ヒューリスティック探索　（経験を用いた探索）

これまでに到達した探索木の末梢状態から展開される状態のうち，解に至る可能性の高い状態に注目し，探索の効率を高める．

末梢状態：　探索木上で，これまでに探索した端の状態．
展開：　与えられた節点に対し，直接移行可能な全ての後継状態を作り出すこと．

探索の効率化に用いる判断基準　（ヒューリスティック情報）

状態　ｓ　における評価関数　（全コスト）　ｆ（ｓ）

$$f(s) = g(s) + h(s)$$

ｇ（ｓ）：　コスト関数
ｈ（ｓ）：　ヒューリスティック関数

コストはゼロ以上：

$$g(s) \geqq 0, \quad h(s) \geqq 0$$

コスト関数　ｇ：　初期状態から現状態までのコストを与える．
同じ状態であっても，初期状態からの経路によりｇの値は異なる．

ヒューリスティック関数　ｈ：
現状態から目的状態までのコストの予測値を与える．
同じ目標状態へのｈであっても経路により値は異なる．

全コストが小さくなるような問題解決を進める．
ヒューリスティック情報，複数の探索枝候補からの選択法
→　問題，解法固有．

```
              ／
            ｓ１  h=7
        ２ ／ ＼ ４
h=6   ｓ２      ｓ３     h=5
 ７ ／ ＼      ／ ＼ ４
  ｓg                ｓg’
g(s1)+9           g(s1)+8
     実際のコスト
```

各枝の数字が，その枝のコスト．
ｓ１ までのコスト　g(s1)
ｓ１ からどちらに行こうか？

f（s2） = g(s1) + 2 + 6
　　　　= g(s1) + 8

f（s3） = g(s1) + 4 + 5
　　　　= g(s1) + 9

コスト関数とヒューリスティック関数に注目した効率化探索法

Ａアルゴリズム

初期状態も含めて，これまで展開された状態の評価関数値　ｆ（ｓ）が，より小さい状態を優先的に次の探索対象とする．

（ｇ　を探索木の深さ，ｈ　をゼロとすれば，横形探索）

Ａ＊ アルゴリズム

節点　のヒューリスティックス関数値　ｈ（ｓ）　が，節点　ｓ　から目標節点までの実際のコスト以下である，Ａアルゴリズム．

Ａ＊ アルゴリズムでは最小のコストの枝を辿る．

```
                    ／
                  ｓ1        h ＝ 7
              2 ／ ＼ 4
h ＝ 6        ｓ2       ｓ3      h ＝ 3
         7  ／ ＼    ／ ＼   4
        ｓg           ｓg’
       g1+9          g1+8
```

ｓ1 までのコスト：g1

ｓ1 からどちらに行くか？

ｆ（s2）＝ g1 ＋ 2 ＋ 6

ｆ（s3）＝ g1 ＋ 4 ＋ 3

ｓ3を選択

実際のコスト

最適解の探索

最小コストの解（最適解）が得られることを示す.

探索経路　（ｓｇ，　ｓｇ’：　目標状態）

ｓ0 → ｓ1 → ｓ2 → ･･･ → ｓｉ ･･･ → ｓｇ　　最適解

→ ｓｊ ･･･ → ｓg’　　非最適解

最適解に至る状態列：　ｓ0，ｓ1，ｓ2，･･･，ｓｉ，･･･，ｓｇ

最適解の全コスト：　f_best　（最適解の実際のコスト）．

これは ｆ（ｓg） に等しい.

ｓｉ での全コストの見積もり：　ｆ（ｓｉ）＝ｇ（ｓｉ）＋ｈ（ｓｉ）.

初期状態ならば ｆ（ｓ0）.

ｓｉでの実際の全コスト：　ｆ＊（ｓｉ）＝ｇ（ｓｉ）＋ｈ*（ｓｉ）.

これは最適のコスト f_best に等しい.

Ａ＊ の仮定より，　ｈ（ｓｉ）≦ ｈ*（ｓｉ）.

よって，ｆ（ｓｉ）≦ ｆ＊（ｓｉ）.

つまり，ｆ（ｓｉ）≦ f_best.

今，最適でない経路で達する目標状態　ｓg’　があるとする.

すると，f_best ＜ ｆ （ｓg’）.

これより，ｆ（ｓｉ）≦　f_best ＜ ｆ （ｓg’）.

つまり，ｆ（ｓｉ）＜ ｆ （ｓg’）.

Ａアルゴリズムは，最小の　ｆ　の状態を選ぶ.

→　ｓg’　より　ｓｉ　が選ばれるはず.

↓

ｓg に到達する以前に　ｓg’が選ばれることはない.

