

0,5	Matrícula:	Nome:
	201910202	Igor Lima Rocha

1.-(2,0 pontos)

- Exemplifique a unificação de expressões com $\{ p(a, x), p(y, h(b)) \}$.
- Exemplifique a resolução-SLD com a consulta $G: p(a, x)$, e a clausula de programa $C: p(x, w) :- q(w), r(x, x, w)$.
- Qual é o significado semântico da substituição resposta de um ramo fechado numa árvore-SLD de uma consulta G e um programa lógico P ?
- Escreva um programa PROLOG cuja árvore SLD tenha dois ramos infinitos e dois ramos finitos. Calcule a substituição resposta dos ramos finitos e 3 passos em cada um dos ramos infinitos.

2.-(8,0 pontos) A linguagem de programação **logpro** é definida pela seguinte gramática, com símbolo inicial **<program>**:

$p_0: \langle \text{program} \rangle \rightarrow \langle \text{clauses} \rangle$

$p_1: \langle \text{clauses} \rangle \rightarrow \langle \text{clause} \rangle \mid \langle \text{clause} \rangle. \langle \text{clauses} \rangle$

$p_3: \langle \text{clause} \rangle \rightarrow \langle \text{fact} \rangle \mid \langle \text{hornclau} \rangle$

$p_5: \langle \text{fact} \rangle \rightarrow \langle \text{pred} \rangle$

$p_6: \langle \text{hornclau} \rangle \rightarrow \langle \text{pred} \rangle :- \langle \text{preds} \rangle$

$p_7: \langle \text{preds} \rangle \rightarrow \langle \text{pred} \rangle \mid \langle \text{pred} \rangle, \langle \text{preds} \rangle$

$p_9: \langle \text{pred} \rangle \rightarrow \langle \text{pname} \rangle (\langle \text{parmlist} \rangle)$

$p_{10}: \langle \text{pname} \rangle \rightarrow l \mid m \mid o \mid p \mid q \mid r \mid s$

$p_{17}: \langle \text{parmlist} \rangle \rightarrow \langle \text{name} \rangle \mid \langle \text{name} \rangle, \langle \text{parmlist} \rangle$

$p_{19}: \langle \text{name} \rangle \rightarrow \langle \text{num} \rangle \mid \langle \text{const} \rangle \mid \langle \text{var} \rangle$

$p_{22}: \langle \text{num} \rangle \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$p_{32}: \langle \text{const} \rangle \rightarrow a \mid b \mid c \mid d \mid e$

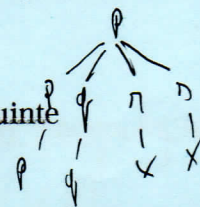
$p_{37}: \langle \text{var} \rangle \rightarrow u \mid v \mid w \mid x \mid y \mid z$

a) Escreva um programa em **logpro** para determinar se uma matriz 2x2 é um quadrado latino.

b) Desenvolva a árvore SLD para a consulta $G: m(1, u, y, 7)$, onde m é o predicado principal do programa (2.a). Quantos ramos fechados existem?

c) Desenvolva na linguagem C o analisador descendente recursivo para a gramática do **logpro**.

d) Projete o autômato de pilha que reconhece a linguagem **logpro**, analise o programa (2.a) e construa a árvore de análise correspondente.



2) b)

$G_0: m(L, U, Y, Z)$

$1. \{ U_1 = L, V_1 = U, W_1 = Y, X_1 = Z \}$

$2. \{ Y_2 = L, Z_2 = U \}$

$G_1: p(L, U), p(Y, Z), p(L, Y), p(U, Z)$

$G_2: q(L), q(U), r(L, U), r(U, L), p(Y, Z), p(L, Y), p(U, Z)$
 $4. \{ \}$

$G_4: q(U), r(L, U), r(U, L)$
 $5. \{ U = 2 \}$

$G_5: r(L, 2), r(2, U)$
 $10. \{ \}$

$G_{10}: r(2, L)$
 $20. \{ \}$

Existen 26 ramas finales.

```
2-c) void program() {
    clause()
}

void clause() {
    clause()
    if (next token == '.') {
        loc()
        if (next token) {
            clause()
        }
    }
}
```

