# cs113 Lab1::By: Amuldeep Dhillon

## cs113 lab1

Text written to file build.sh

```
doctex lab.doc
pptexenv latex lab.tex
dvipdf lab.dvi
```

Bourne Shell

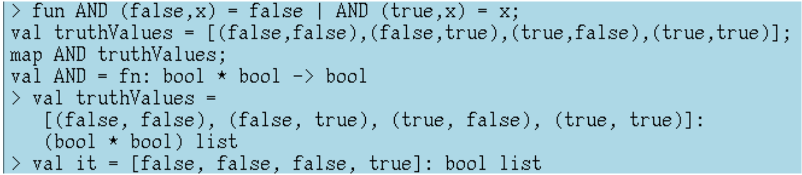
```
chmod 777 build.sh
```

# ch1 Proposiational Calculus (Conjuntion)

Exploring compound propositions

- Conjunction: $p \wedge q$

| $p$ | $q$ | $(p \wedge q)$ |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |



```
> fun AND (false,x) = false | AND (true,x) = x;
val truthValues = [(false,false),(false,true),(true,false),(true,true)];
map AND truthValues;
val AND = fn: bool * bool -> bool
> val truthValues =
   [(false, false), (false, true), (true, false), (true, true)]:
   (bool * bool) list
> val it = [false, false, false, true]: bool list
```

SML

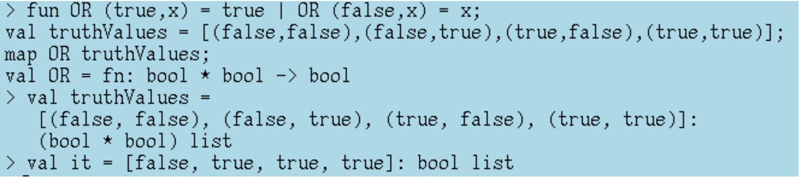
```
fun AND (false,x) = false | AND (true,x) = x;
val truthValues = [(false,false),(false,true),(true,false),(true,true)];
map AND truthValues;
```

# ch1 Proposiational Calculus (Disjunction)

Exploring compound propositions

- Disjunction: $p \vee q$

| $p$ | $q$ | $(p \wedge q)$ |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |



```
> fun OR (true,x) = true | OR (false,x) = x;
val truthValues = [(false,false),(false,true),(true,false),(true,true)];
map OR truthValues;
val OR = fn: bool * bool -> bool
> val truthValues =
   [(false, false), (false, true), (true, false), (true, true)]:
   (bool * bool) list
> val it = [false, true, true, true]: bool list
```

SML

```
fun OR (true,x) = true | OR (false,x) = x;
val truthValues = [(false,false),(false,true),(true,false),(true,true)];
map OR truthValues;
```

# ch1 Proposiational Calculus (Negation)

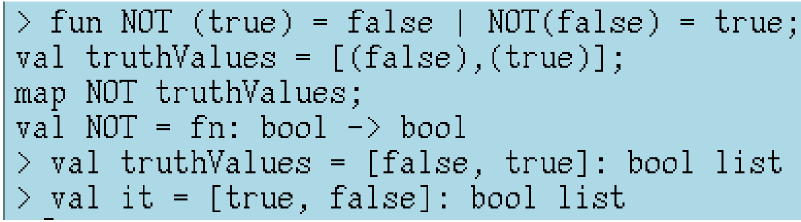
Exploring compound propositions

- Negation: $\sim p$

| $p$ | $\sim p$ |
|---|---|
| F | T |
| T | F |

SML

```
fun NOT (true) = false | NOT(false) = true;
val truthValues = [(false),(true)];
map NOT truthValues;
```



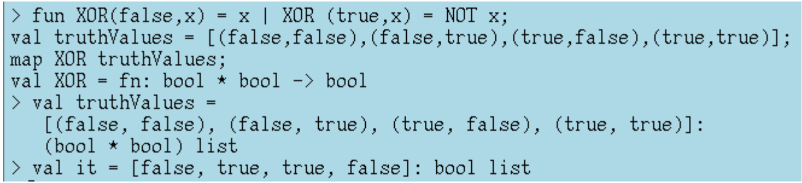
```
> fun NOT (true) = false | NOT(false) = true;
val truthValues = [(false),(true)];
map NOT truthValues;
val NOT = fn: bool -> bool
> val truthValues = [false, true]: bool list
> val it = [true, false]: bool list
```

