

Trabalho Prático

Lógica Computacional

2020/2021

Departamento de Informática
Universidade da Beira Interior

Data limite para entrega: Será anunciada após votação no Slack na qual será decidida uma extensão. Este enunciado será alterado após o fecho da votação para reflectir a data decidida (votação fecha no dia 20 de Abril).

Grupos: Este trabalho é para ser realizado em grupos de dois elementos. Os grupos podem ser formados por elementos de turnos práticos distintos mas apenas entre grupos que são lecionados pelo mesmo docente, i.e. alunos dos turnos práticos PL1, PL2 e PL3 podem formar grupo entre si e os alunos do PL4 e PL5 também. Alunos dos turnos PL4 e PL5 não podem formar grupo com alunos dos turnos PL1, PL2 e PL3.

1 Trabalho a Realizar

O objectivo deste trabalho é implementar o **algoritmo de Horn**. Dada uma fórmula, o programa deverá verificar se é fórmula de Horn e:

- Se é fórmula de Horn, deverá determinar e indicar se a fórmula é possível ou contraditória.
- Se não é fórmula de Horn, deverá indicar que o algoritmo não é aplicável.

Na secção seguinte são apresentados os detalhes relativos ao *input* que o programa terá de processar e ao *output* esperado. Terá de seguir estas instruções exactamente como indicadas.

1.1 Input e Output

O input consistirá em fórmulas na **forma normal conjuntiva**, uma por linha, onde cada conectivo é representado da seguinte forma:

- \neg é representado por \sim
- \wedge é representado por $\&$
- \vee é representado por $|$

Parênteses são utilizados apenas para delimitar as disjunções (incluindo as unitárias).

As fórmulas terão, no máximo, 100 cláusulas. O programa tem de ler do *standard input* uma fórmula por linha até não existirem mais fórmulas para processar. Para cada fórmula deverá imprimir o **output** correspondente antes de ler a seguinte.

Os literais (proposições atômicas) são representados por letras minúsculas e maiúsculas. Portanto, existem no máximo 52 proposições atômicas distintas. Note que, neste trabalho, T e F são interpretados como literais e não como os valores verdadeiro e falso.

Por exemplo, a fórmula:

$$(\neg p \vee Q \vee \neg r) \wedge (\neg t) \wedge (P) \wedge (\neg T \vee x \vee \neg P) \wedge (F)$$

que contém 5 cláusulas e 8 proposições atômicas distintas, será passada ao programa no seguinte formato:

$$(\sim p \mid Q \mid \sim r) \& (\sim t) \& (P) \& (\sim T \mid x \mid \sim P) \& (F)$$

Os literais, positivos ou negativos, podem estar repetidos numa disjunção. Por exemplo, a fórmula $(\neg p \vee Q \vee Q) \wedge (\neg t)$ deve ser considerada fórmula de Horn.

O programa deverá produzir como *output*:

- **SAT**, se a fórmula é possível
- **UNSAT**, se a fórmula é contraditória
- **NA**, se a fórmula não é fórmula de Horn

Cada output tem de ser seguido do caracter de newline (`'\n'`), caso contrário, mesmo que a resposta esteja correcta, será rejeitada.

Exemplo de execução:

Input:

```
(~p | Q | ~r) & (~t) & (P) & (~T | x | Z) & (F)
(~p | R ) & (~x) & (p) & (~p | x | ~R | x)
(~p | a ) & (p) & (~p | ~R | x) & (~W | P | ~h) & (F) & (~f | ~q | e) & (~t |
~k | ~l)
```

Output:

```
NA
UNSAT
SAT
```

Poderá encontrar no Moodle o ficheiro `input_demo.txt` que contém exemplos de *input*. Poderá utilizar este ficheiro para testar o programa mas **note que se o seu programa produzir os *outputs* esperados para este exemplo, tal não certifica que o programa está correcto nem que vai passar nos testes executados pelo Mooshak**. Os testes executados pelo Mooshak serão diferentes e em maior número. O ficheiro `output_demo.txt`, também disponibilizado no Moodle, contém o *output* esperado para o exemplo disponibilizado. Deverá também executar testes adicionais definidos por si.

1.2 Linguagens de Programação

Poderá implementar o programa em C (standard C99) ou em Java 8 (Java(TM) SE Runtime Environment (build 1.8.0_91-b14)). **Se utilizar funcionalidades não suportadas por estas versões, o programa não será aceite pelo Mooshak e, consequentemente, a nota do trabalho será de zero valores. Não serão abertas excepções.**

Implementações em C: deverá garantir que a sua solução compila e funciona correctamente após compilação com o seguinte comando (onde `main.c` é substituído pelo nome do seu ficheiro):

```
gcc -o main main.c -Wall -std=c99 -lm
```

e só deverá ser submetido para avaliação após compilação com sucesso e sem avisos.

