Primeira Prova de Linguagens de Programação
- DCC024 -
Ciência da Computação

Nome: _____
"Eu dou minha palavra de honra que não trapacearei neste exame."

Número de matrícula: _____

As regras do jogo:

- A prova é sem consulta.
- Quando terminar, não entregue nada além do caderno de provas para o instrutor.
- Quando escrever código, a sintaxe correta é importante.
- Cada estudante tem direito a fazer uma pergunta ao instrutor durante a prova. Traga o caderno de provas quando vier à mesa do instrutor.
- A prova termina uma hora e quarenta minutos após seu início.
- Seja honesto e lembre-se: **você deu sua palavra de honra**.

Alguns conselhos:

- Escreva sempre algo nas questões, a fim de ganhar algum crédito parcial.
- Se não entender a questão, e já tiver gasto sua pergunta, escreva a sua interpretação da questão junto à resposta.
- Lembre-se do P.O.F. da aula quatro. E lembre-se também: perguntando qual o P.O.F. da aula quatro, você perde a sua pergunta.
- A prova não é difícil, ela é divertida, então aproveite!

Tabela 1: Pontos acumulados (para uso do instrutor)

Questão 1	Questão 2	Questão 3	Questão 4	Questão 5	Questão 6

1. (10 pontos) Seja `SET` a assinatura que descreve um conjunto de inteiros em SML:

```
signature SET =
sig
  type set
  val new : set
  val add : set -> int -> set
  val contains : set -> int -> bool
  val remove : set -> int -> set
end ;
```

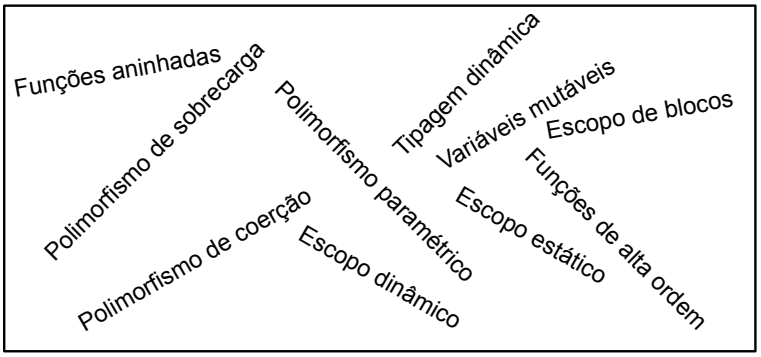
Esta assinatura pode ser implementada de várias formas diferentes. Por exemplo, a implementação abaixo representa conjuntos como funções:

```
structure FunSet :> SET =
struct
  type set = int -> bool
  val new = fn x => false
  fun add f i = fn x => if x = i then true else f x
  fun contains f i = f i
  fun remove f i = fn x => if x = i then false else f x
end ;
```

Esta implementação da assinatura `SET`, contudo, não é muito eficiente. Complete a implementação abaixo, usando uma abordagem baseada em listas **ordenadas** de inteiros.

```
structure ListSet :> SET =
struct
  type set = int list

  val new =



  fun add s i =



  fun contains s i =



  fun remove s i =



end ;
```

2. (10 pontos - 1 ponto para cada resposta correta) Considere o quadro abaixo, em que estão escritos diversos conceitos típicos de linguagens de programação:



E sejam as duas linguagens: ANSI C, e o subconjunto de SML que vimos em sala. Separe tais conceitos entre os quatro grupos abaixo.

(a) Conceitos que existem somente em ANSI C:

(b) Conceitos que existem somente no subconjunto de SML que vimos em sala:

(c) Conceitos que existem em ambas as linguagens.

(d) Conceitos que não existem em nenhuma delas.

3. As questões que se seguem dizem respeito à função lógica NAND, cuja tabela verdade é dada abaixo:

True	True	False
True	False	True
False	True	True
False	False	True

(a) (3 pontos) Escreva uma função binária `nand` em SML de tipo `bool * bool -> bool` que calcule a função NAND de dois booleanos.

