

INSTITUTO SUPERIOR TÉCNICO



TÉCNICO
LISBOA

IASD

INTELIGÊNCIA ARTIFICIAL E SISTEMAS DE DECISÃO

Laboratório Nr. 2

Autores:

Eduardo Lima Simões da Silva, 69916

José Pedro Braga de Carvalho Vieira Pereira, 70369

21 de Novembro de 2014

Contents

1	Introdução	2
2	Algoritmo	2
3	Conclusão	3

1 Introdução

Este relatório apresenta a resolução do problema proposto como o segundo laboratório de IASD, no qual se pretende criar um "CNF converter". O objectivo deste laboratório é criar um programa que converte frases logicas no formato de logica proposicional para CNF(Clausal normal form), com um conjunto de funções descritas no enunciado.

2 Algoritmo

A primeira acção que o algoritmo executa é ler o ficheiro onde as diversas frases estão escritas, a maneira como esta acção é feita é um elemento chave para a execução do restante algoritmo. É também relevante salientar a importância do formato com que as diversas frases são escritas pois se o ficheiro não apresentar o formato standard um erro na execução ocorre.

conversão de tuple para lista

Após as diversas frases estarem presentes numa lista o processo de conversão inicia-se. Este processo consiste numa sequência de passos que foram implementados através de funções individuais(para cada passo) onde se fez recurso de uma implementação recursiva. O processo de conversão pode ser dividido, sendo de seguida apresentadas as diversas acções realizadas:

Funções criadas para a detecção do tipo de logica

Separar o ficheiro das funções do ficheiro da main

1. Eliminar equivalências(\Leftrightarrow), onde $(A \Leftrightarrow B)$ fica $(A \Rightarrow B) \wedge (B \Rightarrow A)$. Com o objectivo de executar esta acção a função `solve_equivalence` foi implementada, nela uma frase (lista de entrada) é processada. Se o primeiro elemento da lista de entrada é um equivalente (\Leftrightarrow) a função cria duas clausulas com o primeiro elemento igual a implicação(\Rightarrow) e os elementos da frase logica tal como mostrado previamente. É importante também salientar que os elementos escritos como argumentos da implicação são também processados pela função em si(processo recursivo), a função retorna uma lista com as implicações geradas. Se o primeiro elemento for outra operação lógica a função é chamada de novo com o argumento de entrada igual aos argumentos da operação logica, retornando depois a mesma frase logica mas com os seus argumentos processados, pois dentro destes pode haver uma equivalência;
2. Eliminar implicações(\Rightarrow), onde $(A \Rightarrow B)$ fica $(\neg A \vee B)$. Para resolver esta situação foi implementada outra função com o nome de `solve_implication` muito semelhante à mencionada anteriormente. A maior diferença encontra-se no facto de ao analisar uma frase com uma implicação no primeiro argumento o retorno da função ser a disjunção da negação do primeiro argumento com o segundo argumento. Nesta função todos os argumentos ;
3. Colocar negações(\neg) para o interior, onde $\neg(A \vee B)$ fica $(\neg A \wedge \neg B)$. À semelhança dos passos anteriores para este uma função, `solve_negation`, também foi criada com o objectivo de passar as negações provenientes da acção anterior para o interior das frases que lhe dizem respeito. Para tal.... ;
4. Aplicar a regra distributiva, onde para os casos mais simples $(A \wedge B) \vee E$ fica $(A \vee E) \wedge (B \vee E)$ mas facilmente casos mais complexos são gerados como $(A \wedge E) \vee (B \wedge D)$ que fica

$(A \vee B) \wedge (A \vee D) \wedge (E \vee B) \wedge (E \vee D)$. Este facto fez com que a função `solve_disjunction` tenha sido especialmente desafiante de desenvolver.;

5. Passar para clausulas;

Após estes quatro passos é possível converter o conjunto de frases lógicas num conjunto de cláusulas. Sendo ainda necessário eliminar possíveis repetições e executar simplificações. Com esse objectivo realizaram-se os seguintes passos:

1. Eliminar impossíveis, condições unitarias etc etc

Referir menu criado e diversas opções

Escrita dos resultados

3 Conclusão

Successo??

foram todas as funções implementadas??

dificuldades encontradas??

pontos a melhorar??

References

- [1] Luís Custódio, Rodrigo Ventura, North tower – ISR / IST, 2014, *Lecture notes - Course Artificial Intelligence and Decision Systems*
- [2] Stuart Russell, Peter Norvig , 2003, Prentice Hall, Second Edition, *Artificial Intelligence: A Modern Approach*