# Assignment 4 – COMP 527: Computation and Logic

Winter 2016
Due 24 March, 2016

## 1 Induction principles for binary trees (35 pts)

We can define binary trees containing numbers as follows:

- $\mathsf{Empty}$ is a binary tree
- Given a natural number $N$ and a binary tree $T_1$ and a binary tree $T_2$, $\mathsf{Node}(N,\ T_1,\ T_2)$ is a binary tree.

$$\frac{}{\mathsf{Empty} : \mathsf{tree}} \qquad \frac{N : \mathsf{nat} \qquad T_1 : \mathsf{tree} \qquad t_2 : \mathsf{tree}}{\mathsf{Node}(N,\ T_1,\ T_2) : \mathsf{tree}}$$

1. (6 pts) Define the induction rule we would need to add to our logic to prove properties about binary trees.
2. (6 pts)Give a proof term (primitive recursion) for the defined induction principle and annotate the induction principle with it arriving at a typing rule for primitive recursion over trees.
3. (8 pts) Define reduction rules for primitive recursion for binary trees.
4. (10 points) Prove that your reduction rules for primitive recursion satisfy subject reduction, i.e. typing is preserved.
5. (5 pts) Give a program using primitive recursion which computes the size of a binary tree using the proof term you introduced in 2.

## 2 Sequent calculus

### 2.1 Practice (10 points)

Give sequent calculus derivations for the following propositions:

- $(A \wedge (B \vee C) \Longrightarrow (A \wedge B) \vee (A \wedge C)$
- $(D \supset B) \supset D \supset C,\ B \supset A,\ (C \supset D) \supset B \Longrightarrow A$

## 2.2 Optimization (25 points)

In class and in the notes, we have given the following sequent calculus rule for initial sequents:

$$\overline{\Gamma, A \Rightarrow A} \; init$$

This rule is can be used for any proposition $A$. For example, we may use it to conclude $\Gamma, A \wedge B \Rightarrow A \wedge B$. This is not very convenient, if we want to implement a program which does automatically search for a proof, since at every step in the proof, we would need to check if we can prove $A$ from our assumptions in $\Gamma$.

It would be more convenient, if we would know that it suffices to apply this rule once the proposition is atomic, i.e. it does not contain any conjunctions, disjunctions, implications etc. We will refer to atomic propositions by writing $P$. Therefore, we would like to replace the previous init rule by the following:

$$\overline{\Gamma, P \Rightarrow P} \; init'$$

This is clearly sound, since we just restricted the use of the original init rule. To show completeness, we must argue that we do not loose anything. The crucial lemma is therefore following:

**Lemma** For all propositions $A$, $\Gamma, A \Rightarrow A$.

Hint: You may need to use the weakening lemma.

## 2.3 Proving Cut (30 points)

The cut-elimination theorem can be stated as follows:

$$\text{If } \mathcal{D} : \Gamma \Longrightarrow A \text{ and } \mathcal{E} : \Gamma, A \Longrightarrow C \text{ then } \mathcal{F} : \Gamma \Longrightarrow C.$$

In class we proved cut-elimination and the notes show several additional cases in detail. Show the following cases for disjunctions.

- $A \vee B$ is the principal formula of the final inference in both $\mathcal{D}$ and $\mathcal{E}$.
- $A \vee B$ is not the principal formula of the last inference in $\mathcal{D}$.
- $A \vee B$ is not the principal formula of the last inference in $\mathcal{E}$.

### 2.4 Optional Question

Consider a system of normal deduction where the elimination rules for disjunction are allowed to end in an extraction.

$$\frac{\Gamma^{\downarrow} \vdash A \vee B \downarrow \quad \Gamma^{\downarrow}, u{:}A \downarrow \ \vdash C \downarrow \quad \Gamma^{\downarrow}, w{:}B \downarrow \ \vdash C \downarrow}{\Gamma^{\downarrow} \vdash C \downarrow}$$

Discuss the relative merits of allowing or disallowing such a rule and show how they impact subsequent developments, in particular bi-directional type checking and the relationship to the sequent calculus.