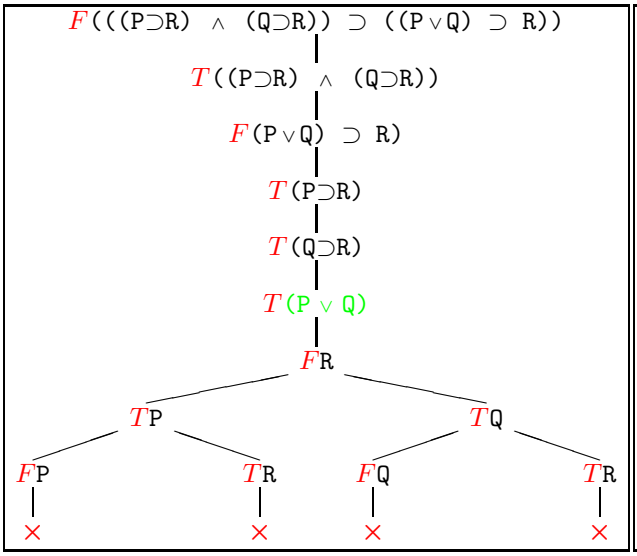
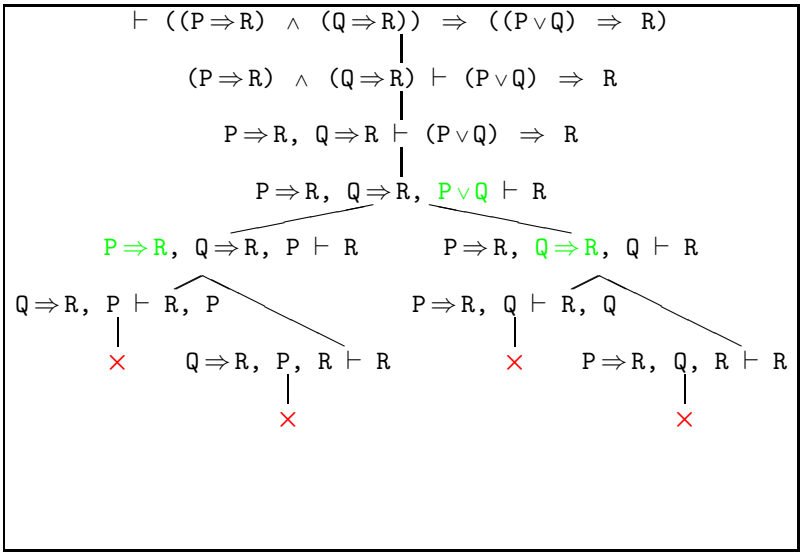
9.1 Gentzen Systems

Although developed independently (and actually a few years earlier), *Gentzen Sequents* can be considered a step towards a more intuitive proof method. They are technically similar to tableau systems but look at differently at the nature of proofs. While the tableau method tries to *refute* a formula X by searching for a *counterexample* for its validity, concluding from a failed search (i.e. a closed tableaux) that the formula is valid, Gentzen systems aim at *constructing a proof for* X or finding *evidence* for its truth. Gentzen systems therefore consider the formula X as *proof goal* and denote this by $\vdash X$.

Let me illustrate the difference and similarities by a simple example





In the proof on the left we have used the known tableau rules to generate consequences of the signed formula at the root of the proof tree. Decomposing $T(\mathtt{P} \vee \mathtt{Q})$, $T(\mathtt{P} \supset \mathtt{R})$, and $T(\mathtt{P} \supset \mathtt{R})$ causes the refutation tree to branch and eventually all branches are closed by a pair of conjugates (e.g. $T\mathtt{P}$ and $F\mathtt{P}$).

In the Gentzen proof on the right we start with the formula that we want to prove and state the fact that we still need to prove it by writing it right of the $\vdash$ symbol We then use proof rules that more or less follow directly from an intuitive understanding of the logical connectives.

To prove $\vdash A \Rightarrow B$ we assume that A is true and try to prove B. We write this as $A \vdash B$, indicating that A is now an assumption under which we have to prove B. Thus in a sequent we write our (current) assumptions left of the $\vdash$ symbol and the conclusions that we want to prove from these assumptions on the right of the $\vdash$.

