

アルゴリズムとデータ構造III 6回目:11月18日

構文解析 チャート法
グラフ ダイクストラ法

授業資料
<http://ir.cs.yamanashi.ac.jp/~ysuzuki/public/algorithm3/>

授業の予定(中間試験まで)

1	10/07	スタック(後置記法で書かれた式の計算)
2	10/14	チューリング機械, 文脈自由文法
3	10/21	構文解析 CYK法
4	11/04	構文解析 CYK法
5	11/11	構文解析(チャート法)
6	11/18	構文解析(チャート法), グラフ(ダイクストラ法, DPマッチング)
7	11/19 4時限 B2-41	グラフ(DPマッチング, A*アルゴリズム)
8	11/25	グラフ(A*アルゴリズム), 前半のまとめ
9	12/02	中間試験

授業の予定(中間試験以降)

10	12/09	全文検索アルゴリズム(simple search, KMP)
11	12/16	全文検索アルゴリズム(BM, Aho-Corasick)
12	01/06	全文検索アルゴリズム(Aho-Corasick), データ圧縮
13	01/13	暗号(黄金虫, 踊る人形) 符号化(モールス信号, Zipfの法則, ハフマン符号)テキスト圧縮
14	01/20	テキスト圧縮(zip), 音声圧縮(ADPCM, MP3, CELP), 画像圧縮(JPEG)
15	02/03	期末試験

本日のメニュー

- 構文解析
 - チャート法
 - 解析例
 - アルゴリズム
 - 動的計画法(最短距離探索)
 - ダイクストラ法
 - 解析例
 - アルゴリズム

構文解析アルゴリズム

- ボトムアップアルゴリズム
 - 戦略
 - 単語列から出発
 - Sを導出 → 解析終了
 - 代表的なアルゴリズム
 - CYK法
 - LR法
- トップダウンアルゴリズム
 - 戦略
 - S(ルートノード)から出発
 - 目的の単語列を導出 → 解析終了
 - 代表的なアルゴリズム
 - トップダウンチャート法
 - アーリー法(Earley parser)
 - LL法

チャート法(構文解析)

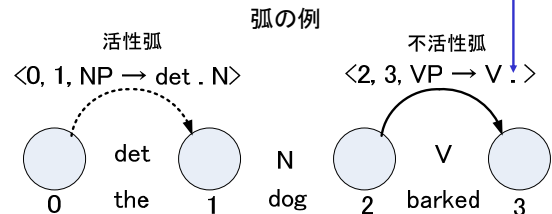
- ▶ ■ トップダウンチャート法
 - Sから出発
 - 目的の単語列を導出 → 解析終了
- ボトムアップチャート法
 - 単語列から出発
 - Sを導出 → 解析終了

チャート法で使用する用語 1/3

- 節点(ノード)
 - 単語と単語の間に存在する仮想的な点
- 弧(アーク)
 - 節点間を結び、文の部分的な構造を表す
 - $\langle i, j, C \rightarrow \alpha . \beta \rangle$
 - i は弧の始点, j は弧の終点
 - $.$ は解析が終了している位置
 - 節点 i から j まで解析すると α
 - β まで解析できると C

チャート法で使用する用語 2/3

- 不活性弧
 - 右辺の最後に「 $.$ 」がある弧
- 活性弧
 - 不活性弧以外の弧

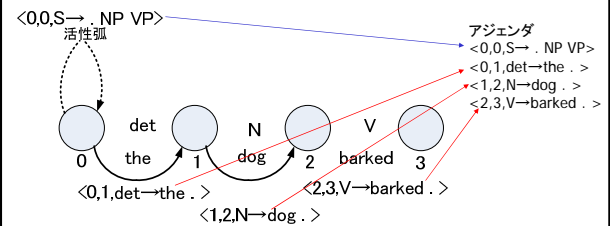


チャート法で使用する用語 3/3

- チャート
 - ノード, 弧の集合
- アジェンダ
 - チャートに追加すべき弧のリスト

トップダウンチャート法のアルゴリズム(1/2)

- 辞書規則の適用
 - 入力文の各単語 w_k について, 不活性弧 $\langle k, k+1, A \rightarrow w_k . \rangle$ をアジェンダに追加
- 活性弧 $\langle 0, 0, S \rightarrow \alpha \rangle$ をアジェンダの先頭に追加

