PROVA 1
ALL'S WELL THAT ENDS WELL

RAUL H.C.LOPES

1. Introdução

Este documento contém três opções de provas. Elas apresentam demandas diferentes e, por isso mesmo, valores diferentes. Escolha a sua, resolva e esteja atento ao prazo de entrega da prova de sua escolha.

Cada prova deve ser resolvida individualmente, sendo, no entanto, permitidas consultas a livros, notas de aula, colegas, namoradas, etc. Apesar disso, cada aluno deverá produzir sua própria solução e ser capaz de responder a uma argüição oral sobre a mesma.

Cada uma das seções a seguir contém uma alternativa de prova. Escolha a sua e divirta-se.

2. Usando seu provador

Para resolver esta versão da prova, você usará o provador que construiu no trabalho prático. As regras são:

- Você submeterá por e-mail um *tarball* contendo os seguintes arquivos:
  - **readme.tex**
    Documento curto, conterá instruições obtenção das provas no seu provador.
  - Arquivos com os axiomas usados em cada prova, com nomes identificados a seguir.
  - Arquivos com as provas obtidas, com nomes identificados a seguir.
- A data final para submissão é dia **15/09**, às **17 horas**.
- Como todos os exercícios desta seção usarão seu provador, a nota obtida na prova representará **70%** da nota do seu trabalho prático: o resto da nota do trabalho será dado pelo resolução da série *tough*, disponível para download.

Os exercícios:

Exercício 1.

- ***Descrição:*** *Apresente axiomas formalizando o problema da torre de Hanoi. Use seu provador para construir a seqüência de movimentos para transferir quatro discos entre duas torres.*
- ***Nome do arquivo de axiomas: hanoi.in***
- ***Nome do arquivo de prova: hanoi.out***

Nos próximos exercícios use uma formalização do cálculo dos números da série de Fibonacci.

Exercício 2.

- ***Descrição:*** *Cálcule o elemento da posição cinco da série, assumindo que o elemento da posição zero é* 0 *e o da posição um é* 1*.*
- ***Nome do arquivo de axiomas: fib.5.x.in***
- ***Nome do arquivo de prova: fib.5.x.out***

Exercício 3.

- ***Descrição:*** *Cálcule a posição do número* 8 *na série.*
- ***Nome do arquivo de axiomas: fib.x.8.in***
- ***Nome do arquivo de prova: fib.x.8.out***

Exercício 4.

- ***Descrição:*** *Mostre que* 8 *não é o elemento da posição três da série.*
- ***Nome do arquivo de axiomas:fib.3.8.in***
- ***Nome do arquivo de prova:fib.3.8.out***

Exercício 5.

- ***Descrição:*** *Mostre que* 13 *é o elemento sete da série.*
- ***Nome do arquivo de axiomas:fib.13.7.in***
- ***Nome do arquivo de prova:fib.13.7.out***

Exercício 6.

- ***Descrição:*** *Mostre que* 4 *não ocorre na série.*
- ***Nome do arquivo de axiomas:fib.4.x.in***
- ***Nome do arquivo de prova:fib.4.x.out***

Exercício 7.

- ***Descrição:*** *Formalize o procedimento de inserção em ordem do algoritmo* ***insertion sort****. Cálcule o resultado de inserir* 2 *em uma lista que contenha* $[1, 2, 3]$*.*
- ***Nome do arquivo de axiomas:insert.in***
- ***Nome do arquivo de prova:insert.out***

Nos próximos itens, você precisará de uma formalização do procedimento de cálculo do reverso de uma seqüência e do procedimento que, dados inteiros positivos x e y, retorna a seqüência composta dos inteiros maiores ou iguais que x e menores ou iguais que y. Assuma que x nunca é maior do que y.

Exercício 8.

- ***Descrição:*** *Calcule a seqüência dos inteiros entre* 2 *e* 9.
- ***Nome do arquivo de axiomas:fromupto.in***
- ***Nome do arquivo de prova:fromupto.out***

Exercício 9.

- ***Descrição:*** *Use uma única cláusula para calcular a seqüência dos inteiros entre* 2 *e* 9 *e seu reverso.*
- ***Nome do arquivo de axiomas: seqrev.in***
- ***Nome do arquivo de prova:seqrev.out***

Exercício 10.

- ***Descrição:*** *Use uma única cláusula para calcular a seqüência* x *dos inteiros entre* 1 *e* 4 *e para calcular a seqüência que tem* x *como reverso.*
- ***Nome do arquivo de axiomas: invrev.in***
- ***Nome do arquivo de prova:invrev.out***

Exercício 11.

- ***Descrição:*** *Mostre que o reverso de* $[1, 2, 3, 4, 5]$ *não é* $[5, 4, 3, 2, 2]$.
- ***Nome do arquivo de axiomas: notrev.in***
- ***Nome do arquivo de prova:notrev.out***

3. Escolha um provador

Para resolver esta versão da prova, você usará um dos seguintes provadores: Otter, Isabelle, PVS, provador que você construiu no trabalho prático. As regras são:

- Você submeterá por e-mail um *tarball* contendo os seguintes arquivos:
  - **readme.tex**
    Documento curto, conterá instruições obtenção das provas no seu provador.
  - Arquivos com os axiomas usados em cada prova com nomes identificados a seguir.
  - Arquivos com as provas obtidas com nomes identificados a seguir.
- A data final para submissão é dia **15/09**, às **17 horas**.

- A resolução deste exercício substitui todas as notas do semestre: um semestre em quatro dias.

O exercício:

Escolha um provador, formalize uma *Máquina de Turing não determinística* e resolva com sua máquina ao menos cinco problemas diferentes de satisfatibilidade do Cálculo de Proposições, com, no mínimo, três variáveis proposicionais cada. Restrição: sua formalização dos problemas não pode usar cláusulas e cada problema deve conter ao menos três constantes lógicas diferentes.

4. Cérebro, suor, caneta e papel

As regras para a resolução desta opção de prova são:

- A resolução deverá ser entregue até **13/09/04**, às **12 horas**.
- A solução será entregue em folha de papel decente, manuscrita com caneta, com o mínimo de rasuras.
- As provas construídas deverão explicitar claramente o sistema formal usado, os axiomas e regras de inferência de cada passo.

O exercício:

- Formalize em uma lógica de primeira ordem uma versão de **merge sort**.
- Prove a correção total de sua formalização.

5. They think it's all over...

There's a sign on the wall
But she wants to be sure
'Cause you know sometimes words have two meanings.
In a tree by the brook
There's a songbird who sings,
Sometimes all of our thoughts are misgiven.

Stairway to heaven, Led Zeppelin