02156 Exercises-01

Jørgen Villadsen

2021

Exercise 1

Use truth tables for the formula $(A \to B) \wedge (B \to A)$ (usually abbreviated $A \leftrightarrow B$) and compare it to the xor-operator $\oplus$ defined on today's slides.

Exercise 2

The present exercise considers the modelling of logic design as shown on today's slides.

A half adder is a logical circuit that performs an addition operation on two binary digits. The half adder produces a sum and a carry value which are both binary digits.

$\text{Sum} \;=\; \text{Bit1} \oplus \text{Bit2}$

$\text{Carry} \;=\; \text{Bit1} \wedge \text{Bit2}$

A full adder is a logical circuit that performs an addition operation on three binary digits (the third is considered a carry value). The full adder produces a sum and carry value, which are both binary digits.

$\text{Sum} \;=\; (\text{Bit1} \oplus \text{Bit2}) \oplus \text{Carryin}$

$\text{Carryout} \;=\; (\text{Bit1} \wedge \text{Bit2}) \vee (\text{Carryin} \wedge (\text{Bit1} \oplus \text{Bit2}))$

Write Prolog programs `halfadder` and `fulladder` such that the following examples work:

```
?- halfadder(Bit1,Bit2,Sum,Carry).

Bit1 = 0
Bit2 = 0
Sum = 0
Carry = 0 ;

Bit1 = 0
Bit2 = 1
Sum = 1
Carry = 0 ;

...

?- fulladder(Bit1,Bit2,Carryin,Sum,Carryout).

...
```

For both programs the definitions above must be used.

Exercise 3

Use truth tables for the formula $((A \to B) \wedge ((\neg A \to C) \wedge (\neg B \to \neg C))) \to B$ and comment on the resulting truth value.

Exercise 4

Write a Prolog program `factorial(+Integer1,?Integer2)` that succeeds if and only if `Integer2` is the factorial of `Integer1` (the factorial of n is $n! = n(n-1)(n-2)\cdots 1$ and $0! = 1$).

Hint: The program `power` on today's slides might be useful.

Exercise 5

Consider the following fragment of a food ingredient database:

```
ingredient(pizza,ham).
ingredient(pizza,sauce).
ingredient(pizza,cheese).
ingredient(ham,meat).
ingredient(ham,salt).
ingredient(cheese,milk).
ingredient(cheese,salt).
ingredient(sauce,tomato).
ingredient(sauce,water).
ingredient(sauce,salt).
```

Hence `pizza` contains the ingredients `ham`, `sauce` and `cheese`. An ingredient may contain other ingredients, for example `ham` contains the ingredients `meat` and `salt`.

Write a Prolog program `component(?Term1,?Term2)` that succeeds if and only if `Term1` is an ingredient in `Term2` either directly or indirectly because it is a component of an ingredient in `Term2`.

Sample Prolog queries:

```
?- component(salt,pizza).

Yes

?- component(jam,pizza).

No

?- component(X,pizza).

X = ham ;

X = sauce ;

X = cheese ;

X = meat ;

X = salt ;

X = tomato ;

X = water ;

X = salt ;

X = milk ;

X = salt ;

No
```

Try to explain the last query in details (as far as possible at this moment). What happens if `pizza` is replaced with a variable `Y` in the queries?

Hint: The program `ancestor` on today's slides might be useful.