10 From Gentzen Systems to Refinement Logic

10.1 A review of the Rules of Gentzen Systems

— see handwritten notes and the rules given in lecture 9 —

It is important to understand that the justification for these rules is based on the idea of provability. Technically, Gentzen Systems are isomorphic to block tableau – there is only a change in notation.

10.2 Some oddities of Multi-Conclusioned Gentzen Systems

— see handwritten notes —

The fact that we allow multiple goals in a sequent leads to two kinds of oddities. The first shows up in a sequent proof of Pierce's Law $((P\supset Q)\supset P)\supset P$

```
     ⊢ ((P⊃Q)⊃Q)⊃P        by ⊃R
     (P⊃Q)⊃P ⊢ P          by ⊃L
[1]    ⊢ P,  P⊃Q          by ⊃L
       P ⊢ P, Q           by axiom
[2]    P ⊢ P              by axiom
```

In this case we used the ⊃R rule in an attempt to prove $P\supset Q$ but in the subgoal we used the new assumption P not to prove Q but to prove the other goal P that we already had. For some people that appears like cheating, like a bait-and-switch strategy, or at least weird. It seems more natural to have proofs that focus on one particular conclusion instead of changing the proof goal in the middle of a proof. So restricting the right hand side to one conclusion appears to be appropriate.

The second oddity comes up in the proof of the law of contraposition

```
     ⊢ (P⊃Q) ⊃ (~Q⊃~P)              by ⊃R
      P⊃Q ⊢ ~Q⊃~P                   by ⊃R
      P⊃Q, ~Q ⊢ ~P                  by ~R
      P⊃Q, ~Q, P ⊢                  by ⊃L
[1]     ~Q, P ⊢ P                   by axiom
[2]     Q, ~Q, P ⊢                  by ~L
        Q, P ⊢ Q                    by axiom
```

Although this proof is perfectly ok, there are two subgoals with no conclusion at all. What is the meaning of such a goal, considering the fact that we understand a sequent to mean "prove one of the conclusions"? Given that a set of conclusions corresponds to a disjunction of these conclusions, an empty set means that we have to prove false, i.e. that the hypotheses are contradictory.

A solution for that is to introduce a constant `f` to represent falsehood and to consider negation $\sim X$ as an abbreviation for $X\supset$`f`. The rules for negation then become superfluous, but we have to add a rule for `f`:

$H, \mathtt{f} \vdash G$

It is easy to see that this rule, together with the rules for implication covers the two rules for negation.

10.3 Introduction to Refinement logic

The logic that we intend to use in the rest of this course, *refinement logic*, results from restricting Gentzen systems to single-conclusioned sequents, dropping negation, and adding the constant `f`. The resulting calculus is simpler and more focused, but it will limit our flexibility when it comes to actually proving things (no more bait-and-switch).

As we will show in the next lecture, the rules of refinement logic can be derived from those of Gentzen systems by dropping the extra conclusions from each sequent. The $\vee$R rule – the only one that explicitly generates two conclusions – needs to be replaced by two rules, each generating only one. However, with these modifications alone, the system becomes incomplete – a goal like $P \vee \sim P$ becomes unprovable – and we must add one extra rule to fix that.