(b) (3 pontos) Considere as seguintes funções:

```
fun nandl l = foldl nand true l;

fun nandr l = foldr nand true l;
```

Sendo `nand` a mesma função implementada na questão anterior, qual é o tipo inferido para a função `nandl`? Note que `nandl` e `nandr` possuem o mesmo tipo.

(c) (4 pontos) Considere a função `comp`, implementada a partir das funções `nandl` e `nandr` da questão anteiror:

```
fun comp l = nandl l = nandr l;
```

Existe alguma situação em que a função `comp` retorna o valor `false`? Em caso negativo, justifique a sua resposta, doutro modo mostre um exemplo.

4. (10 pontos) Considere a função `f` abaixo, implementada em C:

```
int f(int a, int b, int c) {
  int aa = a+3;
  int bb = b << 2;
  int cc = c ^ 1001;
  return aa + bb + cc;
}
```

Esta função, quando compilada para assembly do x86, produz um programa muito parecido com este abaixo. A sintaxe deste código assembly foi simplificada:

```
f:
   pushl   %ebp
   movl    %esp, %ebp
   subl    24, %esp
   movl    8(%ebp), %eax
   addl    3, %eax
   movl    %eax, -20(%ebp)
   movl    12(%ebp), %eax
   sall    2, %eax
   movl    %eax, -16(%ebp)
   movl    16(%ebp), %eax
   xorl    1001, %eax
   movl    %eax, -12(%ebp)
   movl    -16(%ebp), %eax
   addl    -20(%ebp), %eax
   addl    -12(%ebp), %eax
   leave
   ret
```

Um programa em assembly do x86 é uma sequência de **instruções**. Cada instrução é formada por um **opcode**, e zero, um ou dois **operandos**. Um operando pode ser um **número**, um **registrador** ou um **endereço**. Os oito registradores são `eax, ebx, ecx, edx, epb, edi, esi` e `esp`. Endereços são formados por um número seguido de um registrador, entre parênteses. Escreva a gramática do assembly do x86. Suponha que já foi definida um não-terminal OPCODE, que produz todos os opcodes válidos do x86. Suponha também que todos os opcodes podem usar os mesmos operandos, da mesma forma. Não se preocupe se a sua gramática for ambígua.

5. Considere a implementação da função `fib` abaixo, em SML/NJ, que calcula o n-ésimo número da sequência de Fibonacci:

```
fun fib 0 = 0
  | fib 1 = 1
  | fib n = (fib (n-1)) + (fib (n-2))
```

   (a) (2 pontos) Uma chamada como `fib 0` invoca a função `fib` uma vez. Uma chamada como `fib 2` invoca a função `fib` três vezes, isto é, além da própria chamada `fib 2`, teremos também `fib 0` e `fib 1`. Quantas vezes esta função é invocada por uma chamada igual a `fib 4`.

   (b) (8 pontos) Esta implementação de `fib` não é muito eficiente, pois uma quantidade exponencial de invocações (no tamanho da entrada em bits) é feita a cada chamada. Escreva uma implementação de `fib` mais eficiente, que perfaça uma quantidade linear de chamadas.

6. Uma linguagem é estaticamente tipada quando o tipo de cada expressão pode ser resolvido em tempo de compilação. Neste caso o compilador pode usar inferência ou anotações para descobrir o tipo das variáveis. Uma linguagem é dinamicamente tipada quando o tipo da variável é resolvido em tempo de execução.

   (a) (2.5 pontos) SML é dinamicamente tipada ou estaticamente tipada? Como o compilador descobre o tipo das variáveis em SML?

   (b) (2.5 pontos) PHP é estaticamente tipada ou dinamicamente tipada? Como o compilador descobre o tipo das variáveis em PHP?

   (c) (2.5 pontos) C é estaticamente tipada ou dinamicamente tipada? Como o compilador descobre o tipo das variáveis em C?

   (d) (2.5 pontos) Cite uma linguagem de programação que não possua sistema de tipos. Por linguagem de programação nos referimos a qualquer sistema de descrição de computações que seja Turing completo