# ch1 Proposiational Calculus (Exclusive OR)

Exploring compound propositions

- Exclusive OR: $p \oplus q$

| $p$ | $q$ | $(p \oplus q)$ |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | F |



```
> fun XOR(false,x) = x | XOR (true,x) = NOT x;
val truthValues = [(false,false),(false,true),(true,false),(true,true)];
map XOR truthValues;
val XOR = fn: bool * bool -> bool
> val truthValues =
   [(false, false), (false, true), (true, false), (true, true)]:
   (bool * bool) list
> val it = [false, true, true, false]: bool list
```

SML

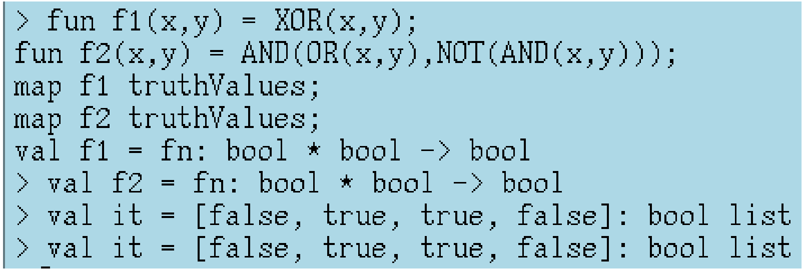
```
fun XOR(false,x) = x | XOR (true,x) = NOT x;
val truthValues = [(false,false),(false,true),(true,false),(true,true)];
map XOR truthValues;
```

# ch1 Proposiational Calculus (Problem 1.18)

Using compound propositions

- $p \oplus q \equiv (p \vee q) \wedge \sim (p \wedge q)$

| $p$ | $q$ | $(p \oplus q)$ | $((p \vee q) \wedge \sim (p \wedge q))$ |
|---|---|---|---|
| F | F | F | F |
| F | T | T | T |
| T | F | T | T |
| T | T | F | F |



```
> fun f1(x,y) = XOR(x,y);
fun f2(x,y) = AND(OR(x,y),NOT(AND(x,y)));
map f1 truthValues;
map f2 truthValues;
val f1 = fn: bool * bool -> bool
> val f2 = fn: bool * bool -> bool
> val it = [false, true, true, false]: bool list
> val it = [false, true, true, false]: bool list
```

– Since the last two colums of the truth table are the same, the two propositions are equivalent

SML

```
fun f1(x,y) = XOR(x,y);
fun f2(x,y) = AND(OR(x,y),NOT(AND(x,y)));
map f1 truthValues;
map f2 truthValues;
```

# ch1 Proposiational Calculus (Problem 1.13 a Part 1)

Using compound propositions

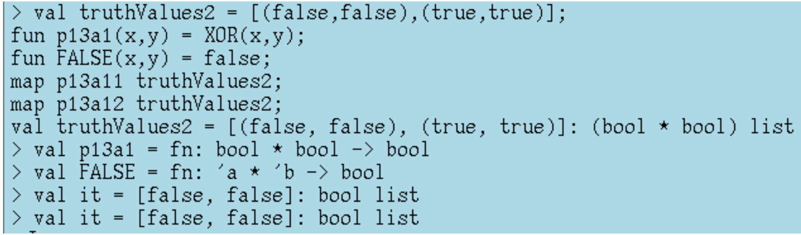
- Simplify $p \oplus p$ and $p \oplus (p \oplus p)$

| $p$ | $(p \oplus p)$ | $F$ |
|---|---|---|
| F | F | F |
| T | F | F |

  - Since the last two columns, "F" and "$p \oplus p$" are identical the propositions are equivalent

SML

```
val truthValues2 = [(false,false),(true,true)];
fun p13a1(x,y) = XOR(x,y);
fun FALSE(x,y) = false;
map p13a11 truthValues2;
map p13a12 truthValues2;
```



```
> val truthValues2 = [(false,false),(true,true)];
fun p13a1(x,y) = XOR(x,y);
fun FALSE(x,y) = false;
map p13a11 truthValues2;
map p13a12 truthValues2;
val truthValues2 = [(false, false), (true, true)]: (bool * bool) list
> val p13a1 = fn: bool * bool -> bool
> val FALSE = fn: 'a * 'b -> bool
> val it = [false, false]: bool list
> val it = [false, false]: bool list
```

