Logics Exercise		
	TU München Institut für Informatik	Prof. Tobias Nipkow Lars Hupel
SS 2018	Exercise Sheet 3	24.04.2018

Submission of homework: Wednesday 02.05.2018, before noon; either via email or on paper in the TA's office (MI 00.09.063). Until further notice, homework has to be submitted in groups of two students.

Exercise 3.1. [**System G1c**]
An alternative definition of the sequent calculus ("G1c") is defined as follows:

Axioms

Ax $A \Rightarrow A$ L$\bot$ $\bot \Rightarrow$

Rules for weakening (W) and contraction (C)

$$\text{LW}\ \frac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta} \qquad \text{RW}\ \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}$$

$$\text{LC}\ \frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta} \qquad \text{RC}\ \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}$$

Rules for the logical operators

$$\text{L}\wedge\ \frac{A_i, \Gamma \Rightarrow \Delta}{A_0 \wedge A_1, \Gamma \Rightarrow \Delta}\ (i = 0, 1) \qquad \text{R}\wedge\ \frac{\Gamma \Rightarrow \Delta, A \qquad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}$$

$$\text{L}\vee\ \frac{A, \Gamma \Rightarrow \Delta \qquad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta} \qquad \text{R}\vee\ \frac{\Gamma \Rightarrow \Delta, A_i}{\Gamma \Rightarrow \Delta, A_0 \vee A_1}\ (i = 0, 1)$$

$$\text{L}\rightarrow\ \frac{\Gamma \Rightarrow \Delta, A \qquad B, \Gamma \Rightarrow \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta} \qquad \text{R}\rightarrow\ \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B}$$

Notably, weaking and contraction are built-in rules. Show that sequent calculus can be simulated by G1c, i.e., $\vdash_G \Gamma \Rightarrow \Delta$ implies $\vdash_{G1c} \Gamma \Rightarrow \Delta$.

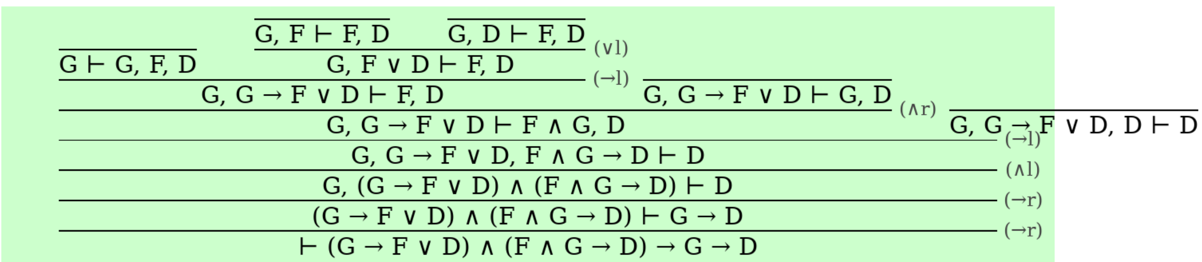
Solution:
We consider two rules, $\wedge L$ and $\neg R$. We show how those can be simulated in G1c.

$$\text{LC}\ \frac{L\wedge\ \dfrac{L\wedge\ \dfrac{\mathbf{F}, \mathbf{G}, \mathbf{\Gamma} \Rightarrow \mathbf{\Delta}}{F, F \wedge G, \Gamma \Rightarrow \Delta}}{F \wedge G, F \wedge G, \Gamma \Rightarrow \Delta}}{\mathbf{F} \wedge \mathbf{G}, \mathbf{\Gamma} \Rightarrow \mathbf{\Delta}} \qquad \frac{R\rightarrow\ \dfrac{\text{RW}\ \dfrac{\mathbf{F}, \mathbf{\Gamma} \Rightarrow \mathbf{\Delta}}{F, \Gamma \Rightarrow \bot, \Delta}}{\Gamma \Rightarrow F \rightarrow \bot, \Delta}}{\mathbf{\Gamma} \Rightarrow \neg \mathbf{F}, \mathbf{\Delta}}$$

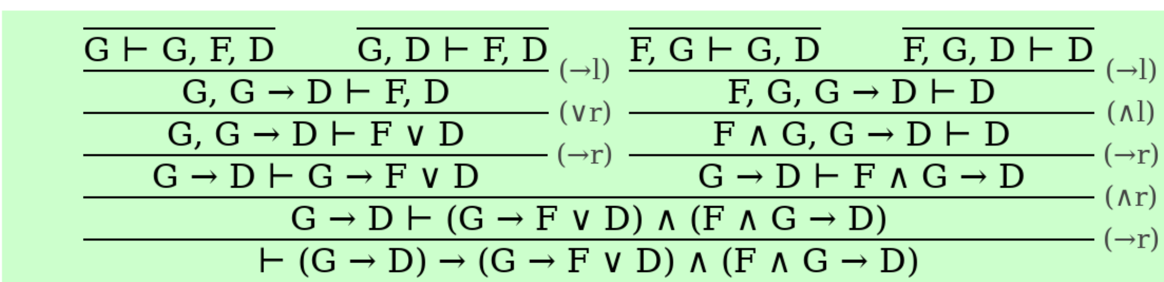
Exercise 3.2. [Cut Elimination, Semantically]
Semantically prove the admissibility of the following rule:

$$\text{If } \vdash_G \Gamma \Rightarrow F, \Delta \text{ and } \vdash_G F, \Gamma \Rightarrow \Delta \text{ then } \vdash_G \Gamma \Rightarrow \Delta$$

