Pós-Graduação em Ciência da Computação
*Teoria de Linguagens*
Professor: Newton José Vieira
e-mail: nvieira@dcc.ufmg.br
DCC/ICEx/UFMG
$2^{\underline{o}}$ semestre de 2003

**Trabalho Especial**

Entrega: até 24/11/03

---

**Observações:**

- **O trabalho é individual.**
- Cada questão vale 1,5 pontos.
- A interpretação de cada questão faz parte do trabalho. Caso alguma questão esteja ambígua ou incorreta, ou caso falte elemento(s) importante(s) para o seu entendimento, faça as suposições e/ou correções que achar convenientes **e explicite-as**.
- Na correção de cada questão será analisado se a mesma foi respondida com **rigor**, **clareza** e **objetividade**. O aluno perderá pontos nas seguintes situações, por exemplo, *mesmo se sua resposta estiver correta*:
  - usa frases incoerentes, ou sem sentido, ou incorretas, ou que nada tenham a ver com o que foi perguntado;
  - apresenta uma solução sem apresentar a(s) idéia(s) subjacente(s), dificultando a correção;
  - apresenta uma solução conceitualmente deselegante;
  - é prolixo, escrevendo coisas desnecessárias ao entendimento da solução.

  Dependendo do "grau" de tais impropriedades, o aluno pode até perder todos os pontos relativos à questão.
- Devolução: até as 16:00 horas do dia 24/11/2003, na sala 4023, ou em meu escaninho na Recepção do DCC, ou via *email*.

---

1. Segue uma definição recursiva do conjunto dos palíndromos sobre $\Sigma$, $Pal$:

   (a) $\lambda \in Pal$;

   (b) $a \in Pal$ para todo $a \in \Sigma$;

   (c) se $x \in Pal$ e $a \in \Sigma$ então $axa \in Pal$;

   (d) nenhuma palavra é membro de $Pal$, a menos que possa ser obtida como mostrado em a, b e c.

O conjunto dos palíndromos sobre $\Sigma$ pode também ser definido utilizando-se a operação *reverso*: *Pal'* $= \{w \in \Sigma^* \mid w = w^R\}$. A operação reverso, por sua vez pode ser assim definida:

(a) $\lambda^R = \lambda$;

(b) $(xa)^R = ax^R$ para quaisquer $x \in \Sigma^*$ e $a \in \Sigma$.

Prove que *Pal* = *Pal'*.

2. Seja, para $n \geq 1$ e $b \geq 2$,

$$L_{n,b} = \{x \in \{0, 1, \ldots, b-1\}^* \mid \eta_b(x) \ mod \, n = 0\},$$

onde $\eta_b(x)$ é o número representado por $x$ na base $b$. Na Seção 2.1.2 da apostila foi mostrado um AFD para $L_{6,2}$. Em anexo, segue um trecho de Kozen[1] em que é mostrado como construir um AFD para $L_{3,2}$, e onde está desenvolvida uma prova da correção de tal AFD. Generalize o método e a prova para $n$ e $b$ arbitrários.

3. Seja a linguagem:

$$L = \{a^i b^j c^k \mid i \neq j \ ou \, j \neq k\}.$$

Suponha que na aplicação do lema do bombeamento (LB) para provar que $L$ não é regular, seja escolhida $z = a^k b^{k!+k} c^{k!+k}$, onde $k$ é a constante do LB. Descubra a razão "misteriosa" para tal escolha.

4. Mostre que as linguagens regulares são fechadas sob a operação *metade*:

$$metade(L) = \{x \mid xy \in L \ e \ |x| = |y|\}.$$

*Sugestão*: mostre como obter um AF $M'$ tal que $L(M') = metade(L)$, a partir de um AF $M$ para $L$; em seguida, prove que $M'$ reconhece $metade(L)$.

5. Construa AFN$\lambda$'s para as linguagens a seguir, usando transições $\lambda$ para *simplificar* a concepção.

   (a) O conjunto das palavras que consistem de zero ou mais `a`'s, seguidos por zero ou mais `b`'s, seguidos por zero ou mais `c`'s.

   (b) O conjunto das palavras que consistem, ou de `01` repetida uma ou mais vezes, ou de `010` repetida uma ou mais vezes.

   (c) O conjunto das palavras de `0`'s e `1`'s tais que pelo menos uma das dez últimas posições contém `1`.

[1] Kozen, D.C. *Automata and Computability*, Springer, 1997.