

アルゴリズムとデータ構造III 9回目: 11月27日

中間試験

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

1

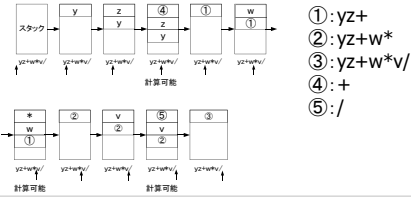
問題1. 中置記法 ↔ 後置記法 (10点)

- 1) 次の中置記法を後置記法に直せ
 $((y \times y) + (y \times (y + y)))$
 $\rightarrow yy \times yyy + \times +$
- 2) 次の後置記法を中置記法に直せ
 $yyy + \times yy + \times$
 $\rightarrow (y \times (y + y)) \times (y + y)$

2

問題2. スタック (10点)

- 下の図は後置記法の式 $yz+w*v/$ のスタックを使用した計算過程である。図中の①, ②, ③, ④, ⑤にはそれぞれ何が入るか答えよ。



3

問題3. 文脈自由文法 (10点)

- 文脈自由文法と文脈依存文法の違いを200文字以内で説明せよ。
- 文脈依存文法の生成規則は $uav \rightarrow u\beta v$ (a は非終端記号, β, u, v は終端または非終端記号列)の形で表される。これは非終端記号 a の前後の記号 u, v により a から β の導出が制限される事を意味する。 $uav \rightarrow u\beta v$ で u と v が ϵ であるとき $a \rightarrow \beta$ となり、前後の記号(文脈)は a から β の導出に影響を与えない。そのため $a \rightarrow \beta$ のような生成規則だけを持つ文法を前後の文脈(記号)に対して影響を受けないという意味で文脈自由文法と呼ぶ。

4

問題4. 構文解析 (6点)

- 構文解析の代表的手法を3つ挙げよ。
- CKY法
- チャート法
- LR法
- Earley法(アーリー法)など

5

問題5. CKY法 1) (12点)

- 下の図は「Nana met with Ramos.」を構文解析中のCKY表である。
- 1) 図中の①, ②, ③, ④, ⑤, ⑥には何が入るか答えよ。

	Nana	met	with	Ramos	
Nana	N → Nana	①	④	⑥	
met		V → met	②	⑤	
with			P → with	③	
Ramos				N → Ramos	

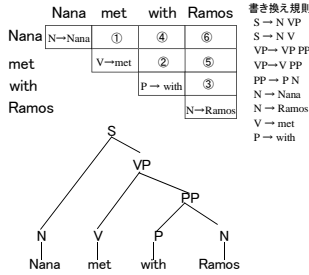
書き換え規則
 $S \rightarrow NVP$
 $S \rightarrow NV$
 $VP \rightarrow VPP$
 $VP \rightarrow VP$
 $PP \rightarrow PN$
 $N \rightarrow Nana$
 $N \rightarrow Ramos$
 $V \rightarrow met$
 $P \rightarrow with$

①: $S \rightarrow NV$
 ②: 該当なし
 ③: $PP \rightarrow PN$
 ④: 該当なし
 ⑤: $VP \rightarrow VPP$
 ⑥: $S \rightarrow NVP$

6

問題6. CKY法 2) (2点)

- 下の図は「Nana met with Ramos.」を構文解析中のCKY表である。
- 2) CKY表から得られる「Nana met with Ramos.」の構文木を描け。



問題6. トップダウンチャート法 (10点)

- CKY法と比較したときのトップダウンチャート法の特徴を簡潔に説明せよ。
- 文脈自由文法で書かれた文を構文解析するための代表的な手法
- アークとノードを使ったグラフで表される
- CKY法ではチョムスキーの標準形以外は扱えないが、チャート法では $X \rightarrow ABC$ のような変換規則も扱うことができる。
- 簡単な予測を使うことが出来るため、CKY法より効率が良い

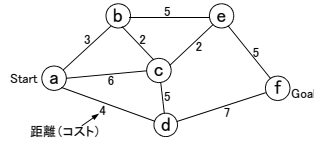
問題7. 動的計画法 (10点)

- 動的計画法を200文字以内で説明せよ。
- 解くのに時間のかかる問題を、複数の部分問題に分割することで効率的に解くアルゴリズム
- 動的計画法の適用例として、最短経路検索のためのダイクストラ法、パターンマッチングのためのDPマッチングがある。

問題8. ダイクストラ法 (10点)

- 下の図でStartからGoalまでの最短ルートを、ダイクストラ法を用いて求めよ。どのような手順で処理を行うかを簡潔に説明せよ。

最短経路: a-d-f
最短距離: 11



- Startノードを確定ノードとする
- 確定ノードに隣接する各ノードまでの距離を求める
 - 既に別ルートで距離が計算されているノードについて新しく求めた距離の方が短ければ距離を更新する
 - 確定ノードに隣接するノードの内スタートノードからの距離が一番短いノードを確定ノードとする。
 - ゴールのノードが確定ノードになるまで1. ~3. を繰り返す。

問題9. DPマッチング (10点)

- abcdeとabbdeの類似度を、DPマッチングを用いて計算しなさい。計算方法を簡潔に説明しなさい。
- 但し、不一致ペナルティ=3, 挿入ペナルティ=1, 脱落ペナルティ=1とする。

通行ペナルティ積算表

	a	b	c	d	e
a	0	4	8	12	16
b	4	0	4	8	12
b	8	1	3	7	11
d	12	5	4	3	7
e	16	9	8	7	3

通行ペナルティ積算表の空欄を左上から右下へ積算コストを記入していくことにより処理を進める。一般に1つの空欄を埋めるにはその空欄の上隣経路の積算コスト、斜め上経路の積算コスト、左隣経路の積算コストの内最小コストを選択する。右下の空欄に記入した積算コストが2つの文字列の類似度となる。

問題10. A*アルゴリズム (10点)

- A*アルゴリズムとダイクストラ法の類似点と相違点を300文字以内で説明せよ。
- 類似点
 - どちらも最短経路問題を解くのに使われる。
 - マイナスのコストをもつ辺を含む経路の最短経路は解くことが出来ない。
- 相違点
 - ダイクストラ法はスタートから各節点までの移動コストを利用して最短経路を求めるのに対してA*アルゴリズムはスタートから各節点までの移動コストと節点からゴールまでの予測コスト ($0 \leq \text{予測コスト} \leq \text{実際のコスト}$) を利用する。
 - A*アルゴリズムは予測コストを利用するためダイクストラ法よりも効率よく問題を解くことが出来る。各節点からゴールまでの予測コストがすべて0である場合、ダイクストラ法のアルゴリズムと同じになる。