非最適解に至る途中の状態　ｓｊ に対して，

f_best ＜ ｆ （ｓj）

であるならば，最適解が見つかった時点では，ｓｊ は選ばれていない.

Ａ＊ アルゴリズムは最適解を与える.

より情報を持つヒューリスティック関数　（ｈの良し悪し）

ｈ2 は ｈ1 より情報を持つ

iff　$h1(s) < h2(s) \leqq h^*(s)$

より情報を持つヒューリスティック関数

→　より少ない節点をたどって最適解にたどり着く.　(*1)

ｈ1 を用いたＡ* アルゴリズム：　Ａ1*

$f1 = g + h1$

ｈ2 を用いたＡ* アルゴリズム：　Ａ2*

$f2 = g + h2$

(*1) を示すために，次のことを示す.

Ｓ1：　Ａ1＊ により解を求めた時に展開される状態の集合,
Ｓ2：　Ａ2＊ により解を求めた時に展開される状態の集合,
において,
Ｓ1 ⊇ Ｓ2　(*2)

各Ａ＊ アルゴリズムで解が得られた時点を考える.

ｓ0 ＝ ｓg の場合：　展開せずに解探索は終り.

ｓ0 ≠ ｓgの場合：
１回目の展開：評価対象となる次の状態が全て展開されるため，Ａ1＊ でも Ａ2＊ でも，同じ数の状態が作られる.
ここで解が得られたとしても，確かに（*2）が成立.

探索が進み，Ａ1*，Ａ2* ともに解に達したとしよう．

（背理法の仮定）
Ａ1* が選択せず，Ａ2* が選択する状態，ｓｉ が存在する．

ｓｉ ∈ Ｓ1 でないが，ｓｉ ∈ Ｓ2 である．

探索が終了した時，Ａ* アルゴリズムが最適解を与えることより，

$$f1(sg) = f2(sg) = f\_best$$

となっている．

一方，終了時点でＡ1* が選択しない節点 ｓｉ に対しては，

$$f\_best < f1(si)$$

つまり，

$$f\_best < g(si) + h1(si) \qquad (*3)$$

また，ｓｉ は Ａ2* で探索終了時には選択されているため，

$$f2(si) \leqq f\_best \qquad (*4)$$

(*3)，(*4) より，

$$g(s_i)+h_2(s_i) < g(s_i)+h_1(s_i)$$

$$h_2(s_i) < h_1(s_i)$$

これは，

h2 は、h1 より情報を持つこと，つまり，

$$h_1(s) < h_2(s) \leqq h^*(s),$$

に反する.

よって背理法の仮定は誤り.

Ａ1* が展開せず，　Ａ2* が展開する状態は存在しない.

Ａ2* が展開する状態の数は Ａ1* が展開する状態の数以下である.

探索による問題解決の例

探索問題としての自然言語構文解析

「大きい机がある。」を解析する.

品詞

- STATEMENT： 文 （最後の 。を含む）
- S： 文本体 （ 。を含まない）
- SUBJ （subject）： 主部
- V （verb）： 動詞
- N （noun）： 名詞
- P （postposition）： 助詞
- ADJ （adjective）： 形容詞

