

به نام خدا

جلسه ی پنجم

به چه شکل میشه گزاره را تبدیل کرد به Clause:

۱. حذف اگر انگاه
۲. کاهش دامنه ی نفی
۳. استاندارد سازی نام متغیر ها: متغیر همنام نداشته باشیم
۴. اسکولم سازی
۵. حذف سور های عمومی
۶. تولید CNF
۷. حذف عطف ها
۸. نامگذاری مجدد متغیر ها

مثال کتاب راسل:

$\text{All } x [\text{All } y \text{ Animal}(y) \Rightarrow \text{Loves}(x, y)] \Rightarrow [\text{Exist } y \text{ Loves}(y, x)]$

- $\Rightarrow 1. \text{All } x [\sim^1 \text{All } y \sim \text{Animal}(y) \text{ or Loves}(x, y)] \text{ or } [\text{Exist } y \text{ Loves}(y, x)]$
- $\Rightarrow 2. \text{All } x [\text{Exist } y (\sim \text{Animal}(y) \text{ or Loves}(x, y)) \text{ or } [\text{Exist } y \text{ Loves}(y, x)]]$
- $\Rightarrow \text{All } x [\text{Exist } y \text{ Animal}(y) \text{ And } \sim \text{Loves}(x, y)] \text{ or } [\text{Exist } y \text{ Loves}(y, x)]$
- $\Rightarrow 3. \text{All } x [\text{Exist } y \text{ Animal}(y) \text{ and } \sim \text{Loves}(x, y)] \text{ or } [\text{Exist } z \text{ Loves}(z, x)]$
- $\Rightarrow 4. \text{All } x [\text{Animal}(F(x)) \text{ And } \sim \text{Loves}(x, F(x))] \text{ or } [\text{Loves}(G(x), x)]$
- $\Rightarrow 5. [\text{Animal}(F(x)) \text{ And } \sim \text{Loves}(x, F(x))] \text{ or } [\text{Loves}(G(x), x)]$
- $\Rightarrow 6. \text{Animal}(F(x)) \text{ or Loves}(G(x), x) \text{ and } [\sim \text{Loves}(x, F(x)) \text{ or Loves}(G(x), x)]$
- $\Rightarrow 7. [\text{Animals}(F(x)) \text{ or Loves}(G(x), x)] , [\sim \text{Loves}(x, F(x) \text{ or Loves}(G(x), x)]$

Commented [AT1]: توزیع عطف روی فصل ها، یعنی باید بین شون عطف باشه

Commented [AT2]: وقتی سور اول گزاره میاد، به معنای این است که روی تمام گزاره اثر داده میشود.

Commented [AT3]: ۲ تا متغیر یک نام داریم. پس تغییر متغیر میدهیم

Commented [AT4]: ۲ تا کلاوز میده بهمون

¹ ~ = Negation = Naghiz

به نام خدا

مثال :

$\text{All } x (\text{All } y P(x, y) \Rightarrow \text{All } y Q(x, y) \Rightarrow R(x, y))$

- $\Rightarrow 1. \text{All } x \sim(\text{All } y P(x, y) \text{ or } \sim(\text{All } y Q(x, y) \text{ or } R(x, y)))$
- $\Rightarrow 2. \text{All } x(\text{Exist } y \sim P(x, y) \text{ or } (\text{Exist } y Q(x, y) \text{ and } \sim R(x, y)))$
- $\Rightarrow 3. \text{All } x(\text{Exist } y \sim P(x, y) \text{ or } (\text{Exist } z Q(x, y) \text{ and } \sim R(x, y)))$
- $\Rightarrow 4. \text{All } x(\sim P(x, F_1(x)) \text{ or } (Q(x, F_2(x)) \text{ and } \sim R(x, F_2(x))))$
- $\Rightarrow 5. (\sim P(x, F_1(x)) \text{ or } (Q(x, F_2(x)) \text{ and } \sim R(x, F_2(x))))$
- $\Rightarrow 6. (\sim P(x, F_1(x)) \text{ or } Q(x, F_2(x)) \text{ and } (\sim P(x, F_1(x)) \text{ or } \sim R(x, F_2(x))))$
- $\Rightarrow 7. \sim P(x, F_1(x)) \text{ or } Q(x, F_2(x)) , \sim P(x, F_1(x)) \text{ or } \sim R(x, F_2(x))$

یونیفیکیشن (Unification) صفحه ی ۳۲۶ کتاب راسل

اتصال (Binding): x/t می نویسیم. به معنی جایگزین نمودن T به جای X در تمامی مکان های وقوع آن است. جایگزینی یا Substitution مجموعه ی منتناهی از اتصالات به صورت $\{x_1/t_1, \dots, x_n/t_n\}$ که در آن

متغیرهای x_1 تا x_n متغیر های مجزا و $\text{All } i (x_i \neq t_i)$

Subst(Theta , Alpha) : به معنی استفاده از جایگزینی تتا و اعمال آن بر آلفا است.

یکسان سازی (یونیفیکیشن): ورودی یکسان سازی، حتما کلاوز است. دو جمله ی ورودی مانند آلفا و بتا را دریافت می‌کنده و جایگزینی همانند تتا تولید میکند که در این جایگزینی آلفا و بتا به یک جمله ی واحد تبدیل خواهد شد. الگوریتم یکسان سازی به صورت زیر است.

به نام خدا

```
Unify( $E_1, E_2$ )
{
    Case Both  $E_1$  and  $E_2$  are Constants or Empty
        If ( $E_1 == E_2$ ) return {};
        Else return fail
    Case  $E_1$  is a variable:
        If ( $E_1$  Occurs in  $E_2$ ) return fail;
        Else return  $\{E_1/E_2\}$ 
    Case  $E_2$  is a variable :
        If ( $E_2$  Occurs in  $E_1$ ) return fail;
        Else return  $\{E_2/E_1\}$ 

    Case
        if ( $E_1 == \{\}$  ||  $E_2 == \{\}$ ) return fail;
    Default :
    {
         $HE_1$  = first element of  $E_1$ ;
         $HE_2$  = first element of  $E_2$ ;
         $Subs1$  = Unify ( $HE_1$ ,  $HE_2$ );
        If ( $subs1 == fail$ ) return fail;
         $TE_1$  = apply( $subs1$ , rest of  $E_1$ )
         $TE_2$  = apply( $subs1$ , rest of  $E_2$ )
         $Subs2$  = unify( $TE_1$ ,  $TE_2$ );
        If ( $Subs2 == fail$ ) return fail;
        Else return Concat( $Subs1, subs2$ )
    }
}
```

$\left\{ \begin{array}{l} \text{Alpha} = P(A, x) \\ \text{Beta} = P(y, B) \end{array} \right.$
 $\Rightarrow \{y/A, x/B\}$

به نام خدا

مثال از کتاب راسل :

$\text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(\text{John}, \text{Jane})) = \{x/\text{Jane}\}$

$\text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(y, \text{Bill})) = \{x/\text{Bill}, y/\text{John}\}$

$\text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(y, \text{Mother}(y))) = \{y/\text{John}, x/\text{Mother}(\text{John})\}$

$\text{UNIFY}(\text{Knows}(\text{John}, x), \text{Knows}(x, \text{Elizabeth})) = \text{fail}.$