Trabalho Prático - Algoritmo de Horn:

Começámos por percorrer o standard input de caracter a caracter até encontrarmos uma nova linha, em qual momento processamos o input segundo o Algoritmo de Horn e escrevemos o output de acordo com o tipo de fórmula (Não formula de Horn (NA), contraditória (UNSAT) e possível (SAT)). Fazemos isto até encontrarmos o End-of-File, em qual momento paramos o programa.

Para aplicar o algoritmo, começámos por dividir a nossa fórmula num array de cláusulas, que estavam separadas por um '&'.

Procedemos, pois, à remoção de literais positivos repetidos em cada cláusula. Depois, verificámos se cada cláusula tinha no máximo um literal positivo; caso não tivesse, concluiríamos que não era fórmula de Horn; caso contrário continuávamos com o algoritmo.

Transformámos, então, todas as cláusulas em implicações, para a identificação dos literais positivos e negativos, de modo a facilitar a conseguinte procura do BOTTOM.

De seguida, percorremos as cláusulas à procura de literais do lado esquerdo da implicação que já se encontrassem no nosso conjunto de proposições (inicialmente apenas com o TOP), introduzindo os que estão à direita desta no nosso conjunto caso seja possível. Como otimização, em cada iteração testamos a existência de BOTTOM no nosso conjunto, saindo imediatamente da função caso exista, visto que sabemos que é contraditória; caso não, somos obrigados a esperar que mais nenhuma proposição consiga ser inserida no nosso conjunto, pelo que, se não houver BOTTOM, concluímos que a fórmula é possível.

Tivemos o cuidado de nos preparar contra memory leaks, buffer overflow e qualquer tipo de erro/warning. Dentro do source code serão encontrados inúmeros comentários em Inglês (por forma a tornar o programa mais global), que explicam minuciosamente cada função e auxiliam o leitor com a sua leitura.