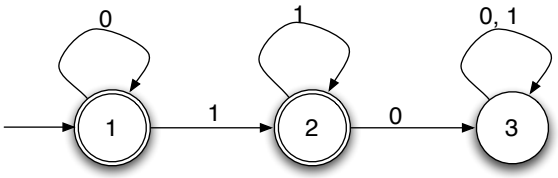
平成 19 年度「コンパイラ」定期試験問題

國島丈生

2007-07-18

1　正則表現と有限オートマトン

1. 以下の言語を表す正則表現を示せ。(各 5 点)
   (a) アルファベット $\{0,1\}$ 上の文字列で、01 または 10 から始まるものすべてからなる言語。
   (b) Unix コマンドのオプションとして使える文字列すべてからなる言語。Unix コマンドのオプションは、(1) − (マイナス記号)1 つに続けて英大文字・英小文字・数字が 1 文字来るもの（例: `-b`, `-9`）(2) − (マイナス記号)2 つに続けて英小文字が複数個並ぶもの（例: `--version`）の 2 通りのみ考えればよい。
2. 前問の言語を受理する有限オートマトンを示せ。(各 5 点)
3. 以下の有限オートマトンが受理する言語を正則表現で表せ。(10 点)



2　文脈自由文法

次の文脈自由文法について、以下の問に答えよ。

$$S \to SbS \mid ScS \mid a$$

1. 文字列 $abaca$ に対する解析木をすべて示せ。(10 点)
2. 1 で示した解析木それぞれに対応する最左導出を示せ。(10 点)

3　LL(1) 文法

次の文脈自由文法について、以下の問に答えよ。

$$\begin{aligned} S &\to D \\ A &\to aA \mid \epsilon \\ B &\to bAC \mid AcD \\ C &\to bC \mid Ac \\ D &\to BA \mid d \end{aligned}$$

1. 各非終端記号について FIRST と FOLLOW を計算せよ。(20 点)
2. この文法は LL(1) 文法か、判定せよ。(10 点)

4 翻訳スキーム

次の翻訳スキームについて、以下の問に答えよ。ただし、生成規則中で同じ非終端記号が複数回出現するとき、これらを区別するために添字をつけることがある。

$$
\begin{aligned}
T &\to H : M : S \, \{T.val = 3600 * H.val + 60 * M.val + S.val\}^{[1]} \\
H &\to D_1 D_2 \, \{H.val = 10 * D_1.val + D_2.val\}^{[2]} \\
M &\to D_1 D_2 \, \{M.val = 10 * D_1.val + D_2.val\}^{[3]} \\
S &\to D_1 D_2 \, \{S.val = 10 * D_1.val + D_2.val\}^{[4]} \\
D &\to 0 \, \{D.val = 0\}^{[5]} \\
D &\to 1 \, \{D.val = 1\}^{[6]} \\
D &\to 2 \, \{D.val = 2\}^{[7]} \\
D &\to 3 \, \{D.val = 3\}^{[8]} \\
D &\to 4 \, \{D.val = 4\}^{[9]} \\
D &\to 5 \, \{D.val = 5\}^{[10]} \\
D &\to 6 \, \{D.val = 6\}^{[11]} \\
D &\to 7 \, \{D.val = 7\}^{[12]} \\
D &\to 8 \, \{D.val = 8\}^{[13]} \\
D &\to 9 \, \{D.val = 9\}^{[14]}
\end{aligned}
$$

1. 文字列 07 : 40 : 32 に対する意味動作付き解析木を示し、根節点の属性 val の値を求めよ。なお、意味動作付き解析木中では、意味動作そのものを書く代わりに、上に示した番号を用いてもよい。(10 点)
2. この翻訳スキームは何を求めるものか、答えよ。(5 点)

5 実行時環境

次の C プログラムを実行すると、どのように出力されるか。(5 点)

```
int x = 2;
void b() { x = x + 1; printf("%d\n", x); }
void c() { int x = 1; printf("%d\n", x + 1); }
void main() { b(); c(); printf("%d\n", x); }
```

平成 19 年度「コンパイラ」定期試験の解説

国島丈生

2007-07-19

1 正則表現と有限オートマトン

1. (a) $(01\,|\,10)(0\,|\,1)^*$

   目立った間違いはありませんでした。

   (b) $-(a|\cdots|z|A|\cdots|Z|0|\cdots|9)\,|\,--(a|\cdots|z)(a|\cdots|z)^+$

   もしくは $-[a-zA-Z0-9]\,|\,--[a-z][a-z]^+$

   後半は「複数個」が 1 個以上なのか 2 個以上なのか曖昧だったので、1 個以上としたもの、つまり $--[a-z]^+$ でも正解としました。この問題で目立ったのは、正則表現の略記法 [] の誤用です。$[a-z]$ は $(a|\cdots|z)$ の略記法ですが、$a,\cdots,z$ は記号でなければなりません。[] 内に正則表現を入れることも、[] を入れ子にすることも、$[a-z|A-Z|0-9]$ のように | 演算子を用いることもできません。今回は明らかな誤用を除いて大幅な減点はしませんでしたが、まず略記法を使わない書き方ができるようにしておいて下さい。略記法を使う場合は、定義を充分に理解してからにして下さい。