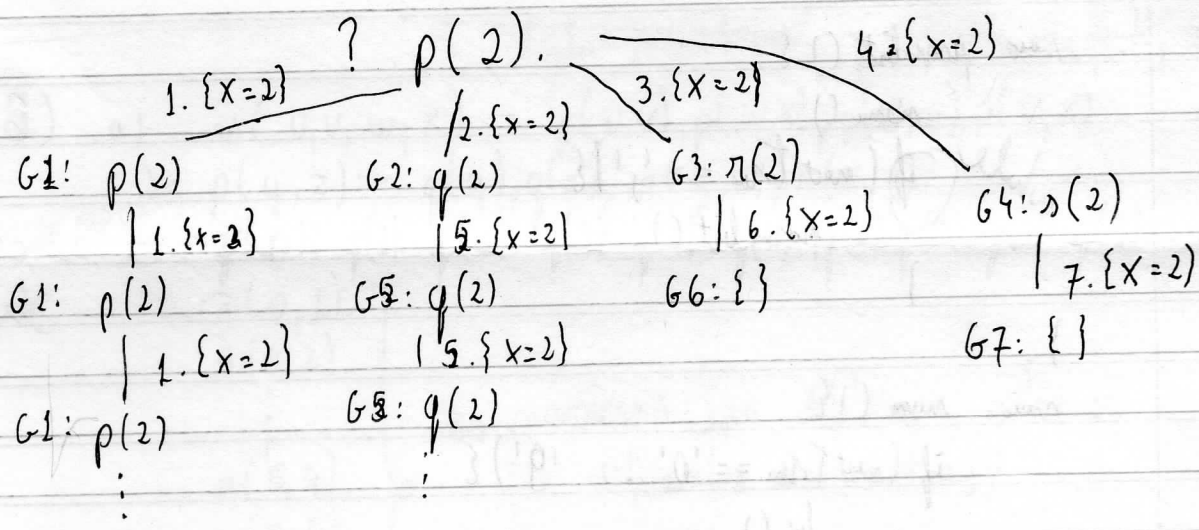
```
void clause() {
    fact()
    horn clause()
}

void fact() {
    pred()
}

void horn clause() {
    pred()
    if (next token == ":-") {
        preds()
    } else {
        error()
    }
}
```

1) d) ~~$p(x) :- p(x).$~~
 ~~$p(x) :- q(x).$~~
 ~~$q(x).$~~
 ~~$q(1).$~~

- 1 $p(x) :- p(x).$ — ramo infinito
- 2 $p(x) :- q(x)$
- 3 $p(x) :- r(x)$ com 6 ramo Finito
- 4 $p(x) :- s(x)$ com 7 ramo Finito
- 5 $q(x) :- q(x) c/2$ ramo infinito
- 6 $r(x).$
- 7 $s(x).$



b) $G0: p(a, x)$
 | 1. $\{x_1 = a, w_1 = x\}$

 $G1: -q(x), r(a, a, x)$

```

void preds() {
  pred()
  while (nextToken == ',') {
    loc()
    preds()
  }
}

```

```

void pred() {
  pname()
  if (nextToken == '(') {
    loc()
    paramlist()
    if (nextToken == ')') {
      loc()
      if (error()) {
        //
      }
    }
  }
}

```



```

void pname () {
    if (nextToken == 'l' || 'm' ... 's') {
        loc()
    } else {
        error()
    }
}

void paramlistname () {
    num()
    const()
    var()
}

void paramlist () {
    name()
    while (if (nextToken == ',') {
        paramlist()
    }
}

void num () {
    if (nextToken == '0' ... '9') {
        loc()
    }
}

void const () {
    if (nextToken == 'a' ... 'x') {
        loc()
    }
}

void var () {
    if (nextToken == 'u' ... 'z') {
        loc()
    }
}

```

2) a) ~~$m(A1, A2, B1, B2) :- \text{perm}(A1, A2), \text{perm}(B1, B2), \text{perm}(A1, B1), \text{perm}(A2, B2).$~~
 ~~$\text{perm}(X, Y) :- X \neq Y.$~~

b)

~~$G0: m(1, U, Y, 7)$~~

~~$1. \{A1=1, A2=U, B1=Y, B2=7\}$~~

~~$2. \{X=1, Y=U\}$~~

~~$G1: \text{perm}(1, U), \text{perm}(Y, 7), \text{perm}(1, Y), \text{perm}(U, 7)$~~

2) a) $m(u, v, w, x) :-$ $\overbrace{p(u, v), p(w, x)}^{\text{linhas permutadas}}, \overbrace{p(u, w), p(v, x)}^{\text{colunas permutadas}}.$

$p(y, z) :- q(y), q(z), \pi(y, z), \pi(z, y).$
 números $\rightarrow q(0), q(1), q(2), q(3), q(4), q(5), q(6), q(7), q(8), q(9).$
 diferente $\rightarrow \pi(0, 1), \pi(0, 2), \dots, \pi(5, 4), \pi(5, 6), \dots, \pi(9, 7), \pi(9, 8).$

os números são diferentes entre si
 (e diferentes das letras)

letras são constantes
 $q(a), q(b), q(c), q(d), q(e).$
 $\pi(a, b), \pi(a, c), \dots, \pi(c, b), \pi(c, d), \dots, \pi(e, c), \pi(e, d).$

as letras são diferentes entre si
 (e diferentes dos números)