

大阪大学大学院情報科学研究科 2014 年度 過去問解答

渡邊 航平

平成 27 年 7 月 9 日

1 【必須問題】 アルゴリズムとプログラミング

1.1

5 6 10
2 5 5 7 13 18 20

1.2

front : 3, rear : 5

1.3

1.4

insert : $O(n)$

15 行目から 19 行目は二分探索をしているので, $O(\log n)$. また, 24 行目は $i - p$ 回だけ入替操作を行っている. これは n に比例して大きくなるので, $O(n)$. したがって, $O(\log n) + O(n) = O(n)$.

delete : $O(1)$

繰り返しもなく単純な代入しかないため, $O(1)$ となる.

1.5

rear は減算することなく常に加算されていくため, 1 回の実行でデータ列のサイズである 20 回より多くデータを挿入しようとする, 9 行目の条件文よりプログラムが停止し, 挿入が失敗する.

1.6

2 【必須問題】 計算機システムとシステムプログラム

2.1

2.1.1

めんどい

2.1.2

調べて

2.1.3

W, Y_i を入力し, Y'_i を出力とした XOR

2.1.4

被演算列 Y で, 上記の XOR を一回はさみ, 桁上げ入力にも W を接続させた回路となる.

2.2

2.2.1

a : アンロック, b : ロック, c : 割り込み

2.2.2

共有資源 Z がアンロック状態を確認し, プロセス X が共有資源 Z のセクションを開始する. ここで割り込みで, プロセス Y も共有資源 Z を使用しようとする. しかしまだロック状態にはなっていないのでプロセス Y も共有資源 Z を使用可能である. このように, テストとセットを不可分操作にしない場合, プロセス X とプロセス Y が同時に共有資源 Z を使用する状態に陥る可能性がある.

