Programming and Modelling
Weekly Practice Exercises

Week 4: Automated Reasoning about Programs

Work on your solutions in groups of 2-3 students. In the next programming café the TAs will go over the questions and your answers together as a class.

Exercises

(1) This week you will continue reading through the "Course Notes" online pages of the Logika website. The online course notes are available at:
`https://logika.v3.sireum.org/dschmidt/index.html`
Remark: There might seem like a lot of sections, but many of them are very short, just a few paragraphs.

- Section 3.4. "Logika Proof Syntax"
- Section 3.4. "Logika Proof Syntax"
- Section 3.5. "The premise Justification"
- Section 3.6. "The assume Justification"
- Section 3.7. "Logical Operator Based Justifications"
- Section 3.8. "And-Introduction and And-Elimination"
- Section 3.9. "Or-Introduction and Or-Elimination"
- Section 3.10. "Implies-Introduction and Implies-Elimination"
- Section 3.11. "Negation"
- Section 3.12. "Negation Introduction"

(2) In the lecture we discussed the automated theorem proof checker tool Sireum Logika.
  (a) Install the Integrated Verification Environment (IVE) of Sireum Logika (version 3). Follow the instructions in the lecture slides. The download page is:
  `https://logika.v3.sireum.org/doc/01-getting-started/index.html`
  (b) Following the instructions in the lecture slides, create a new Logika project and file, and verify the proof given in the lecture slides for the sequent:
  $\vdash \mathbf{a} \rightarrow \mathbf{a}$

(3) Consider the following Logika proof:

```
s, q, r ⊢ r ∧ (q ∧ s)
{
  1. p                    assume
  2. q                    premise
  3. r                    premise
  4. q ∨ p                ∨i 2 1
  5. r ∧ (q ∧ p)          ∧i 3 4
}
```

When I type this proof into Sireum Logika, I get a number of errors.

(a) Create a new Logika file and type in this proof to reproduce the errors.
(b) Create your own correct proof using graphical notation.
(c) Translate your graphical proof into Logika's textual notation, and verify your proof in Logika.

(4) In the lectures we went through a number of examples for translating graphical proofs into Logika textual proofs.
(a) Verify the *"Idempotence of conjunction 1"* textual Logika proof from the lecture slides using Logika.
(b) Translate the following graphical proof of *"Idempotence of disjunction 1"* into the textual Logika notation, and verify it using Logika:

$$\dfrac{\dfrac{\overset{1}{[\mathbf{A}]}}{\mathbf{A} \vee \mathbf{A}}\ \vee\mathbf{I_1}}{\mathbf{A} \rightarrow \mathbf{A} \vee \mathbf{A}}\ \rightarrow\mathbf{I}.1$$

(5) Prove the following sequents, and then translate your proofs into textual Logika notation and verify them using Logika:
(a) $\mathbf{A \wedge B \vdash A \vee B}$
(b) $\mathbf{P \vee P \vdash P}$
Hint: Try applying the or-elimination rule. Start with what you might assume in each of the branches of the disjunction. The or-elimination rule discharges these assumptions

(6) In the lecture we discussed proof rules involving *negation*. Using the graphical notation from lectures, prove the following sequents:
(a) $\mathbf{P \rightarrow Q, \neg Q \vdash \neg P}$
(b) $\mathbf{\neg Q \vdash Q \rightarrow S}$

(7) Translate each of your proofs from exercise (6) above into Logika textual notation, and use Logika to verify your proofs.

(8) *[Optional challenge]* Below are some additional sequents for you to practice creating proofs.
(a) $\mathbf{\vdash ((A \wedge B) \wedge C) \rightarrow (A \wedge (B \wedge C))}$
(b) $\mathbf{P \rightarrow Q, Q \rightarrow R \vdash P \rightarrow R}$
(c) $\mathbf{\vdash (P \rightarrow Q) \rightarrow (Q \rightarrow R) \rightarrow (P \rightarrow R)}$
(d) $\mathbf{(A \vee B), C \vdash (A \wedge C) \vee (B \wedge C)}$
Hint: Look carefully at the or-introduction rule: if you have proved some expression **E1** then you can extend this to $\mathbf{E1 \vee E2}$, for any arbitrary expression **E2** that could be useful in your proof.