# Теория

## Обща резолюция

*Дизюнкт* наричаме множество от литерали. Казваме, че два дизюнкта D1 и D2 *имат непосредствена резолвента*, ако има атом A, така че [A\in D_1](http://moodle.openfmi.net/filter/tex/displaytex.php?A\in+D_1), [\neg A\in D_2](http://moodle.openfmi.net/filter/tex/displaytex.php?\neg+A\in+D_2). *Непосредствена резолвента* на D1 и D2 наричаме дизюнкта D = D1 U D2 \ {A,!A}. Ако [\sigma](http://moodle.openfmi.net/filter/tex/displaytex.php?\sigma)е субституция, [D\sigma = \{ L\sigma | L\in D \}](http://moodle.openfmi.net/filter/tex/displaytex.php?D\sigma+=+\%7b+L\sigma+|+L\in+D+\%7d), т.е. прилагаме субституцията над всички елементи на дизюнкта. [D\sigma](http://moodle.openfmi.net/filter/tex/displaytex.php?D\sigma)наричаме *частен случай* на D. *Резолвента* на два дизюнкта D1 и D2 наричаме някоя непосредствена резолвента на частни случаи на D1 и D2.

Най-добре е като вземаме обща резолвента да търсим възможно най-обща резолвента. Затова е добре да преименуваме променливите, така че в дизюнктите, които резолвираме да няма общи променливи преди да започнем да унифицираме съответните атоми. Например ако не преименуваме общата резолвента на  D1 = { p(X), q(Y,f(X)) } и D2 = { !p(T), r(g(X,Y)) } е D' = { q(Y,f(X)), r(g(X,Y)) }, докато с предварително преименуваме бихме могли да получум доста по-общата резолвента D'' = { q(Y',f(X')), r(g(X,Y)) }.

Дизюнктите съответстват смислово на дизюнкции от литерали. В този смисъл празното множество (празният дизюнкт) е тъждествената лъжа.

## Алгоритъм за получаване на множество от дизюнкти от логическа формула:

1. Привеждаме формулата в скулемова нормална форма.
2. Изтриваме кванторите [\forall](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall).
3. Привеждаме в конюнктивна нормална форма с помощта на правилото A v (B & C) = (A v B) & (A v C). Казваме, че една безкванторна формула е в *конюнктивна нормална форма*, ако тя е конюнкция от дизюнкции от литерали. В частност можем да имаме само една дизюнкция
4. Всяка дизюнкция записваме като множество от нейните литерали (дизюнкт).

Казваме, че едно множество от дизюнкти {D1,..., Dn} е *съвместимо* (*несъвместимо*), ако формулата D1 & .. & Dn е изпълнима (неизпълнима), т.е. има (няма) модел и оценка, при които формулата да е вярна.

Методът на общата резолюция е метод за доказване на несъвместимост на множество от дизюнкти. Всяка резолвента на два дизюнкта е логическо следствие от тези два дизюнкта. Затова ако с помощта на вземане на резолвенти успеем да получим празния дизюнкт (противоречие), тогава множеството от което сме тръгнали е несъвместимо.

Методът на общата резолюция не е програмируем, защото нямаме общо правило, което да ни казва на всяка стъпка кои два дизюнкта да резолвираме. Нещо повече за всяко програмируемо правило може да се намери за контрапример подходящо множество от дизюнкти, за което правилото да не работи.

Дизюнктите можем условно да разделим на 4 типа:

1. без положителен литерал (само отрицателни)
2. само един положителен литерал (и без други)
3. един положителен литерал и още отрицателни
4. с поне два положителни литерала

Тип 2 съответства на факт в ПРОЛОГ - A.  
Тип 3 съответства на правило в ПРОЛОГ:

B :- A1,...,An = (A1 & ... & An) -> B = !A1 v ... v !An v B.

Тип 2 и Тип 3 се наричат заедно *Хорнови клаузи*.

Тип 1 съответства на отрицание на цел в ПРОЛОГ, т.е. отрицание на конюнкция от атоми (*Хорнова цел*):

!(A1 & ... & An) = !A1 v ... v !An

Типове 1, 2 и 3 се наричат заедно *Хорнови формули*. ПРОЛОГ решава с помощта на метода на резолюцията задачата дали от една програма от Хорнови клаузи следва логически Хорнова цел и ако да, при каква субституция. Дизюнкти от тип 4 не могат да се зададат на ПРОЛОГ - те задават дизюнктивно знание от типа: "днес ще вали или ще е облачно".

Задачата за логическо следване може да се реши с метода за резолюцията като се сведе до задача за доказване на неизпълнимост: по следния начин:

От формулите F1,...,Fn следва логически G <-> формулата (F1 & ... & Fn -> G) е тавтология <-> формулата !(F1 & ... & Fn -> G) = F1 & ... & Fn & !G е неизпълнима.

Така получаваме следния *метод на общата резолюция* за доказване на логическо следване*:*

1. Получаваме дизюнктите съответстващи на формулите F1,...,Fn , G.
2. Опитваме се с помощта на резолвенти да достигнем до празния дизюнкт.

ПРОЛОГ използва следното програмируемо правило за избор на дизюнкти за резолвента: винаги се резолвира дизюнкт от тип 1 (целта) с първия дизюнкт от тип 2 или 3, който има положителен литерал, съответстващ на първия отрицателен литерал в целта. Това правило, както ни е известно от факта, че има забиващи ПРОЛОГ програми, не винаги дава резултат.

