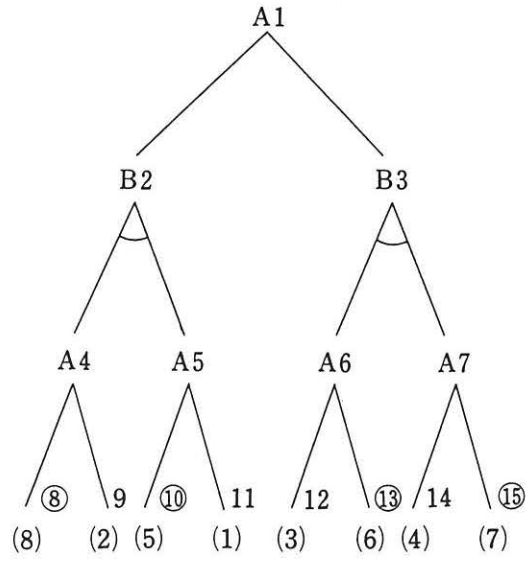


文章生成問題の１例

５．ミニマックス法

MAX：最初の手を指す人，（自分が）最大の評価値を得ることを目標とする
MIN：つぎの手を指す人，（相手が）最小の評価値を得ることを目標とする
双方は最善をつくすものとする.
MAX は，MIN の節点にはその子節点の最小の評価値が与えられることを仮定して，自分の手を選ぶ.

木の深さが一定以上になると，調べなければならない節点の数は急激に増大する
　→　「枝刈り」が必要

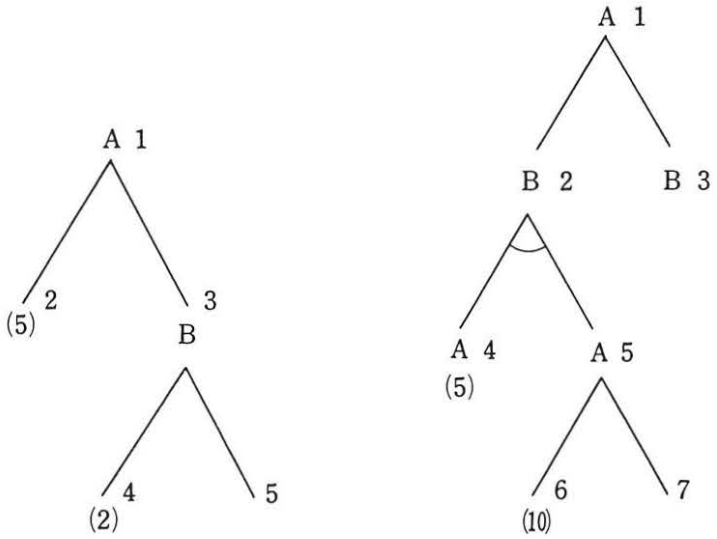


ゲーム木の探索におけるミニマックス法の説明図

６．アルファ・ベータ枝刈り

アルファカット：MIN のすべての子節点の評価を行わずに，その親節点である MAX の評価値を決める方法

ベータカット：MAX のすべての子節点の評価を行わずに，その親節点である MIN の評価値を決める方法



ゲーム木の探索におけるアルファ・ベータ法の説明図

(I) アルファカット　　(II) ベータカット

参考書

白井 良明：人工知能の理論（増補）（コロナ社）
長田 正 他著：ＫＥ養成講座・ＡＩ入門（オーム社）

キーワード

グラフ理論，巡回セールスマン問題
ヒューリスティックス，水平線効果