In preparation for all labs, make sure that you have read and understood the Rodin User’s Handbook (currently covers Rodin v2.7). Pay careful attention to sections 2.9 and 3.4 on Proving.

Contents

1 The Nand Gate problem specification ........................................ 2
2 Hand Proof of thm1 .................................................................. 2
3 Hints for Rodin ........................................................................ 4
   3.1 Manual Auto/Post Tactic ...................................................... 4
   3.2 Lemmas .............................................................................. 4
   3.3 The theorem ....................................................................... 5

List of Figures

1 Proof of lemma1 ..................................................................... 6
2 Using lemma1 to re-write the goal .......................................... 8
3 MP on hypothesis .................................................................... 9
4 Transformation to the equational form ................................. 10

List of Tables
1 The Nand Gate problem specification

**CONTEXT**
C0

**CONSTANTS**

- $D$
- $\text{and}$
- $\text{inv}$  -- inverter
- $\text{nand}$

**AXIOMS**

- **axm1**: $D = \{0, 1\}$
- **axm2**: $\text{and} \in D \times D \rightarrow D$
- **axm3**: $\text{inv} \in D \rightarrow D$
- **axm4**: $(\forall x, y \cdot x \in D \land y \in D \Rightarrow (\text{and}(x \leftrightarrow y) = 1 \iff x = 1 \land y = 1))$
  definition of AND gate
- **axm5**: $(\forall x \cdot x \in D \Rightarrow ((x = 1 \iff \text{inv}(x) = 0) \land (x = 0 \iff \text{inv}(x) = 1)))$
  definition of inverter
- **axm6**: $\text{nand} \in D \times D \rightarrow D$
- **axm7**: $(\forall x, y \cdot x \leftrightarrow y \in D \times D \Rightarrow (\text{nand}(x \leftrightarrow y) = 0 \iff x = 1 \land y = 1))$
  Specification of NAND gate
- **thm1**: $(\forall x, y \cdot x \in D \land y \in D \Rightarrow (\text{inv}(\text{and}(x \leftrightarrow y)) = \text{nand}(x \leftrightarrow y)))$
  Can we implement the NAND gate specification by inverting an AND gate?

**END**

2 Hand Proof of thm1

The Rodin automated theorem prover does not discharge thm1. So we will have to guide the theorem prover. It is always a good idea to do your own proof first on paper. Then you can translate your proof into terms that the theorem prover will understand.

We can easily prove the following theorems about variables and expressions of the BIT type D:
Theorems for variables $d_1, d_2$: $D$ where $D = \{0, 1\}$:

\[ d_1 = d_2 \equiv d_1 = 1 \equiv d_2 = 1 \quad (1) \]

\[ \neg(d_1 = 0) \equiv d_1 = 1 \quad (2a) \]
\[ d_1 = 0 \equiv \neg(d_1 = 1) \quad (2b) \]
\[ d_1 = 0 \not\equiv d_1 = 1 \quad (2c) \]
\[ \neg(d_1 = 0) \equiv d_1 = 1 \quad (2d) \]

A theorem for $\equiv$ is:

\[ p \equiv q \equiv \neg p \equiv \neg q \quad (3) \]

**Proof strategy:** First prove the RHS of (1) with $d_1 := \text{inv}(\text{and}(x, y))$ and $d_2 := \text{nand}(x, y)$. Then use (1) to obtain the LHS, which is what must be proved.

**Prove:** $\text{inv}(\text{and}(x, y)) = 1 \equiv \text{nand}(x, y) = 1$:

\[ \text{inv}(\text{and}(x, y)) = 1 \]
\[ = \langle \text{axm5 with } x := \text{and}(x, y) \rangle \]
\[ \text{and}(x, y) = 0 \]
\[ = \langle (2b) \text{ with } d_1 := \text{and}(x, y) \rangle \]
\[ \neg(\text{and}(x, y) = 1) \]
\[ = \langle \text{axm4} \rangle \]
\[ \neg(x = 1 \land y = 1) \]
\[ = \langle \text{axm7} \rangle \]
\[ \neg(\text{nand}(x, y) = 0) \]
\[ = \langle (2a) \text{ with } d_1 := \text{and}(x, y) \rangle \]
\[ \text{nand}(x, y) = 1 \]

\[ \blacksquare \]

Note that $\equiv$ in equational logic is the same as $\leftrightarrow$ in Rodin. However the operator $\equiv$ is associative. Although Rodin does allow you to to use Leibniz to replace equivalent terms, it does not allow you to replace equivalent
predicates as we do in equational logic. So that you have to first do ⇒ and then ⇐.