Implementações em Java: deverá garantir que não utiliza funcionalidades de versões do Java que não sejam suportadas no Java 8. Deverá também compilar o seu programa com o seguinte comando (onde `Main.java` é substituído pelo nome do seu ficheiro):

```
javac -encoding utf8 Main.java
```

e só deverá submeter para avaliação após compilação com sucesso.

Para executar o programa localmente deverá utilizar o comando seguinte (assumindo o nome `Main`):

```
java -Xmx64M -Xss32M -classpath . Main
```

que corresponde ao comando que será utilizado no Mooshak.

2 Classificação e Critérios de Correção

A nota obtida no Mooshak é a **nota máxima** que poderá obter (0 a 200 pontos, onde 200 pontos representam 20 valores). A nota final terá em consideração vários factores, incluindo:

1. Eficiência e elegância da solução;
2. Estilo de programação, uso de algoritmos e estruturas de dados apropriados e facilidade de leitura do código;
3. Comentários no código fonte que explicam o código e as abordagens seguidas em cada algoritmo;
4. Qualidade do ficheiro README;
5. Optimizações implementadas.

Todas as funcionalidades têm de ser implementadas de raiz utilizando a linguagem C ou Java. A utilização de chamadas ao sistema ou a sistemas externos para que estes executem tarefas do programa não é aceite.

Trabalhos que apresentem código indicativo de tentativa de passar nos testes do Mooshak sem implementar e/ou executar os algoritmos apropriados para obtenção do *output*, **terão nota zero e poderão ser tratados como tentativa de fraude.**

Após a submissão poderá ser requerido que apresente o trabalho, individualmente ou em grupo, e poderá ter de fazer alterações ao seu trabalho durante essa apresentação. A apresentação será utilizada para o cálculo da nota final. A decisão de quem apresenta ou não o trabalho é tomada unicamente pelo corpo docente.

Conduta: Chama-se a atenção para o Código de Integridade e para o Regulamento Disciplinar dos Estudantes da Universidade da Beira Interior.

Casos de fraude académica serão levados muito a sério. **Plágio resultará em reprovação na disciplina de todos os envolvidos e será tratado de acordo com o regulamento disciplinar em vigor, o que poderá ter consequências graves.** O corpo docente da cadeira será o único juiz do que se considera ou não plágio. Não vale a pena copiar código e apenas alterar nomes de variáveis, funções/métodos, adicionar ou remover chavetas, etc. O corpo docente utilizará software especializado na detecção de plágio em trabalhos de programação. Relembremos que quem faz o trabalho tem o direito **e o dever** de o manter para si e de não partilhar nem deixar consultar a sua solução. Relembremos também que quem tenta passar o trabalho dos outros como seu está a cometer fraude académica e, em caso de utilização de trabalho de outros colegas, está, directamente e conscientemente, a prejudicar esses colegas. Plágio contempla não apenas a submissão, integral ou parcial, de trabalho de terceiros mas também submissão de trabalho de outras fontes, tais como

websites e livros. Discussões sobre o trabalho prático entre colegas de grupos diferentes são permitidas (e encorajadas). No entanto, apenas devem discutir ideias e nunca partilhar ou discutir os detalhes da vossa implementação.

3 Entrega do Trabalho

A entrega do trabalho deve ser feita via Mooshak e Moodle. **No início do ficheiro de código tem de identificar claramente o nome, número de aluno, e turno prático dos elementos do grupo (num comentário).** Todas as funções devem estar devidamente documentadas com comentários no código.

Entrega no Mooshak. O primeiro passo da entrega consiste no registo de uma conta no sistema Mooshak. **Só se pode registar um elemento do grupo.** Para obter a sua senha, terá de inserir o seu número de aluno (e.g., a12345 ou Z123) no formulário disponível em

`https://penhas.di.ubi.pt/~amendes/cgi-bin/getpass`

A password será gerada e enviada para o seu email registado na plataforma Moodle (não se esqueça de verificar a pasta de “E-mail de Lixo” caso não receba a password). A partir desse momento, poderá aceder ao sistema Mooshak onde deverá entregar o seu trabalho (deverá também entregar no Moodle, ver informação abaixo):

`https://penhas.di.ubi.pt/~amendes`

A submissão do trabalho deverá ser feita por apenas um elemento do grupo. Para submeter o trabalho, deverá:

- Utilizar o botão “Choose file”, seleccionando o ficheiro contendo todo o código fonte do seu trabalho.
- De seguida clique no botão “Submit” para efetuar a submissão. **Aguarde (cerca de 1 minuto) para que o sistema processe a sua submissão.**
- Quando a submissão tiver sido processada, poderá visualizar na tabela o resultado correspondente. Se o resultado for *Wrong answer*, poderá clicar no resultado para obter dicas.
- Para sair do sistema utilize o botão “Logout”.

