Lógica Computacional

Conectores Booleanos

Negação, Conjunção e Disjunção

Tradução de Linguagem Natural

Fórmulas de 1ª ordem – Definição indutiva

Conectores Booleanos - Negação

- A linguagem de 1ª ordem utilizada até agora só permite exprimir **proposições simples**. Mas a língua natural permite expressões arbitrariamente complexas.
- Quando se pretende negar uma afirmação utiliza-se normalmente o *advérbio de negação* **não**.
- Gramaticalmente um advérbio é uma palavra que altera o significado de um verbo. Por exemplo, na expressão “voar depressa”, o advérbio “depressa” qualifica o forma de voar – não é um voo normal, nem lento, é um voo rápido.
- A negação qualifica uma acção (verbo), contrariando-a completamente. Se a acção é verdadeira depois de negada torna-se falsa e vice-versa.

Exemplo:

1. O Carlos conhece a Rita.
2. O Carlos não conhece a Rita.

- As Linguagens de 1ª Ordem exprimir a negação através do símbolo de negação “**¬**”.

1. Conhece(carlos, rita).
2. ¬Conhece(carlos, rita).

Conectores Booleanos - Negação

- Quando a negação é muito “comum” a linguagem natural contem verbos que já incluem a noção de negação na sua formação, normalmente através de prefixos como “in” ou “des”.

Exemplo:

1. O Carlos não conhece a Rita.
2. O Carlos desconhece a Rita.

- Apesar de se poderem considerar diferentes graduações nestas afirmações (a 1ª será mais “forte” que a 2ª - para algumas pessoas em certos contextos), ambas se podem/devem exprimir através do símbolo de negação “¬”.

1. ¬Conhece(carlos, rita).

Conectores Booleanos - Negação

- Podemos ainda comparar as frases abaixo e verificar que elas têm todas o mesmo significado embora com graduações ligeiramente diferentes.

Exemplo:

1. O Carlos conhece a Rita.
2. Não é verdade que o Carlos desconheça a Rita.
3. Não é verdade que o Carlos não conheça a Rita.

... o que conduz a que se atribua o mesmo valor de verdade a frases duplamente negadas.

1. Conhece(carlos, rita).
2. $\neg(\neg$Conhece(carlos, rita)) $\equiv$ $\neg\neg$Conhece(carlos, rita).
3. $\neg(\neg$Conhece(carlos, rita)) $\equiv$ $\neg\neg$Conhece(carlos, rita).

- Este encadeamento pode ser arbitrariamente complexo
  - Não é verdade que não seja verdade que o Carlos desconheça a Rita.
  - $\neg\neg\neg$Conhece(carlos, rita).

Conectores Booleanos - Conjunção

- Outra forma de obter proposições mais complexas a partir de proposição elementares é a através da sua **conjunção**.

Exemplo:

- O Carlos conhece a Rita e o Jorge conhece a Maria.

- Claramente esta frase inclui duas frases mais simples que podem ser agregadas através do símbolo de conjunção "**∧**".

- Conhece(carlos, rita) ∧ Conhece(jorge, maria) .

- A linguagem natural factoriza frequentemente várias acções em frases "simples", apresentando-as aparente como uma só acção. Mas na sua tradução para uma linguagem de 1ª ordem, a conjunção é sempre entre duas proposições!

Exemplo:

- O Carlos conhece a Rita e a Maria.
- Conhece(carlos, rita) ∧ Conhece(carlos, maria) .... mas não
- Conhece(carlos, rita ∧ maria) ... nem
- Conhece(carlos, rita , maria)

Conectores Booleanos - Conjunção

- A conjunção em língua natural permite ainda exprimir alguma informação adicional.

Exemplo:

- O Carlos encontrou-se com a Rita **e** foram jantar.
- O Carlos encontrou-se com a Rita **mas** não foram jantar

- Tal como anteriormente as frases podem ser decompostas em frases mais simples através do símbolo de conjunção "$\wedge$" e da negação "$\neg$".
  - Encontro(carlos, rita) $\wedge$ jantar(carlos, rita) .
  - Encontro(carlos, rita) $\wedge$ $\neg$ jantar(carlos, rita) .