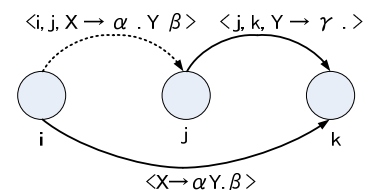


トップダウンチャート法のアルゴリズム(2/2)

- アジェンダが空になるまで以下の操作を繰り返す
 - 弧の選択
 - アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)
 - 弧の結合
 - arcが活性弧 $\langle i, j, X \rightarrow \alpha . Y \beta \rangle$ のとき,
 - arcの右にある不活性弧 $\langle j, k, Y \rightarrow \gamma . \rangle$ を探し, 結合する(次ページ)
 - arcが不活性弧 $\langle i, j, Y \rightarrow \gamma . \rangle$ のとき,
 - arcの左にある活性弧 $\langle k, i, X \rightarrow \alpha . Y \beta \rangle$ を探し, 結合する
 - 結合してできた新しい弧 $\langle i, k, X \rightarrow \alpha Y . \beta \rangle$ をアジェンダに追加
 - 新しい弧の提案
 - arcが活性弧 $\langle i, j, X \rightarrow \alpha . Y \beta \rangle$ のとき,
 - Y を左辺とする規則 $Y \rightarrow \gamma$ (辞書規則を除く)があれば, 新しい活性弧 $\langle j, j, Y \rightarrow \gamma . \rangle$ を作ってアジェンダに追加

トップダウンチャート法のアルゴリズム

- 弧の結合
 - arcが $\langle i, j, X \rightarrow \alpha . Y \beta \rangle$ の時
 - $\langle i, j, X \rightarrow \alpha . Y \beta \rangle + \langle j, k, Y \rightarrow \gamma . \rangle$
 - $\rightarrow \langle i, k, X \rightarrow \alpha Y . \beta \rangle$



- 不活性弧 $\langle 0, n, S \rightarrow \alpha . \rangle$ が生成できれば解析成功

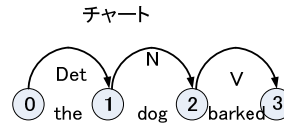
(トップダウン)チャート法を用いた構文解析例 (例文)

- 解析文
 - The dog barked.
- 文法
 - $S \rightarrow NP VP$
 - $NP \rightarrow Det N$
 - $VP \rightarrow V$
 - $VP \rightarrow V NP$
 - $Det \rightarrow the$
 - $N \rightarrow dog$
 - $V \rightarrow barked$

The dog barked. 1/27

辞書規則の適用
 入力文の各単語 w_k について、
 不活性弧 $\langle k, k+1, A \rightarrow w_k \dots \rangle$ をアジェンダに追加

- 文法
 - $S \rightarrow NP VP$
 - $NP \rightarrow det N$
 - $VP \rightarrow V$
 - $VP \rightarrow V NP$
 - $Det \rightarrow the$
 - $N \rightarrow dog$
 - $V \rightarrow barked$



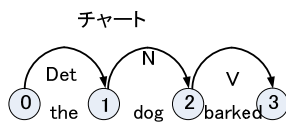
- アジェンダ
- $\langle 0, 1, Det \rightarrow the \dots \rangle$
 - $\langle 1, 2, N \rightarrow dog \dots \rangle$
 - $\langle 2, 3, V \rightarrow barked \dots \rangle$

辞書規則をアジェンダにpush

The dog barked. 2/27

活性弧 $\langle 0, 0, S \rightarrow \alpha \dots \rangle$ をアジェンダの先頭に追加

- 文法の一部)
- $S \rightarrow NP VP$
 - $NP \rightarrow Det N$
 - $VP \rightarrow V$
 - $VP \rightarrow V NP$



- アジェンダ
- $\langle 0, 0, S \rightarrow \dots NP VP \rangle$
 - $\langle 0, 1, Det \rightarrow the \dots \rangle$
 - $\langle 1, 2, N \rightarrow dog \dots \rangle$
 - $\langle 2, 3, V \rightarrow barked \dots \rangle$

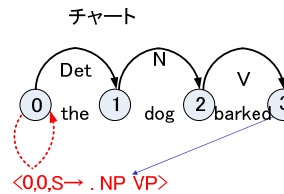
$\langle 0, 0, S \rightarrow \dots NP VP \rangle$ をアジェンダにpush