# ch1 Proposiational Calculus (Problem 1.13 a Part 2)

Using compound propositions

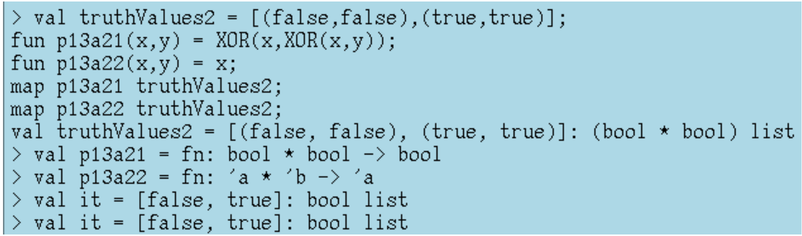
- Simplify $p \oplus p$ and $p \oplus (p \oplus p)$

| $p$ | $p \oplus (p \oplus p)$ |
|---|---|
| F | F |
| T | T |

  - Since the column ”p” and ”$p \oplus (p \oplus p)$ are identical the propositions are equivalent

SML

```
val truthValues2 = [(false,false),(true,true)];
fun p13a21(x,y) = XOR(x,XOR(x,y));
fun p13a22(x,y) = x;
map p13a21 truthValues2;
map p13a22 truthValues2;
```



```
> val truthValues2 = [(false,false),(true,true)];
fun p13a21(x,y) = XOR(x,XOR(x,y));
fun p13a22(x,y) = x;
map p13a21 truthValues2;
map p13a22 truthValues2;
val truthValues2 = [(false, false), (true, true)]: (bool * bool) list
> val p13a21 = fn: bool * bool -> bool
> val p13a22 = fn: 'a * 'b -> 'a
> val it = [false, true]: bool list
> val it = [false, true]: bool list
```

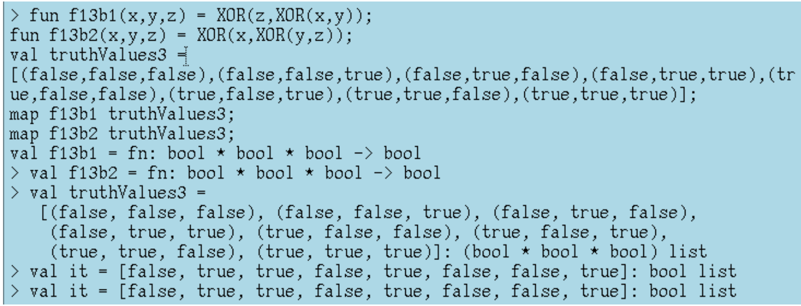
# ch1 Proposiational Calculus (Problem 1.13 b)

Using compound propositions

- Is $(p \oplus q) \oplus r \equiv p \oplus (q \oplus r)$

| $p$ | $q$ | $r$ | $((p \oplus q) \oplus r)$ | $(p \oplus (q \oplus r))$ |
|---|---|---|---|---|
| F | F | F | F | F |
| F | F | T | T | T |
| F | T | F | T | T |
| F | T | T | F | F |
| T | F | F | T | T |
| T | F | T | F | F |
| T | T | F | F | F |
| T | T | T | T | T |

  - Since the last columns are the same then the respective propositions are equivalent



```
> fun f13b1(x,y,z) = XOR(z,XOR(x,y));
fun f13b2(x,y,z) = XOR(x,XOR(y,z));
val truthValues3 =
[(false,false,false),(false,false,true),(false,true,false),(false,true,true),(tr
ue,false,false),(true,false,true),(true,true,false),(true,true,true)];
map f13b1 truthValues3;
map f13b2 truthValues3;
val f13b1 = fn: bool * bool * bool -> bool
> val f13b2 = fn: bool * bool * bool -> bool
> val truthValues3 =
   [(false, false, false), (false, false, true), (false, true, false),
    (false, true, true), (true, false, false), (true, false, true),
    (true, true, false), (true, true, true)]: (bool * bool * bool) list
> val it = [false, true, true, false, true, false, false, true]: bool list
> val it = [false, true, true, false, true, false, false, true]: bool list
```

SML

```
fun f13b1(x,y,z) = XOR(z,XOR(x,y));
fun f13b2(x,y,z) = XOR(x,XOR(y,z));
val truthValues3 = [(false,false,false),(false,false,true),
(false,true,false),(false,true,true),(true,false,false),
(true,false,true),(true,true,false),(true,true,true)];
map f13b1 truthValues3;
map f13b2 truthValues3;
```

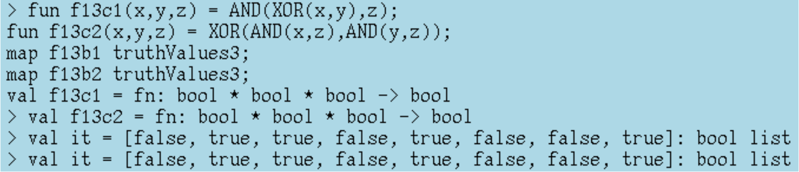
# ch1 Proposiational Calculus (Problem 1.13 c)

