Primeira Prova de Análise e otimização de Código
- DCC888 -
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.
- Você pode sacrificar sua pergunta para saber porque os seres humanos riem, segundo *Robert Heilen*.
- Se não entender a questão, e já tiver gasto sua pergunta, escreva a sua interpretação da questão junto à resposta.
- 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

1. Considere uma linguagem que processa arranjos vindos de uma conexão de rede. Essa linguagem possui o seguinte conjunto de instruções:

Inicialização	$\Rightarrow$	$v = \texttt{recv}()$
Atribuição	$\Rightarrow$	$v = \texttt{subrange}(v', n_1, n_2), \{n_1, n_2\} \subseteq \mathbb{N}$
Asserção	$\Rightarrow$	$\texttt{check}(v, n, l), n \in \mathbb{N}, l \in$ rótulo de instrução
Escrita	$\Rightarrow$	$\texttt{putc}(v, n), n \in \mathbb{N}$

A semântica dessas instruções é descrita - informalmente - a seguir:

- Uma *inicialização* lê uma lista de caracteres da rede (ou de qualquer fonte imaterial de caracteres) e inicializa um vetor v com esse arranjo. A leitura termina quando um caractere de final de linha é lido. O tamanho do arranjo não é conhecido em tempo de compilação.
- A instrução de *atribuição* inicializa o arranjo v com o sub-arranjo de v' que vai de seu índice n_1 até $n_2 - 1$.
- A instrução de *escrita* imprime o caractere na n-ésima posição do arranjo v.
- A instrução de *asserção* verifica se o arranjo v possui no mínimo n posições válidas e desvia o fluxo de programa para l caso contrário. Doutro modo, a instrução simplesmente avança o contador de instruções uma posição.

Abaixo vemos um programa que lê um vetor de caracteres, e imprime a sua segunda posição, se houver mais de uma posição disponível, ou a primeira posição caso contrário. Note que arranjos são indexados a partir do índice zero:

l_0	$v = \texttt{recv}()$
l_1	$b = \texttt{assert}(v, 2, l_3)$
l_2	$\texttt{putc}(v, 0)$
l_3	$\texttt{putc}(v, 1)$

(a) (5 Pontos) Defina as funções de transferência para uma análise que estime o tamanho dos arranjos. Suas funções de transferência devem calcular informações entre dois rótulos consecutivos de programa. Em outras palavras, você estará associando informação às arestas do seu grafo de fluxo de controle. Note que a instrução `assert` gera duas informações diferentes, pois ela possui dois destinos. Essa informação é constituída de pares (nome de arranjo $\times$ tamanho do arranjo). O tamanho de um arranjo pode ser um dentre três valores:

- Uma constante
- O valor "não sei"
- O valor "indefinido"

(b) (5 Pontos) Defina uma operação de *junção* para a sua análise. Essa operação deve calcular o conjunto de entrada em um rótulo l a partir da informação disponível em cada aresta (l', l), em que l' é um predecessor de l. Note que uma mesma operação será suficiente para todas as cinco instruções diferentes do programa.