- No primeiro caso existe alguma informação temporal implícita – o encontro ocorreu antes do jantar.

- No segundo caso, a conjunção adversativa permite antecipar um elemento de surpresa (neste caso a negação de algo que seria esperado no contexto). Mas tal como a conjunção coordenativa, também a conjunção adversativa é traduzida através do operador de conjunção "$\wedge$".

Conectores Booleanos - Disjunção

- Outra forma de obter proposições mais complexas a partir de proposição elementares é a através da sua **disjunção**.

Exemplo:

- O Carlos come a sopa ou a salada.

- A disjunção , expressa pela conjunção disjuntiva "ou", é expressa na linguagem de 1ª ordem pelo símbolo "**v**".
  - Come(carlos, sopa) v Come(carlos, salada) .

- Tal como na conjunção, a linguagem natural permite a factorização de frases simples, mas na sua tradução para uma linguagem de 1ª ordem, a disjunção é sempre entre duas proposições!

Exemplo:

- Come(carlos, sopa) v Come(carlos, salada) . .... mas não
- Come(carlos, sopa v salada) ... nem
- Come(carlos, sopa , salada)

Conectores Booleanos - Disjunção

- Um aspecto importante nesta tradução é distinguir entre a disjunção **inclusiva** e a **exclusiva**. No primeiro caso, ambas as frases disjuntas podem ser verdadeiras, enquanto que no segundo caso não se aceita a sua verdade simultânea.

Exemplo:

- O Carlos come a sopa ou a salada.
- O Carlos vai ao cinema ou ao teatro.

- A diferença muitas vezes apenas se pode verificar pelo contexto – uma forma de as distinguir é acrescentar a frase "... , mas pode fazer ambos" (disjunção inclusiva) ou "... , mas **não** pode fazer ambos" (disjunção exclusiva)

Exemplo:

- O Carlos come a sopa ou a salada, ... mas pode comer ambas.
- O Carlos vai ao cinema ou ao teatro, ... mas não pode ir aos dois.

- **Nota importante:** O símbolo de disjunção "v" é usado para a tradução da disjunção **inclusiva**. A disjunção **exclusiva** pode ser obtida através da disjunção, conjunção e negação (embora também se possa usar um conector booleano - que não usaremos).

Conectores Booleanos – Leis de de Morgan

- É muito normal juntarmos negação, conjunção e disjunção em frases complexas.

Exemplo:

1. O Carlos nem come a sopa nem a salada.
2. É falso que o Carlos coma a sopa ou a salada.

- Estas duas frases têm o mesmo significado mas a sua tradução é feita através de uma dferente combinação de operadores Booleanos.

Exemplo:

1. ¬ Come(carlos, sopa) ∧ ¬ Come(carlos, salada) .
2. ¬ **(**Come(carlos, sopa) ∨ Come(carlos, salada)) .

- De notar a **precedência** dos operadores Booleanos. Em rigor, para evitar ambiguidades deveríamos sempre aplicar os operadores a frases “parentisadas”, mas muitos parênteses podem ser evitados se se considerarem as habituais relações de precedência entre operadores (¬**,** ∧ e ∨, por esta ordem)

1. **(**¬ **(**Come(carlos, sopa))) ∧ (¬ **(**Come(carlos, salada))).
2. ¬ **(**Come(carlos, sopa) ∨ Come(carlos, salada)) .

Conectores Booleanos – Leis de de Morgan

- Uma composição semelhante de frases mais simples é exemplificada abaixo.

Exemplo:

1. O Carlos não come a sopa ou (não come) a salada.
2. É falso que o Carlos coma a sopa e a salada.

- Estas duas frases têm o mesmo significado mas a sua tradução é feita através de uma diferente combinação de operadores Booleanos.

Exemplo:

1. $\neg$ Come(carlos, sopa) $\vee$ $\neg$ Come(carlos, salada) .

2. $\neg$ **(**Come(carlos, sopa) $\wedge$ Come(carlos, salada)) .

Os exemplos anteriores correspondem às conhecidas leis de de Morgan:

- A negação da disjunção tem o mesmo significado da conjunção das negações.
- A negação da conjunção tem o mesmo significado da disjunção das negações.