Cada grupo tem direito a um máximo 15 submissões no Mooshak sem penalização. Por cada submissão adicional serão descontados 0,2 valores.

Mais detalhes sobre o sistema Mooshak serão mostrados na Secção 4.

Entrega no Moodle. Após o envio da **submissão final** no Mooshak, terá de submeter o mesmo ficheiro no Moodle, utilizando o link de submissão do trabalho prático. A submissão tem de ser feita através da conta do aluno que submeteu o trabalho no Mooshak.

Deverá submeter no Moodle um ficheiro .zip (não são aceites outros formatos) contendo os seguintes ficheiros:

1. O ficheiro com o código fonte da sua solução;
2. Ficheiro **README**, com um máximo de 2000 caracteres, no qual descreve, resumidamente mas com detalhe suficiente, os algoritmos que utilizou. Neste mesmo ficheiro deverá listar quaisquer optimizações que tenha implementado.

O nome do ficheiro .zip tem de ser composto pelos números de aluno dos elementos do grupo, e respectivos turnos práticos, separados pelo carácter ‘_’ (underscore). Por exemplo, se os alunos com números a12345 do PL1 e Z123 do PL2 formam grupo, o ficheiro deverá chamar-se **a12345PL1_Z123PL2.zip**

Garanta que fez todos os passos necessários para a submissão. Terá de aceitar a declaração de submissão e finalizar a submissão carregando no botão “Submeter o trabalho”. Note que trabalhos não submetidos como finais não serão considerados e que caso o ficheiro submetido no Moodle seja diferente do ficheiro final submetido no Mooshak, a sua submissão poderá não ser considerada.

4 Sistema Mooshak

Submeta o seu projeto atempadamente, dado que as restrições seguintes podem não lhe permitir fazê-lo no último momento:

- Só poderá efetuar uma nova submissão pelo menos 15 minutos depois da submissão anterior.
- Só são permitidas 10 submissões em simultâneo no sistema, pelo que uma submissão poderá ser recusada se este limite for excedido¹.
- Não pode ter submissões duplicadas, ou seja, não pode ter uma submissão igual a outra anterior.
- Será considerada para avaliação a última submissão (mesmo que tenha pontuação inferior a submissões anteriores). Deverá, portanto, verificar cuidadosamente que a última entrega realizada corresponde à versão do projeto que pretende que seja avaliada. **Não serão abertas exceções.**

¹Note que o limite de 10 submissões simultâneas no sistema Mooshak implica que, caso haja um número elevado de tentativas de submissão sobre o prazo de entrega, alguns grupos poderão não conseguir submeter nessa altura e verem-se, por isso, impossibilitados de submeter o código dentro do prazo.

4.1 Testes Automáticos

A avaliação da execução será feita através do sistema Mooshak, que executará o seu programa num conjunto de testes. O tempo de execução de cada teste está limitado, bem como a memória utilizada. O sistema não deverá ser utilizado para *debug* e como tal, relembra-se que o tempo entre submissões e o número de submissões é limitado.

Cada grupo tem direito a um máximo 15 submissões no Mooshak sem penalização. Por cada submissão adicional serão descontados 0,2 valores.

Não será disponibilizado qualquer tipo de informação sobre os casos de teste utilizados pelo sistema de avaliação automática.

Pontuação: A avaliação automática indicará uma nota de 0 a 200 pontos (200 pontos equivalem a 20 valores). Esta nota será considerada em conjunção com o restantes critérios de avaliação.

5 Recomendações

Esta secção lista recomendações importantes que deverá seguir e ter em consideração:

- Leia todo o enunciado com atenção. Em caso de dúvida de interpretação, utilize as aulas, o horário de dúvidas ou o canal #avaliações no Slack da UC para esclarecer as suas questões. O trabalho será também discutido nas aulas teóricas.
- Se no momento da entrega o trabalho não estiver a compilar e pretende que este seja considerado para avaliação, deverá remover todo o código que provoca erros e submeter uma versão que compile. Para evitar esta situação, termine a implementação do seu programa dias antes da data limite para entrega e não deixe a submissão para o último dia. Note, no entanto, que a nota máxima que poderá obter será a nota atribuída pelo Mooshak, mesmo que seja zero.
- Inicie a resolução deste trabalho assim que possível. Deixar a realização do projecto para os últimos dias é desaconselhado, pois poderá ficar sem tempo para resolver problemas que surjam.
- Deve testar bem o programa (utilizando também casos limite) de forma a detectar erros antes da submissão. Só porque um programa funciona para alguns exemplos não é prova irrefutável de que funciona para todos. Lembre-se:

Ao testar um programa conseguimos mostrar a presença de erros mas não a sua ausência.²

²*Program testing can be used to show the presence of bugs but never to show their absence!*, Edsger Dijkstra em <https://www.cs.utexas.edu/users/EWD/transcriptions/EWD03xx/EWD303.html>