The dog barked. 3/27

弧の選択

アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

- 文法の一部)
- $S \rightarrow NP VP$
 - $NP \rightarrow Det N$
 - $VP \rightarrow V$
 - $VP \rightarrow V NP$



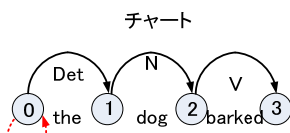
- アジェンダ
- $\langle 0, 0, S \rightarrow \dots NP VP \rangle$
 - $\langle 0, 1, Det \rightarrow the \dots \rangle$
 - $\langle 1, 2, N \rightarrow dog \dots \rangle$
 - $\langle 2, 3, V \rightarrow barked \dots \rangle$

新しい活性弧 $\langle 0, 0, S \rightarrow \dots NP VP \rangle$ をアジェンダからチャートにpop

The dog barked. 4/27

弧の結合
 arcが活性弧 $\langle i, j, X \rightarrow \alpha \dots Y \beta \dots \rangle$ のとき、
 arcの右にある不活性弧 $\langle j, k, Y \rightarrow \gamma \dots \rangle$ を探し、結合する
 結合してできた新しい弧 $\langle i, k, X \rightarrow \alpha \dots Y \beta \dots \rangle$ をアジェンダに追加

- 文法の一部)
- $S \rightarrow NP VP$
 - $NP \rightarrow Det N$
 - $VP \rightarrow V$
 - $VP \rightarrow V NP$



- アジェンダ
- $\langle 0, 1, Det \rightarrow the \dots \rangle$
 - $\langle 1, 2, N \rightarrow dog \dots \rangle$
 - $\langle 2, 3, V \rightarrow barked \dots \rangle$

$\langle 0, 0, S \rightarrow \dots NP VP \rangle$

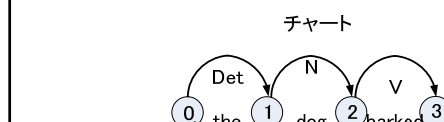
該当無し→何もしない

The dog barked. 5/27

新しい弧の提案

arcが活性弧 $\langle i, j, X \rightarrow \alpha \dots Y \beta \dots \rangle$ のとき、
 Y を左辺とする規則 $Y \rightarrow \gamma$ (辞書規則を除く)があれば、
 新しい活性弧 $\langle j, j, Y \rightarrow \gamma \dots \rangle$ をアジェンダに追加

- 文法の一部)
- $S \rightarrow NP VP$
 - $NP \rightarrow Det N$
 - $VP \rightarrow V$
 - $VP \rightarrow V NP$



- アジェンダ
- $\langle 0, 0, NP \rightarrow \dots Det N \rangle$
 - $\langle 0, 1, Det \rightarrow the \dots \rangle$
 - $\langle 1, 2, N \rightarrow dog \dots \rangle$
 - $\langle 2, 3, V \rightarrow barked \dots \rangle$

$\langle 0, 0, S \rightarrow \dots NP VP \rangle$ $\langle 0, 0, NP \rightarrow \dots Det N \rangle$

$\langle 0, 0, NP \rightarrow \dots Det N \rangle$ をアジェンダに追加

The dog barked. 6/27

文法の一部)

- S → NP VP
- NP → Det N
- VP → V
- VP → V NP

弧の選択
アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

アジェンダ

- <0,0,NP → . Det N>
- <0,1,Det → the .>
- <1,2,N → dog .>
- <2,3,V → barked .>

アジェンダから<0,0,NP → . Det N>をチャートにpop

The dog barked. 7/27

文法の一部)

- S → NP VP
- NP → Det N
- VP → V
- VP → V NP

弧の結合
arcが活性弧<i,j,X → α.Yβ>のとき、
arcの右にある不活性弧<j,k,Y → γ.>を探し、結合する
結合してできた新しい弧<i,k,X → αY.β>をアジェンダに追加

アジェンダ

- <0,1,Det → the .>
- <1,2,N → dog .>
- <2,3,V → barked .>

アジェンダ

- <0,1,NP → Det . N>
- <1,2,N → dog .>
- <2,3,V → barked .>

<0,0,NP → . Det N>と<0,1,Det → the .>を結合して<NP → Det . N>を得る。
<NP → Det . N>をアジェンダにpush

The dog barked. 8/27

文法の一部)

- S → NP VP
- NP → Det N
- VP → V
- VP → V NP

新しい弧の提案
arcが活性弧<i,j,X → α.Yβ>のとき、
Yを左辺とする規則Y → γ (辞書規則を除く)があれば、
新しい活性弧<j,j,Y → γ.>を作ってアジェンダに追加