Fórmulas de 1ª Ordem – Definição Indutiva

- Podemos agora apresentar uma definição formal do que são fórmulas de 1ª ordem (FPO). Elas podem ser arbitrariamente complexas, e podem ser caracterizadas através de uma definição indutiva, isto é, o conjunto de FPOs (fórmulas de 1ª ordem) constitui uma estrutura indutiva.
- Desta muitas propriedades das fórmulas, nomeadamente sintáticas, podem ser formalmente inferidas através de indução estrutural, apresentada anteriormente.
- Como com qualquer estrutura indutiva, a definição do conjunto de fórmulas de 1ª ordem, é composto por 3 partes:
  1. Uma cláusula de base, que especifica os elementos básicos do conjunto
  2. Uma ou mais cláusulas indutivas que explicam como se podem geral elementos do conjunto a partir de elementos conhecidos.
  3. Uma cláusula de fecho, que indica que todos os elementos são obtidos a partir das cláusulas anteriores.

Fórmulas de 1ª Ordem – Definição Indutiva

1. **Cláusula de Base**

Os elementos mais simples das FPOs são as proposições simples, constuídas pelos predicados aplicados a constantes ou funções definidas numa assinatura **Σ**.

2. **Cláusulas Indutivas**

Se P e Q são FPOs, as seguintes fórmulas são ambas FPOs

   i. $(\neg P)$ é uma FPO

   ii. $(P \wedge Q)$ é uma FPO

   iii. $(P \vee Q)$ é uma FPO

3. **Cláusula de Fecho**

- Apenas as fórmulas obtidas pelas cláusulas anteriores são FPOs

Nota: A definição exige que todas as sub-fórmulas não simples sejam "parentisadas". Na prática, alguns parênteses podem ser eliminados se se adoptarem as regras de precedência.

Fórmulas de 1ª Ordem - Definição Indutiva

Exemplo

A formula abaixo é uma FPO.

(TemFome(carlos) ∧((¬Come(carlos,sopa))∨(¬Come(carlos, salada)))).

De facto, a fórmula é do tipo

$$(P \land ((\neg Q) \lor (\neg R))),$$

... em que

P = **TemFome(carlos)**.

Q = **Come(carlos, sopa)**.

R = **Come(carlos, salada)**.

... são proposições simples da linguagem com assinatura **Σ = SF** ∪ **SP**, em que :

SF = {carlos, sopa, salada}

SP = {TemFome/1, Come/2)

Fórmulas de 1ª Ordem - Definição Indutiva

Exemplo: **`F1. (P ∧ ( (¬Q) ∨ (¬R)))`**

A fórmula **`F1`** é uma FPO se forem FPOs as fórmulas **`F2`** e **`F3`** abaixo, (regra 2.ii: **`∧)`**

`F2. P` , e

`F3. ((¬Q) ∨ (¬R))`

A fórmula **`F2`** é uma proposição simples, logo é uma FPO (regra 1). A fórmula **`F3`** é uma FPO se forem FPOs as formulas **`F4`** e **`F5`** abaixo (regra 2.iii: **`∨).`**

`F4. ¬Q` , e

`F5. ¬R`

As fórmula **`F4`** e **`F5`** são FPOs se forem FPOs **`F6`** e **`F7`** abaixo (regra 2.i: **`¬).`**

`F6. Q` , e

`F7. R`

Mas as fórmulas **`F6`** e **`F7`** são proposições simples, logo FPOs (regra 1).

Fica assim verificado que F1 é uma FPO para a assinatura **Σ = SF** ∪ **SP** indicada.

SP = **`{TemFome/1, Come/2)`** **SF =** **`{carlos, sopa, salada}`**

Fórmulas de 1ª Ordem - Definição Indutiva

Exemplo: **`F1. (P ∧ ( (¬Q) ∨ (¬R)))`**

`( TemFome(carlos) ∧((¬Come(carlos,sopa))∨(¬Come(carlos, salada)))).`

A estrutura indutiva das FPOs é bem patente na sua representação em árvore (em informática as árvores são apresentadas “de cabeça para baixo”!).