

# 函数程序设计实验三

## 一、定义命题类型及其 Show 特例

命题的集合可以如下定义

- 真命题 (记作 T) 和假命题 (记作 F) 是命题;
- 由一个字母表示的命题变元是一个命题;
- 如果 p 是命题, 则它的否定是一个命题, 记作  $\sim p$ ; “ $\sim$ ” 称为否定联结词;
- 如果 p, q 是命题, 则它们的合取是一个命题, 记作  $p \ \&\& \ q$ ; “ $\&\&$ ” 称为合取联结词;
- 如果 p, q 是命题, 则它们的析取是一个命题, 记作  $p \ || \ q$ ; “ $||$ ” 称为析取联结词;
- 如果 p, q 是命题, 则它们的蕴含是一个命题, 记作  $p \ ==> \ q$ ; “ $=>$ ” 称为蕴含联结词;
- 所有的命题由以上规则经过有限步生成。

所以, 命题的集合是一个归纳定义的集合, 可以用代数类型描述:

```
data Prop = Const Bool
          | Var Char
          | Not Prop
          | And Prop Prop
          | Or  Prop Prop
          | Imply Prop Prop
          deriving Eq
```

**习题 1.** 试将类型 Prop 定义成类 Show 的特例, 使得 Prop 中元素能够显示成定义中我们习惯的表示, 例如, Var 'A', And (Var 'P') (Var 'Q') 分别显示成 A 和 P && Q 等。

## 习题 2 定义下列值

```
p1, p2, p3 :: Prop
```

使得它们个表示下列命题:

```
A && ~A
A || ~A
A => (A && B)
```

## 二、命题真值表

一个命题或者真, 或者假。例如, 不包含变元的命题  $T$  表示真命题, 命题  $F$  表示假命题。包含变元的命题的真假依赖于变元表示命题的真假, 所以, 一个命题可以看作它包含的变元的函数, 称为真值函数, 其中每个变元的取值为真或者假, 分别用  $True$  和  $False$  表示。每个命题的取值和其中变元的关系可以用一个表来表示, 称之为命题函数的真值表。以下是命题联结词的真值表:

P	$\sim P$	P	Q	$P \&\& Q$
True	False	True	True	True
True	False	True	False	False
False	True	False	False	False
False	True	False	True	False

  

P	Q	$P    Q$	P	Q	$P \Rightarrow Q$
True	True	True	True	True	True
True	False	True	True	False	False
False	False	False	False	False	True
False	True	True	False	True	True

对于任意命题函数, 给定变元的一个代换, 根据上述真值表可以计算命题相应的真值。例如, 命题  $p3: A \Rightarrow (A \&\& B)$  在代换  $[('A', True), ('B', False)]$  下的真值是  $False$ 。我们可以用下列类型表示代换:

```
type Subst = [(Char, Bool)]
```

## 习题 3 试根据以上解释定义函数:

```
eval :: Subst -> Prop -> Bool
```

例如

```
>eval [( 'A', True), ('B', False)] p3
False
```

**习题 4** 试定义下列函数：

```
vars    :: Prop -> [Char]
subst   :: Prop -> [Subst]
```

其中 `vars p` 给出命题 `p` 中出现的所有不同变元, `subst p` 给出命题 `p` 中变元的所有可能代换。例如

```
vars p3 = ['A', 'B']
subst p3 = [(('A', True), ('B', True)),
             (('A', True), ('B', False)),
             (('A', False), ('B', True)),
             (('A', False), ('B', False))]
```

### 三、定义一个判断命题是否永真的函数

如果一个命题函数在变元的任意代换下真值是 `True`, 则称之为永真式。例如, 命题 `p2` 是永真式。试定义判定一个命题是否永真式的函数, 并说明你的函数定义的正确性:

**习题 5** `isTaut :: Prop -> Bool`

例如

```
>isTaut p1
False
>isTaut p2
True
```

### 四、递交实验要求

- 你提交的报告是包含程序和有关说明的文本文件, 说明包括姓名、email、学号和系别。说明作为注释。文本文件形如

```
-- 王力 200901001, wangli@163.com, 环保系;

-- 其他说明
module Lab3(
    Prop(..),
    isTaut -- :: Prop ->Bool
) where
```

- 使用电子邮件将报告作为附件提交到指定邮箱。主题：函数程序实验三  
学号 姓名。注意截止日期。
- 如有问题，请及时联系老师，并注意课程网页有关信息。
- 你的程序应该能够正常运行，并说明做了哪些测试。如果尚不能运行，说明理由或者困难。
- 实验记入成绩，请认真对待。