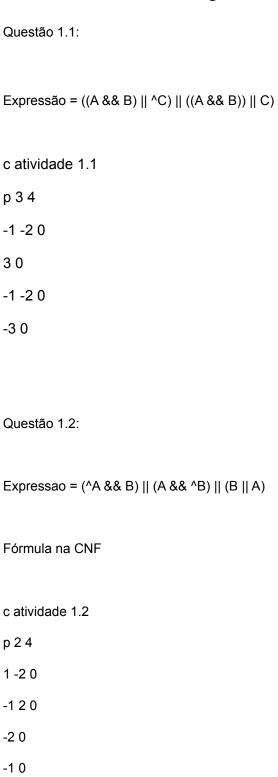
Atividade Prática 2

Rafael Amauri Diniz Augusto - 651047



Questão 2: Resolvedor SAT em C

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <math.h>
#include <time.h>
#define MAX SIZE CONVERTION ARRAY 16
short* dec2bin(int num);
unsigned long long int bin2dec(short *array);
// Converte um numero binario (armazenado em array com tamanho
explicito) pra decimal.
MAX SIZE CONVERTION ARRAY
unsigned long long int bin2dec sized array(short *array, size t size);
int* get combinations(short *mapa expressao, size t
mapa expressao size, short *combinacoes);
int main()
  clock t inicio = clock();
  FILE *fp = fopen("dummy.cnf", "r");
  char *linha = malloc(sizeof(char) * 512);
  const char separator[] = " ";
  char *linha splitted;
```

```
perror("Erro ao ler o arquivo\n");
       exit(EXIT FAILURE);
  bool not parsed = true;
  while(not_parsed)
       fgets(linha, 500, fp);
       if(linha[0] == 'c')
       if(linha[0] == 'p')
           linha splitted = strtok(linha, separator);
           while(linha splitted != NULL)
               if(atoi(linha splitted))
                   num vars = atoi(linha splitted);
                   linha splitted = strtok(NULL, separator);
                   num clausulas = atoi(linha splitted);
               linha splitted = strtok(NULL, separator);
           not parsed = false;
  printf("N de vars = %d /// N de clausulas = %d\n\n", num vars,
num clausulas);
declarado um vetor com esse numero de combinacoes
```

```
short *combinacoes = (short *) calloc(pow(2, num_vars),
sizeof(short));
ficam os valores definidos por cada expressão
   short *mapa_expressao = (short*) malloc((num_vars) * sizeof(short));
   short aux;
  memset(mapa expressao, -1, num vars * sizeof(short));
  not parsed = true;
   for(int i = 0; i < num clausulas; i++)</pre>
       if(fgets(linha, 500, fp) != NULL)
           linha_splitted = strtok(linha, separator);
           while(not parsed)
               if(linha splitted[0] == '0')
                   not parsed = 0;
               aux = atoi(linha splitted);
               if(aux < 0)
                   mapa expressao[abs(aux)-1] = 1;
```

```
mapa expressao[aux-1] = 0;
              linha splitted = strtok(NULL, separator);
           not parsed = 1;
mapa expressao), pegar todas as combinacoes possiveis
           get_combinations(mapa_expressao, num_vars, combinacoes);
          memset(mapa expressao, -1, num_vars * sizeof(short));
          printf("Terminei de processar a clausula %d\n\n", i+1);
expressões. Se houver, é SAT. Se não houver, é UNSAT
   for (unsigned long long int i = 0; i < pow(2, num vars); i++)
      if(combinacoes[i] == 0)
  printf("\n");
      printf("SAT\n");
      printf("UNSAT\n");
```

```
fclose(fp);
  clock t fim = clock();
  printf("\nO tempo de execucao do codigo foi de: %lf segundos\n\n",
int* get combinations(short *mapa expressao, size t
mapa_expressao_size, short *combinacoes)
  for(int i = 0; i < mapa expressao size; i++)</pre>
       if (mapa expressao[i] == -1)
           aux++;
  short *posicoes nao afetadas = (short *)malloc(sizeof(short) * aux);
   for(int i = 0; i < mapa expressao size; i++)</pre>
       if (mapa_expressao[i] == -1)
           posicoes nao afetadas[aux] = i;
           aux++;
   for (int i = 0; i < aux; i++)
```

```
array aux2[i] = 1;
  for(int i = 0; i <= bin2dec sized array(array aux2, aux); i++)</pre>
       short *f = dec2bin(i);
MAX SIZE CONVERTION ARRAY - aux; j--)
mapa expressao[posicoes nao afetadas[MAX SIZE CONVERTION ARRAY - j -
1]] = f[j];
       combinacoes[bin2dec sized array(mapa expressao,
mapa expressao size)] = 1;
       free(f);
short* dec2bin(int num)
  short temp array[MAX SIZE CONVERTION ARRAY], i, aux = 0;
   short *fixed = (short*) calloc(MAX SIZE CONVERTION ARRAY,
sizeof(short));
       temp_array[i] = num % 2;
       fixed[MAX SIZE CONVERTION ARRAY - 1 - i] = temp array[i];
       aux++;
```

```
return fixed;
unsigned long long int bin2dec(short *array)
       if(array[i] == 1)
           a += pow(2, MAX_SIZE_CONVERTION_ARRAY - 1 - i);
  return a;
unsigned long long int bin2dec sized array(short *array, size t size)
       if(array[i] == 1)
          a += pow(2, size - i - 1);
  return a;
```

Questão 3.1: Cláusula 3.1

Cláusula H: UNSAT Cláusula -H: UNSAT

Cláusula I: SAT Cláusula -I: SAT