3 Hints for Rodin

3.1 Manual Auto/Post Tactic

In the Rodin settings, create your own Auto/Post-tactic as shown below (see manual).

The only automation is thus minimal: true goal, false hypothesis and Goal in Hypotheses. As you become more familiar with manually guiding the prover you can add additional automation.

3.2 Lemmas

When submitting \textit{thm1}, the automatic Rodin prover (in version 2.7 at least) does not discharge it. You can try guiding the proof using what you know from what we have done in class and the labs, and sections 3.9 and 3.4 in the Rodin manual. But what happens if you still cannot get the proof?

One way to proceed is to do the proof by hand in any logic you know. In the previous section we used equational logic to do the proof of \textit{thm1}. We can now try to guide Rodin into following our hand proof.

A key theorem about expressions of the BIT type $D$ that we used in the
hand proof was theorem (1): \((d_1 = d_2) \equiv (d_1 = 1 \equiv d_2 = 1)\). We can add this as a **theorem** (say lemma1) in the Rodin context to prove from the axioms.

We have also added lemmas **lemma2a** (2a) and **lemma2b** (2b). We must now prove that these lemmas are theorems that follow from the axioms. **lemma2a** and **lemma2b** prove automatically. **lemma1** needs more guidance as shown in Fig. 1, where we must rewrite equivalence in the goal with \(\Rightarrow\) and then \(\Leftarrow\).

You might want to do the \(\forall\) goal, \(\Rightarrow\) goal (deduction theorem), and \(\text{eh}\) (Leibniz substitution) manually for theorem proving practice (see Fig. 1). Make sure you understand the sequent calculus proof rules you are using.

### 3.3 The theorem

We can now prove theorem **thm1**.

The goal is shown below:

Click on the red \(\forall\) and invoke **forall instantiation** followed by **Deduction** on \(\Rightarrow\) in the goal.
In `lemma1`, rewrite equivalence in the goal with $\Rightarrow$ and then $\Leftarrow$.

Figure 1: Proof of `lemma1`
We can now instantiate the universally qualified variables in \textit{lemma1} in the hypotheses:

\[
\begin{array}{c}
\exists \alpha \quad d_1 := \text{inv}(\text{and}(x, y)) \\
\end{array}
\]

As mentioned in the manual proof do the instantiations \( d_1 := \text{inv}(\text{and}(x, y)) \) and \( d_2 := \text{nand}(x, y) \). You enter these in the little yellow box and then invoke the red \( \forall \).

The result is shown in Fig. 1. The typing well-definedness proof obligations can be blown away with prover ML (eliminate \( \land \) in the goal). We can use modus ponens and then do an equivalence re-write in the hypothesis as follows:

The result of the equivalence re-write in the hypothesis is shown in Fig. 2. We then invoke MP on the hypothesis as shown in the figure to get two proof obligations as shown in the MP-HYP sequent rule below:

\[
\text{MP-HYP} \\
\frac{H \vdash A \\ H, A, B \vdash G}{H, A \Rightarrow B \vdash G}
\]

Note that in our case \( B \) in the consequent (of the hypothesis) is just the goal \( G \). Thus the rule reduces to:

\[
\text{MP-HYP} \\
\frac{H \vdash A \\ H, A, G \vdash G}{H, A \Rightarrow G \vdash G}
\]

The right proof obligation is trivially true so that it is now sufficient to prove \( A \) from the hypotheses which is just: \( \text{inv}(\text{and}(x, y)) = 1 \equiv \text{nand}(x, y) = 1 \). See Fig. 2 for the result of MP-HYP.

The rest should be relatively easy for you to do on your own.
In `lemma1`, rewrite equivalence in the goal with $\Rightarrow$ and then $\Leftarrow$.

**Figure 2:** Using `lemma1` to re-write the goal.
To prove the goal $G$ it is sufficient to prove the antecedent $A$ in the hypothesis.

$$A \triangleq inv(\text{and}(x, y)) = 1 \equiv \text{nand}(x, y) = 1$$

Figure 3: MP on hypothesis
Goal is now in a form that can be used with the axioms.

Figure 4: Transformation to the equational form