Solution:
To prove this semantically, we have to show that given $|\Gamma \Rightarrow F, \Delta|$ and $|F, \Gamma \Rightarrow \Delta|$, $|\Gamma \Rightarrow \Delta|$ holds. In this case, an even stronger property holds: precedent and antecedent are equivalent. That is, $(G \to F \lor D) \land (F \land G \to D) \equiv G \to D$. We can prove this with sequent calculus:[1]



The other direction is similar:[2]



[1] http://logitext.mit.edu/proving/+.28G+.2D.3E+F+.5C.2F+D.29+.2F.5C+.28F+.2F.5C+G+.2D.3E+D.29+.2D.3E+.28G+.E2.86.92+D.29

[2] http://logitext.mit.edu/proving/.28G+.2D.3E+D.29+.2D.3E+.28.28G+.2D.3E+F+.5C.2F+D.29+.2F.5C+.28F+.2F.5C+G+.2D.3E+D.29.29

Exercise 3.3. [More Connectives]
Define sequent rules for the logical connectives "nand" ($\barwedge$) and "xor" ($\otimes$).

Solution:
The simplest way to derive the sequent rules is to consider the definition of $\barwedge$ and $\otimes$.

$$F \barwedge G \equiv \neg(F \wedge G)$$
$$F \otimes G \equiv (F \wedge \neg G) \vee (\neg F \wedge G)$$

One can apply sequent calculus rules on the right-hand sides and simplify accordingly.

$$\barwedge L\ \frac{\Gamma \Rightarrow \Delta, F \qquad \Gamma \Rightarrow \Delta, G}{\Gamma, F \barwedge G \Rightarrow \Delta} \qquad \barwedge R\ \frac{\Gamma, F, G \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, F \barwedge G}$$

$$\otimes L\ \frac{\Gamma, F \Rightarrow \Delta, G \qquad \Gamma, G \Rightarrow \Delta, F}{\Gamma, F \otimes G \Rightarrow \Delta} \qquad \otimes R\ \frac{\Gamma \Rightarrow \Delta, F, G \qquad \Gamma, F, G \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, F \otimes G}$$

Exercise 3.4. [Intermediate Formulas]
Let F, G be formulas such that $F \models G$. Prove that there is an *intermediate formula* H such that the following three conditions hold:

1. H contains only atomic formulas that occur in both F and G
2. $F \models H$
3. $H \models G$

How can H be constructed?

Solution:
This theorem is called "Craig's interpolation theorem". We call H the *interpolant.*

The proof proceeds by induction on the number of elements n in $\text{atoms}(F) \setminus \text{atoms}(G)$.

- Base case: $n = 0$.

  Hence, $|\text{atoms}(F) \setminus \text{atoms}(G)| = 0$. Hence, $\text{atoms}(F) \subseteq \text{atoms}(F) \cap \text{atoms}(G)$. F is a suitable interpolant.

- Inductive step: $n \rightsquigarrow n + 1$.

  There is at least an atomic formula A such that $A \in \text{atoms}(F)$ but $A \notin \text{atoms}(G)$. We define a new formula F' that is the disjunction of F where A is replaced with $\top$ and F where A is replaced with $\bot$:

  $$F' = F[\top/A] \vee F[\bot/A]$$

  Intuitively, F' is a "case distinction" on A. Observe that $A \notin \text{atoms}(F')$. Also, $|\text{atoms}(F') \setminus \text{atoms}(G)| = n$.

  Use the induction hypothesis to obtain an interpolant H for F' and G with $F' \models H$ and $H \models G$.

  We need to show that $F \models H$. This is trivial because $F \models F'$.

Homework 3.1. **[Sequent Calculus]** (2 points)
Prove the formula $((A \to \bot) \to A) \to A$ in System G1c.

Homework 3.2. **[Inversion Rules]** (6 points)
Show that the following inversion rules are admissible:

$$\frac{F \wedge G, \Gamma \Rightarrow \Delta}{F, G, \Gamma \Rightarrow \Delta} \qquad \frac{\Gamma \Rightarrow F \to G, \Delta}{F, \Gamma \Rightarrow G, \Delta}$$

Homework 3.3. **[Sequent Prover]** (12 points)
Implement a sequent calculus prover in a high-level programming language, and test it for examples from this exercise sheet, the lecture, or your own.

Submission: Source code for prover and tests, `README` file containing instructions for how to build the prover and reproduce the tests; by email to `hupel@in.tum.de`. Allowed languages are: Haskell, OCaml, Java, Scala, Rust, Prolog, C++, Python. Only the standard library (i.e. no additional packages) may be used.