

Trabalho I de Lógica para Computação

Prof. Dr. Alexandre Arruda

Considerações sobre o Trabalho

- Máximo de integrantes na equipe: 4 pessoas;
- Cada equipe deverá apresentar um relatório com a metodologia, código e resultados obtidos;
- Data de entrega: 5 de novembro.

UNIVERSIDADE CANGURU

Na cidade de Cangurussas há uma universidade. Nela, todo semestre um quadro composto por horários e disciplinas é fornecido para todo meio acadêmico, onde cangurus professores e alunos se organizam todos os semestres para lecionar e cursar as disciplinas. A universidade oferece dois cursos: um de engenharia de software e outro de ciência da computação.

Em semestres anteriores a organização das disciplinas era feita manualmente, o que rendia grandes dores de cabeça. No ano de 2019, o novo coordenador do curso de ciência da computação, o professor canguru Alexandre, propôs uma solução de alocação de disciplinas de forma automática, utilizando o problema da satisfazibilidade da lógica proposicional.

QUANTO AO PROBLEMA

Resolver o problema da alocação de disciplinas e horários significa, dentre outros planejamentos, que:

1. Cada disciplina é ministrada no máximo 2 vezes por semana (conforme ANEXO I);
2. Cada aula tem 2 horas de duração. Elas podem ocorrer nos seguintes horários: 08:00-10:00, 10:00-12:00, 13:30-15:30 e 15:30-17:30, no período de segunda à sexta;
3. As disciplinas, quando ocorrem duas vezes por semana, são pareadas, ou seja, por exemplo, se a primeira aula é segunda-feira das 10 às 12, a outra aula será quarta-feira das 10 às 12, se a primeira aula for terça-feira das 8 às 10, então a próxima aula será na quinta-feira das 8 às 10, se uma aula ocorrer na quarta, em um horário X, então a outra aula será ou na segunda ou na sexta no mesmo horário X;
4. O curso é dividido por semestres (coincidentemente segue a grade curricular do campus da UFC de Russas). Disciplinas no mesmo semestre não podem ter colisão de horário já que os cangurus alunos querem terminar o curso no tempo certo;
 - a. Grade do curso de ciência da computação:
<http://www.campusrussas.ufc.br/grades/FluxogramadaIntegralizaçãoCurricularCC2018.pdf>

b. Grade do curso de engenharia de software:
<http://www.campusrussas.ufc.br/docs/PPC-ES-Russas-Fluxograma-Obrigat6rias.pdf>

5. Algumas disciplinas são compartilhadas entre os dois cursos. Isso significa que elas não podem ter colisão com as disciplinas que são ofertadas no mesmo semestre nos dois cursos. Por exemplo, se Projeto e Análise de algoritmos for ofertada para os dois cursos, não poderá ter colisão com as outras disciplinas de CC do 4º semestre nem com as de ES do 4º semestre também.
6. Professores cangurus não podem ter o horário conflitante com outras disciplinas que eles lecionam;
7. Professores cangurus gostam de ter ou a segunda-feira livre ou a sexta-feira livre, mas não ambos, para poderem realizar suas pesquisas sobre saltos.

QUANTO A SOLUÇÃO

1. Seu programa deve solucionar o problema de alocação respeitando os itens anteriores;
2. A equipe deverá resolver o problema utilizando um satsolver (ANEXO II) e executá-lo seguindo as instruções (também em ANEXOII);
3. Uma planilha com as alocações por semestre deverá ser feita por cada equipe (pode ser feita manualmente).

DICAS

Crie predicados como:

alexandre_logica-cc
marcio_paa-cces
beatriz_mansoft-es

que serão verdadeiros se o professor Alexandre de fato ministrar lógica para ciência da computação, marcio ministrar PAA que é ofertada para ciência da computação e software e a professora beatriz ministrar manutenção de software para a engenharia de software.

crie predicados como

logica-cc_ter_H1
logica-es_ter_H2
empreendedorismo-cces_qui_H4

que indica que lógica para CC e ES são ofertadas nas terças no primeiro horário e segundo horário, respectivamente, e que empreendedorismo para CC e ES, juntas, é ofertada no último horário da quinta.