アジェンダ

- <0,1,NP → Det . N>
- <1,2,N → dog .>
- <2,3,V → barked .>

アジェンダ

- <NP → Det . N>
- <NP → . Det N>

規則Y → γがない → 何もしない

The dog barked. 9/27

文法の一部)

- S → NP VP
- NP → Det N
- VP → V
- VP → V NP

弧の選択
アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

アジェンダ

- <0,1,NP → Det . N>
- <1,2,N → dog .>
- <2,3,V → barked .>

アジェンダ

- <0,1,NP → Det . N>
- <1,2,N → dog .>
- <2,3,V → barked .>

<0,1,NP → Det . N>をアジェンダに追加

The dog barked. 10/27

文法の一部)

- S → NP VP
- NP → Det N
- VP → V
- VP → V NP

弧の結合
arcが活性弧<i,j,X → α.Yβ>のとき、
arcの右にある不活性弧<j,k,Y → γ.>を探し、結合する
結合してできた新しい弧<i,k,X → αY.β>をアジェンダに追加

アジェンダ

- <1,2,N → dog .>
- <2,3,V → barked .>

アジェンダ

- <NP → Det . N>
- <NP → Det N .>

アジェンダ

- <0,2,NP → Det N .>
- <2,3,V → barked .>

<NP → Det . N>と<N → dog .>を結合して<NP → Det N .>を得る。
<NP → Det N .>をアジェンダにpush

The dog barked. 11/27

文法の一部)

- S → NP VP
- NP → Det N
- VP → V
- VP → V NP

新しい弧の提案
arcが活性弧<i,j,X → α.Yβ>のとき、
Yを左辺とする規則Y → γ (辞書規則を除く)があれば、
新しい活性弧<j,j,Y → γ.>を作ってアジェンダに追加

アジェンダ

- <0,2,NP → Det N .>
- <2,3,V → barked .>

アジェンダ

- <NP → Det . N>
- <NP → Det N .>

規則Y → γがない → 何もしない

The dog barked. 12/27

文法の一部)
 • S→NP VP
 • NP→Det N
 • VP→V
 • VP→V NP

弧の選択
 アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

チャート

アジェンダ
 <0,2,NP→Det N . >
 <2,3,V→barked . >

<0,2,NP→Det N . >をアジェンダからpopしてチャートに追加

The dog barked. 13/27

文法の一部)
 • S→NP VP
 • NP→Det N
 • VP→V
 • VP→V NP

弧の結合
 arcが不活性弧<i,j,Y→γ.>のとき、
 arcの左にある活性弧<k,i,X→α.Yβ>を探し、結合する
 結合してできた新しい弧<i,k,X→α.Y.β>をアジェンダに追加

チャート

アジェンダ
 <0,2,S→NP . VP>
 <2,3,V→barked . >

<NP→Det N . >と<S→ . NP VP>を結合して<S→ NP . VP>を得る。
 <S→ NP . VP>をアジェンダにpush

The dog barked. 14/27

文法の一部)
 • S→NP VP
 • NP→Det N
 • VP→V
 • VP→V NP

新しい弧の提案
 arcが活性弧<i,j,X→α.Yβ>のとき、
 Yを左辺とする規則Y→γ(辞書規則を除く)があれば、新
 しい活性弧<j,j,Y→γ>を作ってアジェンダに追加

チャート

アジェンダ
 <0,2,S→NP . VP>
 <2,3,V→barked . >

Arcは不活性弧 → 何もしない

The dog barked. 15/27

文法の一部)
 • S→NP VP
 • NP→Det N
 • VP→V
 • VP→V NP

弧の選択
 アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

チャート

アジェンダ
 <0,2,S→NP . VP>
 <2,3,V→barked . >

<0,2,S→NP . VP>をアジェンダからpopしてチャートに追加

The dog barked. 16/27

文法の一部)
 • S→NP VP
 • NP→Det N
 • VP→V
 • VP→V NP

弧の結合
 arcが活性弧<i,j,X→α.Yβ>のとき、
 arcの右にある不活性弧<j,k,Y→γ.>を探し、結合する
 結合してできた新しい弧<i,k,X→α.Y.β>をアジェンダに追加