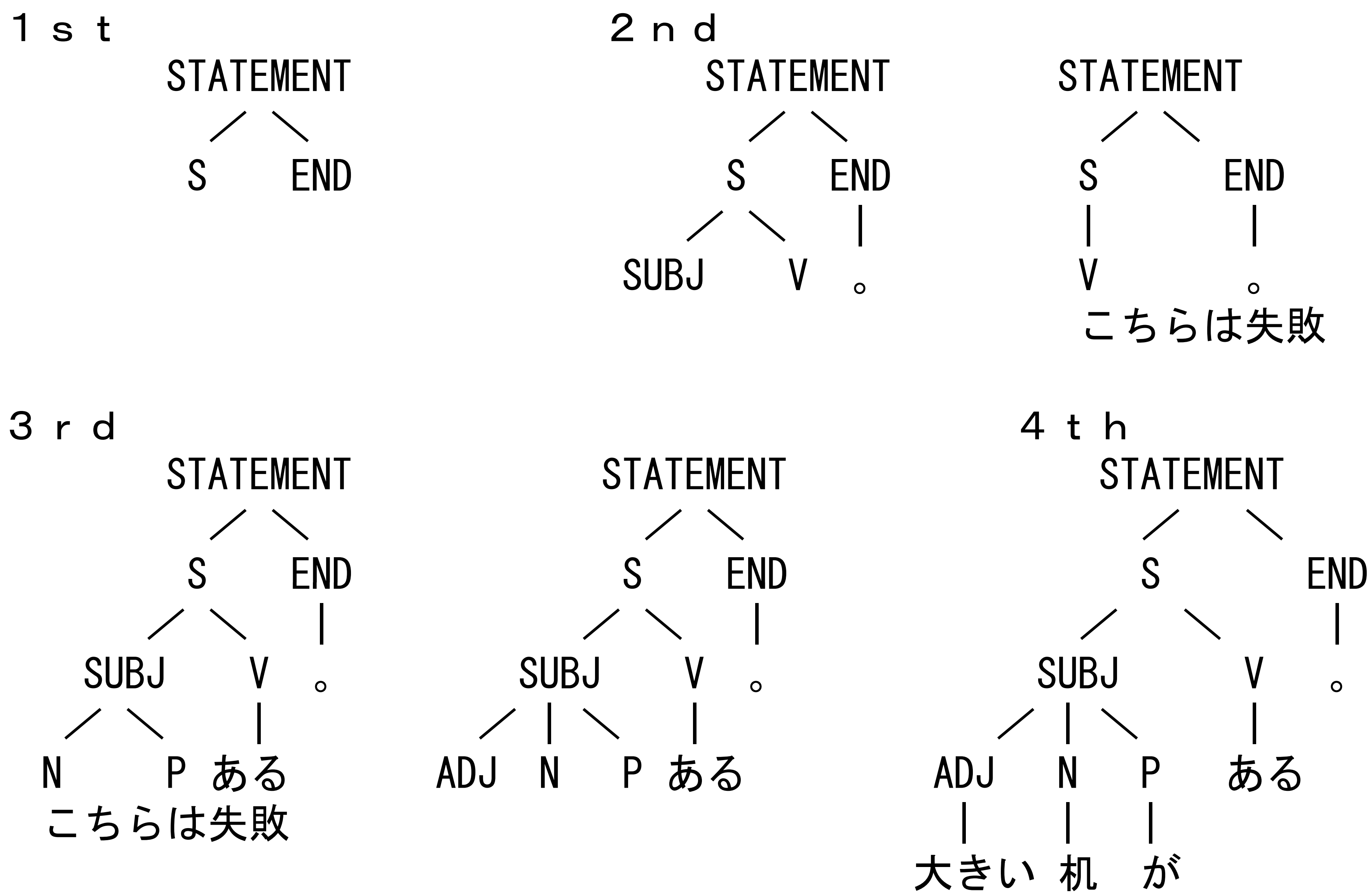
文法

- STATEMENT ← S END
- S ← SUBJ V
- S ← V
- SUBJ ← N P
- SUBJ ← ADJ N P

辞書

- END ← 。
- N ← 椅子
- N ← 机
- P ← が
- V ← ある
- ADJ ← 大きい
- ADJ ← 小さい

トップダウン法


１ｓｔ
STATEMENT
S END
２ｎｄ
STATEMENT
S END
SUBJ V 。
STATEMENT
S END
V 。
こちらは失敗
３ｒｄ
STATEMENT
S END
SUBJ V 。
N P ある
こちらは失敗
STATEMENT
S END
SUBJ V 。
ADJ N P ある
４ｔｈ
STATEMENT
S END
SUBJ V 。
ADJ N P ある
大きい 机 が

ボトムアップ法

１ｓｔ

```
 ADJ     N    P    V    END
  ｜     ｜   ｜   ｜    ｜
大きい   机   が  ある    。
```

２ｎｄ（ADJ を伸ばす）

```
      SUBJ        N    P    V    END
     ／｜＼       ｜   ｜   ｜    ｜
  ADJ   N   P     机   が  ある    。
   ｜
 大きい
```

３ｒｄ（N, P をつなげる, V を伸ばす）

```
        SUBJ        S     END
      ／ ｜ ＼      ｜     ｜
   ADJ   N   P      V      。
    ｜   ｜  ｜     ｜
 大きい  机  が    ある
```

４ｔｈ

S を伸ばし STATEMENT を作り,
END とつなげる. 解析終了.
→ 失敗
S を伸ばしたことを取り消す.
V を伸ばしたことを取り消す.
（バックトラック）

５ｔｈ　（SUBJ を伸ばす）

```
                S
              /   \
        SUBJ       V        END
       /  |  \               |
   ADJ    N    P      V       。
    |     |    |      |
 大きい   机    が    ある
```

６ｔｈ（S を伸ばす．V をつなげる）

```
               STATEMENT
                /    \
              S        END
            /   \
       SUBJ       V        END
      /  |  \     |         |
   ADJ   N   P   ある        。
    |    |   |
 大きい  机   が
```

７ｔｈ

END をつなげて完成