ANEXO I

Quanto a divisão das disciplinas, temos:

1. Professor canguru Alexandre irá ministrar:
 - a. Lógica para Computação (2 vezes na semana) - ES
 - b. Lógica para Computação (2 vezes na semana) - CC
2. Professora canguru Anna Beatriz irá ministrar:
 - a. Arquitetura de Software (2 vezes na semana) - ES
 - b. Introdução à Engenharia de Software (2 vezes na semana) - ES
 - c. Qualidade de Software (2 vezes na semana) - ES
 - d. Projeto de Pesquisa Científica e Tecnológica (1 vez na semana) - (CC e ES)
3. Professor canguru Bonfim irá ministrar:
 - a. Compiladores (2 vezes na semana) - CC
 - b. Teoria da Computação (2 vezes na semana) - CC
 - c. Linguagens Formais e Autômatos (2 vezes na semana) - CC
4. Professor canguru Daniel Márcio irá ministrar:
 - a. Fundamentos de Banco de Dados (2 vezes na semana) - CC
 - b. Fundamentos de Banco de Dados (2 vezes na semana) - ES
 - c. Computação Gráfica (2 vezes na semana) - CC
5. Professor canguru Eurinardo irá ministrar:
 - a. Laboratório de Programação (2 vezes na semana) - CC
 - b. Laboratório de Programação (2 vezes na semana) - ES
 - c. Estrutura de Dados Avançada (2 vezes na semana) - CC
6. Professor canguru Filipe Maciel irá ministrar:
 - a. Desenvolvimento WEB (2 vezes na semana) - CC
 - b. Sistemas Distribuídos (2 vezes na semana) - CC
 - c. Redes de Computadores (2 vezes na semana) - CC
7. Professor canguru Nauber irá ministrar:
 - a. Linguagens de Programação (2 vezes na semana) - ES
 - b. Processos de Software (2 vezes na semana) - ES
 - c. Manutenção de Software (2 vezes na semana) - ES
8. Professor canguru Osvaldo irá ministrar:
 - a. Introdução a Processos e Requisitos de Software (2 vezes na semana) - ES
 - b. Gerência de Projetos de Software (2 vezes na semana) - ES
 - c. Análise e Projeto de Sistemas (2 vezes na semana) - (CC e ES)
9. Professora canguru Josemeire irá ministrar:
 - a. Empreendedorismo (2 vezes na semana) - (CC e ES)
 - b. Ética Profissional (2 vezes na semana) - ES
 - c. Ética Profissional (2 vezes na semana) - CC
10. Professor canguru Márcio Costa irá ministrar:
 - a. Projeto e Análise de Algoritmos (2 vezes na semana) - CC
 - b. Fundamentos de Programação (2 vezes na semana) - CC
 - c. Fundamentos de Programação (2 vezes na semana) - ES
11. Professor canguru João Victor irá ministrar:
 - a. Programação Orientada a objetos (2 vezes na semana) - CC

- b. Programação Orientada a objetos (2 vezes na semana) - ES
 - c. Verificação e Validação (2 vezes na semana) - ES
 - d. Projeto Detalhado de Software (2 vezes na semana) - ES
12. Professor canguru Pablo irá ministrar:
- a. Algoritmos em Grafos (2 vezes na semana) - ES
 - b. Algoritmos em Grafos (2 vezes na semana) - CC
13. Professora canguru Tatiane irá ministrar:
- a. Matemática Computacional (2 vezes na semana) - CC
 - b. Estrutura de Dados (2 vezes na semana) - CC
 - c. Estrutura de Dados (2 vezes na semana) - ES
14. Professora canguru Patrícia irá ministrar:
- a. Requisitos de Software (2 vezes na semana) - ES
 - b. Engenharia de Software (2 vezes na semana) - ES
 - c. Interação Humano-Computador (2 vezes na semana) - ES
 - d. Interação Humano-Computador (2 vezes na semana) - CC
15. Professor canguru Alex Lima irá ministrar:
- a. Sistemas Operacionais (2 vezes na semana) - CC
 - b. Inteligência Artificial (2 vezes na semana) - CC
 - c. Arquitetura de Computadores (2 vezes na semana) - CC
 - d. Arquitetura de Computadores (2 vezes na semana) - ES
16. Professor canguru Gleison irá ministrar:
- a. Matemática Básica (2 vezes na semana) - ES
 - b. Cálculo Diferencial e Integral I (2 vezes na semana) - CC
17. Professor canguru Gastão irá ministrar:
- a. Matemática Básica (2 vezes na semana) - CC
 - b. Álgebra Linear (2 vezes na semana) - CC
18. Professora canguru Rosineide irá ministrar:
- a. Probabilidade e Estatística (2 vezes na semana) - CC
19. Professor canguru Anderson Feitoza irá ministrar:
- a. Matemática Discreta (2 vezes na semana) - CC
 - b. Matemática Discreta (2 vezes na semana) - ES
20. Professora canguru Nilde irá ministrar:
- a. Probabilidade e Estatística (2 vezes na semana) - ES
21. Professor canguru Anderson Magno irá ministrar:
- a. Pré-Cálculo (1 vez na semana) - CC