チャート

アジェンダ
 <2,3,V→barked . >

Arcの右にないので何もしない

The dog barked. 17/27

文法の一部)
 • S→NP VP
 • NP→Det N
 • VP→V
 • VP→V NP

新しい弧の提案
 arcが活性弧<i,j,X→α.Yβ>のとき、
 Yを左辺とする規則Y→γ(辞書規則を除く)があれば、
 新しい活性弧<j,j,Y→γ>を作ってアジェンダに追加

チャート

アジェンダ
 <2,2,VP→ . V >
 <2,3,V→barked . >

新しい活性弧<2,2,VP→ . V>をアジェンダにpush

The dog barked. 18/27

文法の一部)

- S→NP VP
- NP→Det N
- VP→V
- VP→V NP

弧の選択
アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

チャート

アジェンダ
<2,2,VP->.V>
<2,3,V->barked.>

<0,0,S->.NP VP> <0,2,S->NP.VP> <2,2,VP->.V>

<2,2,VP->.V>をアジェンダからpopしてチャートに追加

The dog barked. 19/27

文法の一部)

- S→NP VP
- NP→Det N
- VP→V
- VP→V NP

弧の結合
arcが活性弧<ij,X->α.Yβ>のとき、
arcの右にある不活性弧<jk,Y->γ.>を探し、結合する
結合してできた新しい弧<i,k,X->α.Y.β>をアジェンダに追加
チャート

チャート

アジェンダ
<2,3,V->barked.>
<2,3,VP->.V.>

<0,0,S->.NP VP> <0,2,S->NP.VP> <2,3,VP->.V.> <2,2,VP->.V.>

<VP->.V>+<V->barked.>=<VP->.V.>

アジェンダ
<2,3,VP->.V.>

The dog barked. 20/27

文法の一部)

- S→NP VP
- NP→Det N
- VP→V
- VP→V NP

新しい弧の提案
arcが活性弧<ij,X->α.Yβ>のとき、
Yを左辺とする規則Y->γ(辞書規則を除く)があれば、
新しい活性弧<jj,Y->γ.>を作ってアジェンダに追加

チャート

アジェンダ
<2,3,VP->.V.>
<2,2,VP->.V.>

<0,0,S->.NP VP> <0,2,S->NP.VP> <2,3,VP->.V.> <2,2,VP->.V.>

Y->γがないので何もしない

The dog barked. 21/27

文法の一部)

- S→NP VP
- NP→Det N
- VP→V
- VP→V NP

弧の選択
アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

チャート

アジェンダ
<2,3,VP->.V.>

<0,0,S->.NP VP> <0,2,S->NP.VP> <2,3,VP->.V.> <2,2,VP->.V.>

<2,3,VP->.V.>をアジェンダからpopしてチャートに追加

The dog barked. 22/27

文法の一部)

- S→NP VP
- NP→Det N
- VP→V
- VP→V NP

弧の結合
arcが不活性弧<ij,Y->γ.>のとき、
arcの左にある活性弧<ki,X->α.Yβ>を探し、結合する
結合してできた新しい弧<i,k,X->α.Y.β>をアジェンダに追加

チャート

アジェンダ
<0,3,S->NP VP.>
<2,3,VP->.V.>

<0,0,S->.NP VP> <0,2,S->NP.VP> <2,3,VP->.V.> <2,2,VP->.V.>

<S->NP.VP>+<VP->.V.>を結合して<S->NP VP.>を得る

The dog barked. 23/27

文法の一部)

- S→NP VP
- NP→Det N
- VP→V
- VP→V NP

新しい弧の提案
arcが活性弧<ij,X->α.Yβ>のとき、
Yを左辺とする規則Y->γ(辞書規則を除く)があれば、
新しい活性弧<jj,Y->γ.>を作ってアジェンダに追加

チャート

アジェンダ
<0,3,S->NP VP.>

<0,0,S->.NP VP> <0,2,S->NP.VP> <2,3,VP->.V.> <2,2,VP->.V.>

arcは不活性弧なので何もしない

The dog barked. 24/27

文法の一部)
 . S→NP VP
 . NP→Det N
 . VP→V
 . VP→V NP

弧の選択
 アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

チャート

アジェンダ
 <0,3,S->NP VP.>

<0,0,S->. NP VP> <0,2,S->NP . VP> <2,2,VP->. V> <2,3,VP->V .>

<0,3,S->NP VP .>をアジェンダからpopしてチャートに追加

The dog barked. 25/27

文法の一部)
 . S→NP VP
 . NP→Det N
 . VP→V
 . VP→V NP

弧の結合
 arcが不活性弧<i,j,Y->γ.>のとき、
 arcの左にある活性弧<k,i,X->α.Yβ>を探し、結合する
 結合してできた新しい弧<i,k,X->α.Y.β>をアジェンダに追加