3 【選択問題】 離散構造

4 【選択問題】 計算理論

4.1

4.1.1

各桁の和が3の倍数であればよい。以下の図1のようになる。

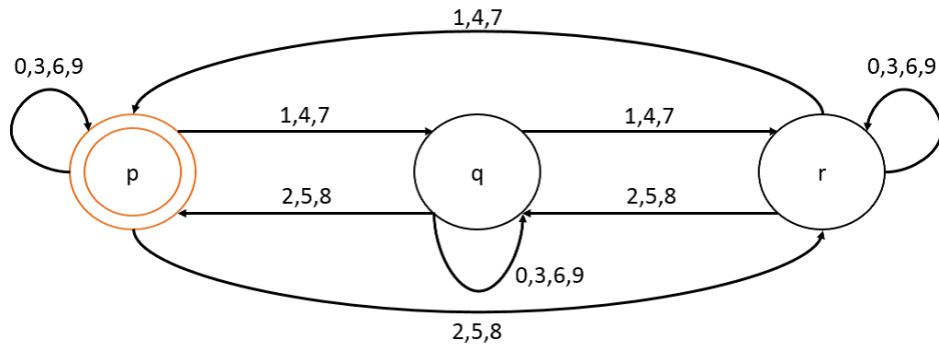


図 1: 3 の倍数

4.1.2

末尾が0か5であればよい。以下の図2のようになる。

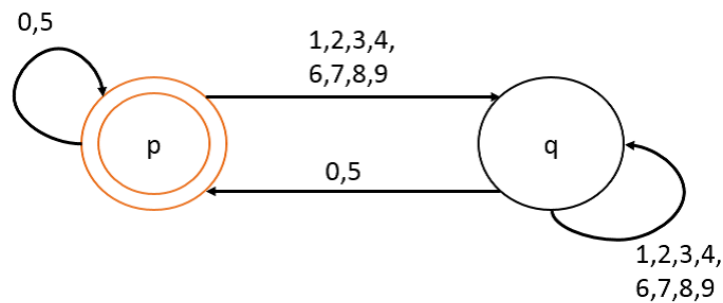


図 2: 5 の倍数

4.1.3

15の倍数が n 桁の入力だとすると、 $n-1$ 桁まで3の倍数判定、 n 桁目で5の倍数判定を行う。以下の図3のようになる。

4.2

4.2.1

$$G_1 = \{\{S\}, \{C\}, \{S \rightarrow cS, S \rightarrow c\}, S\}$$

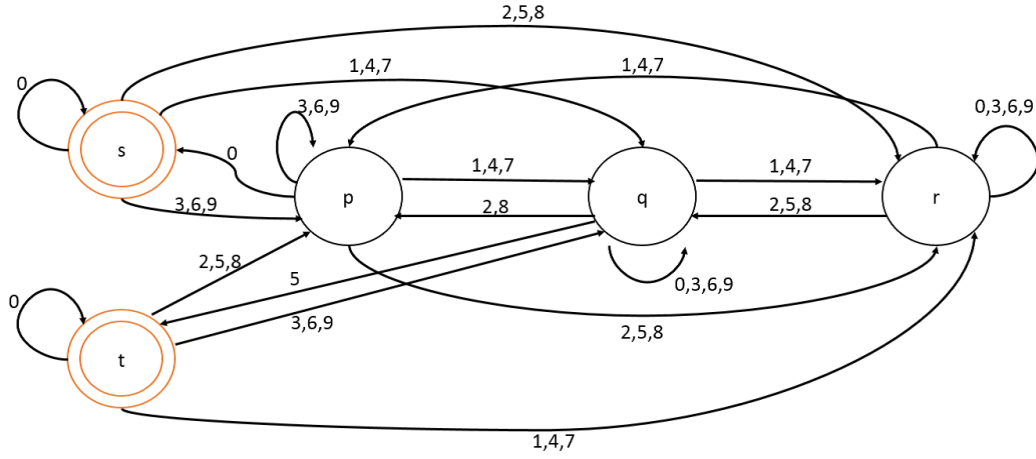


図 3: 15 の倍数

4.2.2

$L_{2A} = \{a^n b^n | n \geq 1\}, L_{2B} = \{c^m | m \geq 1\}$ とする.

L_{2A} を生成する文法は $S \rightarrow aSb \mid ab$ であり, プッシュダウン・オートマトン

$$M = (\{q_0, q_1, q_f\}, \{a, b\}, \{a, z\}, \delta, q_0, \{q_f\})$$

に受容される. ここで δ は以下のように定義される.

$$\delta(q_0, a, z) = (q_0, a)$$

$$\delta(q_0, a, a) = (q_0, a)$$

$$\delta(q_0, b, a) = (q_1, x)$$

$$\delta(q_1, b, a) = (q_1, x)$$

$$\delta(q_1, b, z) = (q_f, z)$$

ここで z は初期スタック記号, x はポップ動作を意味する. よって, L_{2A} は文脈自由文法となる.

したがって, L_{2A} と L_{2B} の連接である $L_2 = L_{2A}L_{2B} = \{a^n b^n c^m | n \geq 1, m \geq 1\}$ も文脈自由文法となる.

L_3 の場合も, 対称性より同様の処理を行うことで証明可能である.

4.2.3

L_4 に関して, 以下の等式が成立する.

$$L_4 = L_2 L_3$$

ここで, (2-2) と補題より, L_2, L_3 は文脈自由文法だが, その積集合である L_4 は文脈自由文法とはならない. したがって, 文脈自由文法全体の集合が積演算については閉じていない.

5 【選択問題】 ネットワーク

5.1

5.1.1

情報源記号が互いに独立に発生する情報源（過去の出力の影響を受けない情報源）

5.1.2

(あ)g, (い)c, (う)e, (え)b

5.1.3

5.2

5.2.1

(あ)e, (い)g, (う)f, (え)g, (お)e

5.2.2

負荷が大きくなると、CSMA/CD 方式では衝突により遅延が急激に増加する。一方、トークンパッシング方式ではトークンの巡回により回線へのアクセスが制御されるので、遅延の増大を抑えられるためである。

5.2.3

512 ビットのスロット時間の整数倍にバックオフ時間を設定する。その際、衝突回数に応じてバックオフ時間を増やしていく。衝突回数を m , スロットタイムを S とすると,

$$T = nS, \quad 0 < n \leq 2^k, \quad k = \min(m, 10)$$

となる。すなわち、衝突回数 10 回までバックオフの長さを 2 のべき乗で増やしていくことで、衝突したフレーム同士が再度衝突する可能性を小さくしている。