2. 図 1 参照。この問題の誤答で目立ったのは、オートマトンの枝のラベルに正則表現を書いているものです（01, $[a-z]$ など）。元々の（決定性）有限オートマトンの定義では、遷移関数は状態 1 つと入力記号 1 つから状態 1 つを返します。つまり、枝のラベルは入力記号 1 つです（$a,\cdots,z$ は a から z までの記号という意味で、枝を複数本書くのをサボるために用いています）。枝に正則表現を書けるように拡張しても計算能力は変わらない（受理される言語クラスはやはり正則言語）のですが、どのような言語を受理するかは定義が必要です。計算能力が変わらないこと、有限オートマトンの厳密な定義を講義では述べていないことから、減点はしませんでしたが、本来は減点すべきものです。

3. $0^*\,|\,0^*1^+$, もしくは 0^*1^*

   状態 0 で受理される言語は 0^*、状態 1 で受理される言語は 0^*1^+ であることから、上のような解答になります。この問題で目立った誤答は、状態 3 に到達する言語（$0^*1^+0(0|1)^*$ など）を解答としたものです。有限オートマトンは、入力を読み切ったとき受理状態（状態遷移図の二重丸の状態）にいる場合、その入力を受理します。最後の状態と受理状態は異なりますので、注意して下さい。

2 文脈自由文法

1. 図 2 参照。目立った誤答はありませんでした。

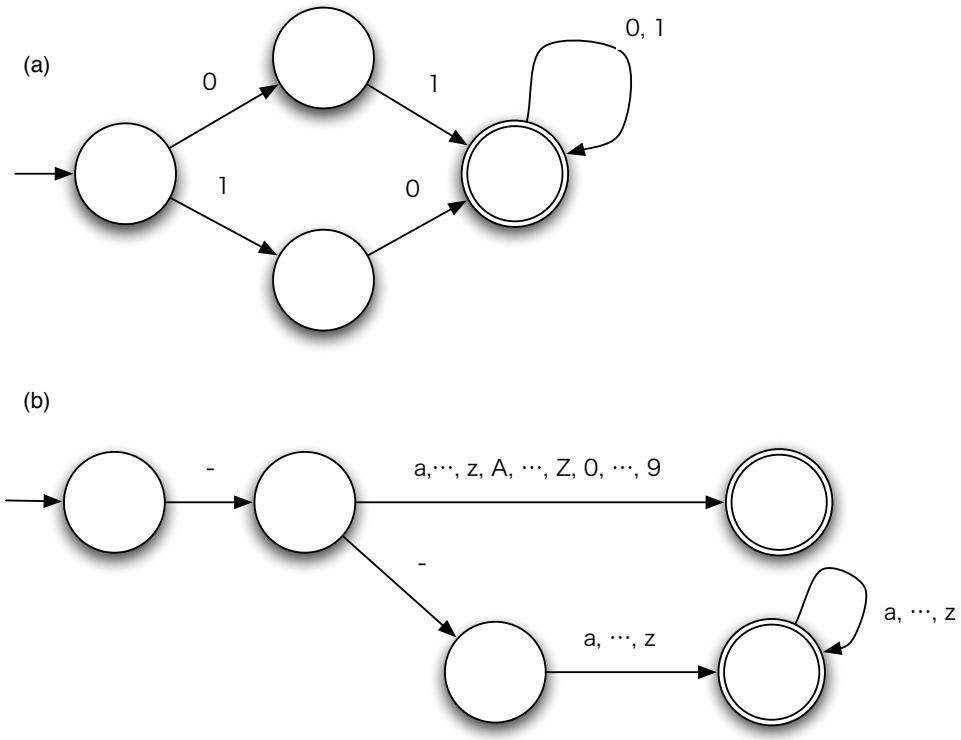


図 1　問 1-2 の解答例

2.

$$S \Rightarrow SbS \Rightarrow abS \Rightarrow abScS \Rightarrow abacS \Rightarrow abacb$$
$$S \Rightarrow ScS \Rightarrow SbScS \Rightarrow abScS \Rightarrow abacS \Rightarrow abaca$$

導出の複数ステップをまとめてしまう（$abScS \Rightarrow abaca$ など）と、最左導出でない導出も含まれてしまうので、解答としてはよくありません。面倒でも、上のようにすべてのステップを書くべきです。また、導出は（生成規則と区別するという意味で）二重矢印を使うべきです。

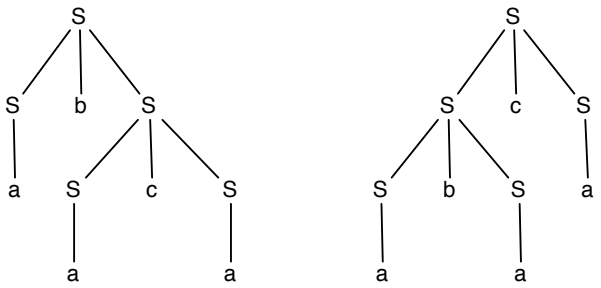


図 2　問 2-1 の解答例