チャート

アジェンダ

<0,0,S->. NP VP> <0,2,S->NP . VP> <2,2,VP->. V> <2,3,VP->V .>

<k,i,X->α.Yβ>がないので何もしない

The dog barked. 26/27

文法の一部)
 . S→NP VP
 . NP→Det N
 . VP→V
 . VP→V NP

新しい弧の提案
 arcが活性弧<i,j,X->α.Yβ>のとき、
 Yを左辺とする規則Y->γ(辞書規則を除く)があれば、新
 しい活性弧<j,j,Y->γ>を作ってアジェンダに追加

チャート

アジェンダ

<0,0,S->. NP VP> <0,2,S->NP . VP> <2,2,VP->. V> <2,3,VP->V .>

Arcは不活性弧なので何もしない

The dog barked. 27/27

文法の一部)
 . S→NP VP
 . NP→Det N
 . VP→V
 . VP→V NP

弧の選択
 アジェンダから弧を1個選びチャートに追加(選んだ弧=arc)

チャート

アジェンダ
 (空)

<0,0,S->. NP VP> <0,2,S->NP . VP> <2,2,VP->. V> <2,3,VP->V .>

アジェンダになにも無いので処理終了
 不活性弧<0,n,S->α.>を生成できたので解析成功!

構文木の復元

- 弧に履歴を残す。
 - 弧に識別番号をつける
 - 右辺がどの不活性弧によって構成されるかを記録
- 不活性弧の履歴をたどれば構文木が復元できる
- 得られる構文木の例
 - 番号は不活性弧の番号

チャート法の特徴

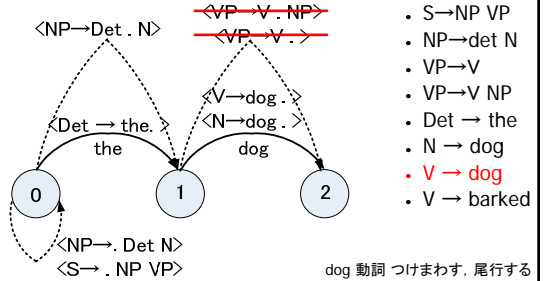
- 任意の文脈自由文法規則が取り扱い可能
 - A→BCDも、A→bCもOK
- 4種類の方式
 - トップダウンとボトムアップ
 - 縦型探索と横型探索
- 文法の予測能力が使える
 - 無駄な弧を生成しないので効率が良い(トップダウンチャート法)
- 広く使われている

縦型探索と横型探索

- 縦型探索
 - 1つの解の候補の解析を優先的に進める
 - 文が文法によって生成できるかだけを調べるときに便利
- 横型探索
 - 全ての解の候補の解析を並列に進める
 - ビームサーチが使える
- チャート法では両方も可能
- アジェンダをスタック(LIFO)にしたときは縦型探索
- アジェンダをキュー(FIFO)にしたときは横型探索

文法の予測能力

- 無駄な弧は生成されない
- 文法によってDetの後にはVが現れないことが予想されている



動的計画法(Dynamic Programming)

- 部分問題の解をより大きな問題を解くために利用
- 同じ問題を2度解かなくても済むように解を格納
- 最適解に利用できない部分問題は省略する
- アルゴリズムの例
 - CYK法(構文解析)
 - ダイクストラ法(最短経路問題)
 - DPマッチング(パターンマッチング DNAの解析にも利用)
 - DPを使った解法(ナップサック問題)
 - ビタビアルゴリズム(音声認識など)

ダイクストラ法

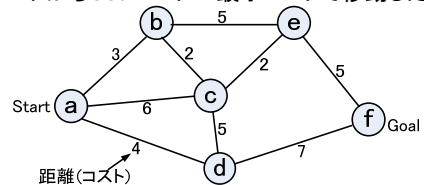
- 動的計画法を最短経路問題に適用
- 最適経路中の部分経路もまた最適経路になっている

身近な最短経路問題

- 道路の経路探索(カーナビなど)