Using compound propositions

- Is $(p \oplus q) \wedge r \equiv p(p \wedge r) \oplus (q \wedge r)$

| $p$ | $q$ | $r$ | $((p \oplus q) \wedge r)$ | $((p \wedge r) \oplus (q \wedge r))$ |
|---|---|---|---|---|
| F | F | F | F | F |
| F | F | T | F | F |
| F | T | F | F | F |
| F | T | T | T | T |
| T | F | F | F | F |
| T | F | T | T | T |
| T | T | F | F | F |
| T | T | T | F | F |

  - Since the last columns are the same then the respective propositions are equivalent



```
> fun f13c1(x,y,z) = AND(XOR(x,y),z);
fun f13c2(x,y,z) = XOR(AND(x,z),AND(y,z));
map f13b1 truthValues3;
map f13b2 truthValues3;
val f13c1 = fn: bool * bool * bool -> bool
> val f13c2 = fn: bool * bool * bool -> bool
> val it = [false, true, true, false, true, false, false, true]: bool list
> val it = [false, true, true, false, true, false, false, true]: bool list
```

SML

```
fun f13c1(x,y,z) = AND(XOR(x,y),z);
fun f13c2(x,y,z) = XOR(AND(x,z),AND(y,z));
map f13b1 truthValues3;
map f13b2 truthValues3;
```

# ch1 Proposiational Calculus (Problem 1.17)

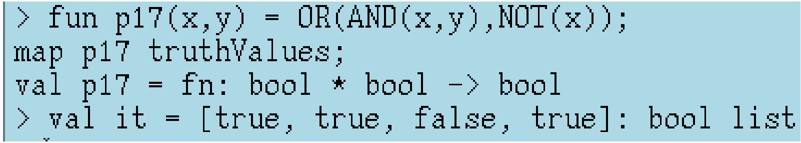
Using compound propositions

- Construct the truth table for $(p \wedge q) \vee (\sim p)$

| $p$ | $q$ | $(p \wedge q)$ | $\sim p$ | $((p \wedge q) \vee \sim p)$ |
|---|---|---|---|---|
| F | F | F | T | T |
| F | T | F | T | T |
| T | F | F | F | F |
| T | T | T | F | T |

SML

```
fun p17(x,y) = OR(AND(x,y),NOT(x));
map p17 truthValues;
```



```
> fun p17(x,y) = OR(AND(x,y),NOT(x));
map p17 truthValues;
val p17 = fn: bool * bool -> bool
> val it = [true, true, false, true]: bool list
```

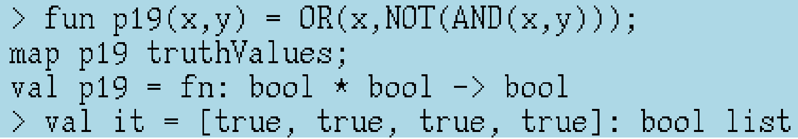
# ch1 Proposiational Calculus (Problem 1.19)

Using compound propositions

- Show that $p \vee \sim (p \wedge q)$ is a tautology

| $p$ | $q$ | $(p \vee \sim (p \wedge q))$ |
|---|---|---|
| F | F | T |
| F | T | T |
| T | F | T |
| T | T | T |

  - Since the last column of the truth table is always "T", then the proposition is a tautology



```
> fun p19(x,y) = OR(x,NOT(AND(x,y)));
map p19 truthValues;
val p19 = fn: bool * bool -> bool
> val it = [true, true, true, true]: bool list
```

SML

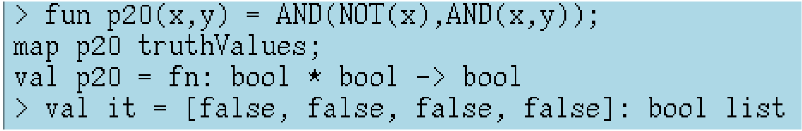
```
fun p19(x,y) = OR(x,NOT(AND(x,y)));
map p19 truthValues;
```

# ch1 Proposiational Calculus (Problem 1.20)

Using compound propositions

- Show that $\sim p \wedge (p \wedge q)$ is a contradiction

| $p$ | $q$ | $(\sim p \wedge (p \wedge q))$ |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | F |



```
> fun p20(x,y) = AND(NOT(x),AND(x,y));
map p20 truthValues;
val p20 = fn: bool * bool -> bool
> val it = [false, false, false, false]: bool list
```

  - Since the last column of the truth table is always ”F”, then the proposition is a contradiction

SML

```
fun p20(x,y) = AND(NOT(x),AND(x,y));
map p20 truthValues;
```