ANEXO II

INSTALAÇÃO DO SATSOLVER

Para utilização do satsolver o aluno deverá baixar o satsolver GLUCOSE SYRUP seguindo os passos:

1. Baixe o glucose em <https://www.labri.fr/perso/lsimon/downloads/softwares/glucose-syrup-4.1.tgz>
2. Coloque o arquivo glucose-syrup-4.1.tgz em uma pasta no qual o caminho absoluto não possui espaços. Por exemplo, /home/alexandre/ (ao invés de /home/alexandre/Área de Trabalho/).
3. Execute o comando "tar -xzf glucose-syrup-4.1.tgz" para descompactar.
4. Após descompactá-lo, execute o comando "cd /glucose-syrup-4.1/simp".
5. Digite "make rs" (para instalar).
6. Digite: export PATH="\$PATH:[CAMINHO_ABSOLUTO]/glucose-syrup-4.1/simp/" (para adicionar o às variáveis de ambiente). Por exemplo, export PATH="\$PATH:/home/alexandre/glucose-syrup-4.1/simp/".
7. Teste a instalação com o comando "glucose_static --help" no prompt.

ALGUMAS CONSIDERAÇÕES SOBRE O SATSOLVER GLUCOSE SYRUP

O satsolver Glucose utiliza como entrada inteiros, por exemplo, um arquivo de entrada para o salsolver poderia ser:

```
p cnf 4 4
1 -2 0
3 -4 0
3 0
-4 0
```

onde na primeira linha "p cnf 4 4" quer dizer que o problema está em CNF (conjunctive normal form) e utiliza 4 variáveis e 4 cláusulas, respectivamente.

As 4 linhas seguintes representam as cláusulas na forma normal conjuntiva, por exemplo, "1 -2 0" significa que o átomo 1 e a negação do átomo 2 estão em uma mesma disjunção, ou seja, "1 -2 0" = $(1 \vee \neg 2)$. A fórmula acima representa:

$$(1 \vee \neg 2) \wedge (3 \vee \neg 4) \wedge (3) \wedge (\neg 4)$$

Observação: Cada cláusula termina com um 0.

Certamente trabalhar com inteiros ao invés de predicados como "logica-cc_H1" é algo complicado. Para isso é fornecido para este trabalho o programa "converttoCNF.cpp" que permitirá o uso de predicados mais representativos.

INSTALAÇÃO DO PROGRAMA CONVERTTOCNF

1. Baixar o arquivo converttoCNF.cpp
2. Compilar com o comando: g++ -std=c++11 -o converttoCNF converttoCNF.cpp

USO DO PROGRAMA CONVERTTOCNF

Crie um arquivo entrada.txt, no qual você colocará todas as cláusulas, por exemplo, o arquivo poderia ser:

```
//entrada.txt
alexandre_logica-cc
-logica-cc_seg_H1 logica-cc_qua_H1
-logica-cc_ter_H1 logica-cc_ter_H1
```

que seria equivalente à:

```
(alexandre_logica-cc) && (-logica-cc_seg_H1 || logica-cc_qua_H1) && (-logica-cc_ter_H1 ||
logica-cc_ter_H1)
```

Observação: não há necessidade de informar a quantidade de variáveis e nem de cláusulas como é feita na entrada padrão do satsolver glucose.

Para executar basta digitar: ./converttoCNF

1. O resultado, se satisfazível, apresentará as cláusulas positivas verdadeiras (para imprimir as negativas basta descomentar as últimas linhas do arquivo converttoCNF.cpp, mas não há necessidade para este trabalho).
2. UNSAT (o problema é insatisfazível).

Um exemplo de retorno seria:

```
alexandre_logica-cc
logica-cc_qua_H1
```

Que indicaria que o professor canguru Alexandre leciona lógica para CC e um dos horários é na quarta das 08-10.