ダイクストラ法(最短経路問題用アルゴリズム)

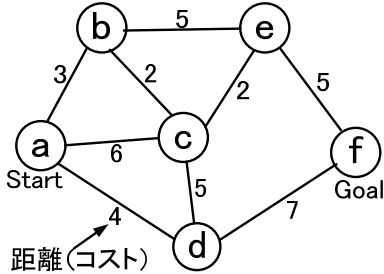
- StartノードからGoalノードへ最小コストで移動したい



- a-e, a-dなどの最短距離をa-fの最短距離を見つけるために利用
- 部分問題の解をより大きな問題を解くために利用
- 同じ問題を2度解かなくても済むように解を格納

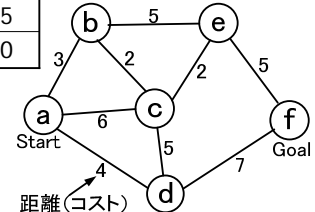
ダイクストラ法(最短経路問題用アルゴリズム)

- StartノードからGoalノードへ最小コストで移動したい



隣接行列(コスト付き)

	a	b	c	d	e	f
a	0	3	6	4	∞	∞
b	3	0	2	∞	5	∞
c	6	2	0	5	2	∞
d	4	-	5	0	∞	7
e	∞	5	2	∞	0	5
f	∞	∞	∞	7	5	0

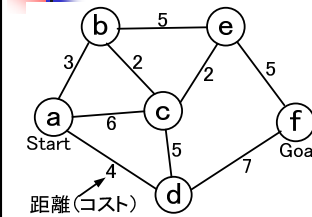


ダイクストラ法 アルゴリズム

- 初期化:
 - コスト表: 各ノードとスタートノード間のコストをコスト表に書き出す(スタートノードと隣接していないノードのコストは∞, スタートノードのコストは0とする)
 - 親ノード表: 各ノードの直前のノードとしてスタートノードを親ノード表に書き出す
- 未確定ノードが無くなるまで以下の処理を繰り返す.
 - 未確定ノードのうち, 最小コストのノードを見つけ確定ノードとする
 - 新しい確定ノードから隣接する未確定ノードを探す
 - 隣接する未確定ノードに対して「確定ノードまでのコスト+確定ノードから未確定ノードまでのコスト」を計算する. 計算結果がそのノードのコスト表の値よりも小さければ
 - コスト表: 計算結果を未確定ノードまでのコストとする
 - 親ノード表: 新しい確定ノードを未確定ノードの親ノードとする

ダイクストラ法 動作例 1/20

コスト行列の作成

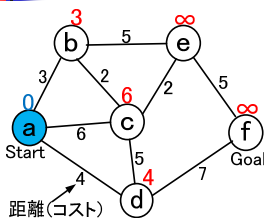


	a	b	c	d	e	f
a	0	3	6	4	∞	∞
b	3	0	2	∞	5	∞
c	6	2	0	5	2	∞
d	4	∞	5	0	∞	7
e	∞	5	2	∞	0	5
f	∞	∞	∞	7	5	0