# Задачи

**Задача 1.** "Всеки който може да чете е грамотен. Маймуните не са грамотни, но някои от тях са интелигентни. Докажете, че някой който е интелигентен не може да чете." Формулирайте задачата във формули и я докажете с метода на общата резолюция.

**Задача 2.** С метода на общата резолюция, докажете, че от формулите F1, F2 (и евентуално F3) логически следва формулата G:

1. F1: [\forall Y(\exists X(p(X,Y))\rightarrow \neg s(Y,Y))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+Y(\exists+X(p(X,Y))\rightarrow+\neg+s(Y,Y)))  
   F2: [\forall X\exists Y(s(Y,Y)\&amp; \forall Z(q(Y,Z))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\exists+Y(s(Y,Y)\&amp;amp;+\forall+Z(q(Y,Z)))  
   F3: [\forall X\exists Y(r(X,X)\&amp; q(X,Y)\rightarrow p(Y,X))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\exists+Y(r(X,X)\&amp;amp;+q(X,Y)\rightarrow+p(Y,X)))

G: [\exists X\exists Y(\neg(s(X,Y)\rightarrow r(X,Y)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\exists+X\exists+Y(\neg(s(X,Y)\rightarrow+r(X,Y))))

1. F1: [\forall X\forall Y\exists Z((q(Z)\rightarrow p(X,Y)) \&amp; \forall N(p(N,Z)\rightarrow r(X,N)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\forall+Y\exists+Z((q(Z)\rightarrow+p(X,Y))+\&amp;amp;+\forall+N(p(N,Z)\rightarrow+r(X,N))))  
   F2: [\forall X\exists Y(\neg p(X,Y) \&amp; \forall Z(\neg p(Y,Z)\rightarrow q(Z)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\exists+Y(\neg+p(X,Y)+\&amp;amp;+\forall+Z(\neg+p(Y,Z)\rightarrow+q(Z))))

G: [\exists X r(X,X)](http://moodle.openfmi.net/filter/tex/displaytex.php?\exists+X+r(X,X))

1. F1: [\forall X\exists Y\forall Z((q(Y)\rightarrow p(X,a)) \&amp; p(Z,a)\rightarrow r(X,Z)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\exists+Y\forall+Z((q(Y)\rightarrow+p(X,a))+\&amp;amp;+p(Z,a)\rightarrow+r(X,Z))))  
   F2: [\forall X\forall Y\forall Z(\neg p(Y,X) \&amp; (\neg p(Y,Z)\rightarrow q(Z)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\forall+Y\forall+Z(\neg+p(Y,X)+\&amp;amp;+(\neg+p(Y,Z)\rightarrow+q(Z))))

G: [\exists X r(X,X)](http://moodle.openfmi.net/filter/tex/displaytex.php?\exists+X+r(X,X))

1. F1: [\exists X(p(X,f(X)) \&amp; \neg p(f(X),X))](http://moodle.openfmi.net/filter/tex/displaytex.php?\exists+X(p(X,f(X))+\&amp;amp;+\neg+p(f(X),X)))  
   F2: [\forall X\forall Y(p(X,Y)\rightarrow p(f(X),Y))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\forall+Y(p(X,Y)\rightarrow+p(f(X),Y)))

G: [\exists X\exists Y((p(f(X),X)\vee q(f(X),Y))\rightarrow p(X,X) \&amp; q(X,X)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\exists+X\exists+Y((p(f(X),X)\vee+q(f(X),Y))\rightarrow+p(X,X)+\&amp;amp;+q(X,X))))

1. F1: [\neg\exists X(p(X,X)\&amp; q(X,X))](http://moodle.openfmi.net/filter/tex/displaytex.php?\neg\exists+X(p(X,X)\&amp;amp;+q(X,X)))  
   F2: [\forall X\forall Y((q(X,Y)\rightarrow q(g(X),Y))\&amp; (p(g(X),Y) \vee q(g(X),X)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\forall+Y((q(X,Y)\rightarrow+q(g(X),Y))\&amp;amp;+(p(g(X),Y)+\vee+q(g(X),X))))

G: [\forall X(\neg q(X,g(X))\vee q(g(X),X))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X(\neg+q(X,g(X))\vee+q(g(X),X)))

1. F1: [\forall X\forall Y(p(X,Y)\rightarrow\neg s(Y,Y))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\forall+Y(p(X,Y)\rightarrow\neg+s(Y,Y)))  
   F2: [\exists Y\forall Z(s(Y,Y)\&amp; q(Y,Z)](http://moodle.openfmi.net/filter/tex/displaytex.php?\exists+Y\forall+Z(s(Y,Y)\&amp;amp;+q(Y,Z))  
   F3: [\forall X\exists Y(r(X,X)\&amp; q(X,Y)\rightarrow p(Y,X))](http://moodle.openfmi.net/filter/tex/displaytex.php?\forall+X\exists+Y(r(X,X)\&amp;amp;+q(X,Y)\rightarrow+p(Y,X)))

G: [\exists X\exists Y(\neg(s(X,X)\rightarrow r(X,Y)))](http://moodle.openfmi.net/filter/tex/displaytex.php?\exists+X\exists+Y(\neg(s(X,X)\rightarrow+r(X,Y))))