Discrete Structures (W1): Boolean Expressions

Kalvis
RBS

You will compile a sample LaTeX document to create math expressions easily and to satisfy the APA v6 style. You will also manipulate Boolean expressions on paper and with the Coq.

Keywords: LaTeX, APA style, Coq, Boolean expressions

Walk-throughs are short exercises that we do together. Their aim is to practice technical skills.

(W1.1) Set up MiKTeX and Coq software.

(W1.2) Create an APA compliant document with some math in LaTeX and create a PDF.

(W1.3) Identify 6 Boolean operations. Build their truth tables.

(W1.4) Translate English into propositional logic.

(W1.5) Use precedence and associativity to draw abstract and concrete syntax trees.

(W1.6) Run CoqIDE, check types in Coq, including atomic types, tuples, function types.

(W1.7) Build truth tables for Boolean expressions.

(W1.8) Rewrite Boolean expressions as Disjunctive Normal Form (DNF).

(W1.9) Prove propositional tautologies with Coq.

(W1.10) Reduce word problems to Boolean satisfiability problems.

(W1.1) MiKTeX and Coq

Note. Lab computers on the 4th floor should have these tools installed. If you wish to run these tools on another Windows OS machine, make sure you have administrator access (for Coq) and follow these steps:

- Download MiKTeX from https://miktex.org/download (file name `basic-miktex-2.9.7269-x64.exe`).
- Visit https://coq.inria.fr/download.
- Click on "Coq 8.10.2 is out" and "8.10.2 release of Coq" links. Download and run `coq-8.10.2-installer-windows-x86_64.exe`.
- Create a shortcut on your desktop to run `c:\Coq\bin\coqide.exe`

For Linux or OS X, please ask instructor.

(W1.2) Create and Compile a LaTeX Document

- Get 2 source files. You can visit https://ctan.org/pkg/apa6 (download and unpack ZIP, locate `shortsample.tex` and `bibliography.bib` in subdirectory "samples").
- Also visit https://bit.ly/2MX7Z09 –this directory will contain LaTeX files for our walk-throughs and homeworks. Then compile:

```
xelatex shortsample
biber shortsample
xelatex shortsample
```

(W1.3) Identify 6 Boolean operators

Draw or review truth tables for the following Boolean operators (see Rosen, 2019, p.4).

- $A \wedge B$ (read: A **and** B).
- $A \vee B$ (read: A **or** B).
- $\neg A$ (read: **not** A).
- $A \rightarrow B$ (**if** A **then** B; A **implies** B).
- $A \leftrightarrow B$ (A **if and only if** B; A **equivalent to** B).
- $A \oplus B$ (**either** A **or** B; A **plus** B **modulo** 2).

(W1.4) Translate English Sentences

Write these sentences as short Boolean expressions, explain the meaning of the letters used.

- Willy gets caught whenever he cheats. (see Rosen, 2019, p.15, Ex24.d).
- My airplane flight is late iff I have to catch a connecting flight. (see Rosen, 2019, p.16, Ex28.e).
- The apple trees will bloom if it stays warm for a week. (see Rosen, 2019, p.15, Ex25.b).
- The trains run late on exactly those days when I take it. (see Rosen, 2019, p.15, Ex27.e).
- $x = 0$ if and only if $x^2 = 0$.
- For a quadrangle to be a parallelogram it is **necessary** that both diagonals intersect in their middle points.
- For a quadrangle to be a parallelogram it is **sufficient** that both diagonals intersect in their middle points.
- You can pay using U.S. dollars or euros. (see Rosen, 2019, p.15, Ex27.e).

(W1.5) Use precedence and associativity

Precedence and associativity defines how to restore omitted parentheses in expressions with more than one operation. Evaluate from Python command-line:

- `2 + 2%3`. Is it `(2 + 2)%3` or `2 + (2%3)`? Which is higher precedence – addition or remainder?
- `4 * 5 * *2`. Is it `(4 * 5) * *2` or `4 + (5 * 2)`? Which is higher precedence – multiplication or raising to a power?
- `31//5//2`. Is it `(31//5)//2` or `31//(5//2)`? Is integer division left or right associative?
- `2 * *3 * *4`. Is it `(2 * *3) * *4` or `2 * *(3 * *4)`? Is raising to a power left or right associative?
- Is addition left or right associative?

(W1.6) Run CoqIDE

Run samples from Bertot (2010, p.2–8).

(W1.7) Build Truth Tables

Example: $(p \vee \neg(q \wedge r)) \rightarrow (\neg\neg r)$.

(a) Draw a truth table for the above Boolean expression. How many combinations can the parameters (p, q, r) have?

(b) Enter this as a Coq proposition using logical connectives:

```
Variable P Q R: Prop.
Check (P \/ (Q /\ ~R)) -> (~~R).
```

(c) Define it as a function from `bool * bool * bool` to `bool` using prefix operations and compute some values:

```
Require Import Bool.
Definition myfun
  (p:bool) (q:bool) (r:bool) : bool :=
  implb (orb p (negb (andb q (negb p))))
    (negb (negb r)).
Eval compute in myfun true true true.
```

(W1.8) Rewrite as a DNF

For the previous Boolean expression write its Disjunctive Normal Form (DNF).

(W1.9) Prove Tautologies with Coq

Note. We suggest that you experiment with many existing proofs, try to modify them slightly. This will help, if you attempt proving new things.

- Visit https://bit.ly/37EMRUq and run the examples. They are identical to the CSE 191 Course Notes (Knepley, 2019, p.12–21).
- Run the examples from the official Coq tutorial (see Nahas, 2012).
- Run the examples from the *Software Foundations* book, see Vol.1 Chapter "Logic in Coq": https://bit.ly/36DtDhy.

(W1.10) Reduce to Satisfiablility

- Write a Boolean expression with 4 variables p_1, p_2, p_3, p_4, which is satisfied if and only if at least 2 of these variables have value `true`.
- Write a Boolean expression with 4 variables, which is satisfied if and only if at least 2 of the variables are `false`.
- How many vertices, edges and faces has the polyhedron shown in Figure 1? Is it regular? How many Boolean variables would you need to write an expression that is satisfied if and only if the polyhedron has a Hamiltonian path?

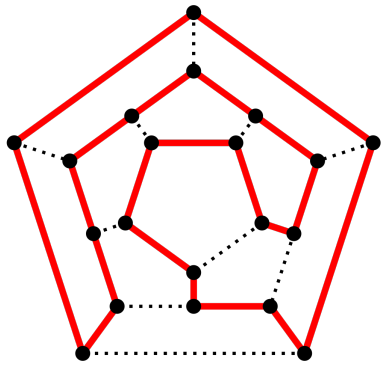

Figure 1: Hamiltonian Path.

References

Bertot, Y. (2010). Coq in a hurry. 3rd cycle. In *Types summer school* (pp. 42–42). inria-00001173v4: Universite de Nice. Retrieved January 5, 2020, from https://cel.archives-ouvertes.fr/inria-00001173v5/document

Knepley, M. G. (2019). *Discrete structures: Lecture notes for cse 191*. University of Buffalo. Retrieved from https://cse.buffalo.edu/~knepley/classes/cse191/ClassNotes.pdf

Nahas, M. (2012). Mike nahas's coq tutorial. Retrieved January 22, 2019, from https://coq.inria.fr/tutorial-nahas

Rosen, K. H. (2019). *Discrete mathematics and its applications* (8th ed.). 2 Penn Plaza, New York: McGraw-Hill Education.