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 2/20

各ノードのStartノードからのコスト, 親ノードを調べる
aのコストと親ノードは確定

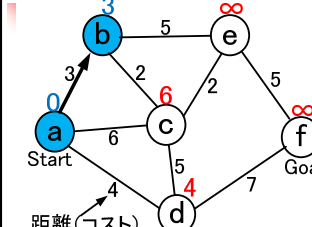


ノード	コスト	親ノード
a	0	a
b	3	a
c	6	a
d	4	a
e	∞	a
f	∞	a

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 3/20

未確定ノードのうち, 最小コストのノードを選び確定リストとする
→ bが選ばれる



ノード	コスト	親ノード
a	0	a
b	3	a
c	6	a
d	4	a
e	∞	a
f	∞	a

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 4/20

新たに確定ノードになったbに隣接する未確定ノードを探す:c,e

ノード	コスト	親ノード
a	0	a
b	3	a
c	6	a
d	4	a
e	∞	a
f	∞	a

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

ダイクストラ法 動作例 5/20

隣接する未確定ノードごとに「bまでのコスト+bからのコスト」を計算し、今までのコストより小さい場合はコストを書き換える

ノード	コスト	親ノード
a	0	a
b	3	a
c	5 > 6	a → b
d	4	a
e	∞ > 8	a → b
f	∞	a

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

ダイクストラ法 動作例 6/20

eの親ノードとcの親ノードをbに書き換える

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	8	b
f	∞	a

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

ダイクストラ法 動作例 7/20

未確定ノードのうち、最小コストのノードを選ぶ→dが選ばれる

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	8	b
f	∞	a

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

ダイクストラ法 動作例 8/20

新たに確定ノードになったdに隣接する未確定ノードを探す:c,f

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	8	b
f	∞	a

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

ダイクストラ法 動作例 9/20

隣接する未確定ノードごとに「dまでのコスト+dからのコスト」を計算し、今までのコストより小さい場合はコストを書き換える

ノード	コスト	親ノード
a	0	a
b	3	a
c	5 < 9	b
d	4	a
e	8	b
f	∞ > 11	a → d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)

ダイクストラ法 動作例 10/20

fの親ノードをdに書き換える

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	8	b
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 11/20

未確定ノードのうち、最小コストのノードを選ぶ→cが選ばれる

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	8	b
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 12/20

新たに確定ノードになったcに隣接する未確定ノードを探す:e

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	8	b
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 13/20

隣接する未確定ノードeについて「cまでのコスト+cからのコスト」を計算した結果今までのコストより小さくなったのでコストを書き換え

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	8 > 7	b → c
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 14/20

eの親ノードをcに書き換える

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	7	c
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 15/20

未確定ノードのうち、最小コストのノードを選ぶ→eが選ばれる

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	7	c
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク
- 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 16/20

新たに確定ノードになったeに隣接する未確定ノードを探す:f

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	7	c
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 17/20

隣接する未確定ノードfについて「eまでのコスト+eからのコスト」を計算した結果今までのコストの方が小さいのでコストは書き換えない

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	7	c
f	11 < 12	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 18/20

fの親ノードは更新されない

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	7	c
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 19/20

未確定ノードのうち、最小コストのノードを選ぶ→fが選ばれる

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	7	c
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 動作例 20/20

新たに確定ノードになったfに隣接する未確定ノードを探す:無し

終了

ノード	コスト	親ノード
a	0	a
b	3	a
c	5	b
d	4	a
e	7	c
f	11	d

- Startからの最短経路が確定していないノード
- Startからの最短経路再計算中のノード
- Startからの最短経路が確定したノード
- 7 Startからの最短距離候補(未確定)
- 7 Startからの最短距離(確定済)
- 確定済ノードからのアーク 最短経路再計算用に使用
- StartからGoalまでの最短経路

ダイクストラ法 アルゴリズム

- 初期化:
 - コスト表: 各ノードとスタートノード間のコストをコスト表に書き出す(スタートノードと隣接していないノードのコストは∞, スタートノードのコストは0とする)
 - 親ノード表: 各ノードの直前のノードとしてスタートノードを親ノード表に書き出す
- 未確定ノードが無くなるまで以下の処理を繰り返す.
 - 未確定ノードのうち、最小コストのノードを見つけ確定ノードとする
 - 新しい確定ノードから隣接する未確定ノードを探す
 - 隣接する未確定ノードに対して「確定ノードまでのコスト+確定ノードから未確定ノードまでのコスト」を計算する. 計算結果がそのノードのコスト表の値よりも小さければ
 - コスト表: 計算結果を未確定ノードまでのコストとする
 - 親ノード表: 新しい確定ノードを未確定ノードの親ノードとする

ダイクストラ法の特徴

- 最短経路の見つけ方
 - ゴールノードから「どこから来たのか」調べ、さかのぼる(距離更新時に直前のノードを記述しておく).
- マイナスのコストを持つエッジは扱えない.
- 特定のノードからの最短距離およびその経路が全てのノードに対して求まる.

DPマッチング

(例:文字列の照合)

- 2つの文字列がどのくらい似ているかを調べる.
 - Yamanashi は kamonohashiとtakahashi
- 音声認識にも使える
 - 音声を文字列に変換した後,登録単語と比較
 - (現在主流の)HMM(Hidden Markov Model)に拡張可能
- DNAの比較にも使える
 - A(アデニン), G(グアニン), C(シトシン), T(チミン)の並び方の比較
 - ACTGAGCATTとCTGGACTACGの比較

本日のまとめ

- 構文解析
 - チャート法
 - 解析例
 - アルゴリズム
- 動的計画法(最短距離探索)
 - ダイクストラ法
 - 解析例
 - アルゴリズム