The proof rule that we used here is called $\Rightarrow \mathtt{R}$ because it decomposes an implication on the right side of the $\vdash$. There are also rules that operate on the assumptions. For instance, we may want to

decompose an assumption $A \wedge B$ by splitting it into its two parts and using A and B as separate assumptions from now on. The rule for that is called $\wedge$L, because it decomposes a conjunction on the left side of the $\vdash$.

Some rules, like the decomposition rule $\vee$L for disjunctions in the assumptions, cause a proof to branch. If we want to prove a goal G from the assumption $A \vee B$, then it is sufficient to show how to prove G from the assumption A and how to prove G from B. Thus $\vee$L decomposes the sequent $A \vee B \vdash G$ into two sequents $A \vdash G$ and $B \vdash G$ which now become the new proof obligations.

Notice that proof rules generate *sufficient* conditions for proving the validity of a sequent. If we knew which of the two formulas A and B in the assumption $A \vee B$ was true we wouldn't have to prove both $A \vdash G$ and $B \vdash G$. But once we have shown $A \vdash G$ and $B \vdash G$ we need only know that A or B is true, because in either case we can prove G.

However, the formulation of the proof rules tries to make sure that applying them doesn't create irreversible effects. This can best be seen in the rule $\Rightarrow$L, which decomposes an implication in the assumptions. The only way to make use of the formula $A \Rightarrow B$ in the sequent $A \Rightarrow B \vdash G$ is to prove A and then use B to prove the goal G. That is, $\Rightarrow$L generates the two proof obligations $\vdash A$ and $B \vdash G$. The rationale is that if we have a proof for A, know that A implies B, and have a proof for G from B, then we can surely conclude G by putting all the evidence together. We will discuss later how to compose such evidence in a more formal setting which will tell us exactly how to build evidence for the validity of the formula that we wanted to prove.

Unfortunately, the proof obligation $\vdash A$ may cause the proof to fail if the validity of G does not depend on $A \Rightarrow B$ but follows from the other assumptions in the proof. While all the other proof rules for assumptions only decompose an assumption without modifying the proof goal itself, $\Rightarrow$L drops the original proof goal and generates a new one, which may not be provable even if the intial formula is valid. To resolve this issue, the original proof goal is preserved as an alternative proof goal and the rule generates the sequent $\vdash G, A$.[1]

As a consequence, sequents may not only have several assumptions, or hypotheses, but also multiple goals, or conclusions. Formally, sequents are defined as objects of the form $H \vdash G$ where both H and G are *sets of formulas* (to make sure the order of formulas is irrelevant). We read such a sequent as "under the assumption that all the formulas in H are true we can show that one of the formulas in G is true" and Gentzen's proof rules show how to do that. Proof rules decompose a formula in the assumptions or in the conclusions until we finally arrive at a sequent that is trivially true, because one of the conclusions is identical to one of the assumptions. Such a sequent is proven by the `axiom` rule $A \vdash A$, which closes the current branch of the proof tree.

Given the above explanations, the tableau proof and the Gentzen proof appear to be entirely different. Technically, however, there are quite a few similarities. We used the same order of proof rules: F-implication ($\Rightarrow$R), then T-conjunction ($\wedge$L), F-implication ($\Rightarrow$R), T-disjunction ($\vee$L), and finally T-implication ($\Rightarrow$L) in each branch, which then is a closed branch in the tableau (closed by the `axiom` rule). We also observe that each sequent contains exactly those formulas of the tableau that have not yet been decomposed ... the once labelled F on the right and the ones labelled T on the left of the $\vdash$.

[1] Obviously one does not have to do that, because we could always go back in our proof tree to the point where $\Rightarrow$L was applied, undo it and try a different proof. This, however, makes the search for a proof more difficult.

We will explore this similarity in greater detail during the next lecture. Today I would like to focus on the intuition behind the Getnzen system by explaining the evidence constructed by the proof rules. The following table summarizes all the rules in their formulation as top down inference rules for propositional logic.

	left	right	
$\wedge$L	$H, A \wedge B \vdash G$ $H, A, B \vdash G$	$H \vdash G, A \wedge B$ $H \vdash G, A$ $H \vdash G, B$	$\wedge$R
$\vee$L	$H, A \vee B \vdash G$ $H, A \vdash G$ $H, B \vdash G$	$H \vdash G, A \vee B$ $H \vdash G, A, B$	$\vee$R
$\Rightarrow$L	$H, A \Rightarrow B \vdash G$ $H \vdash G, A$ $H, B \vdash G$	$H \vdash G, A \Rightarrow B$ $H, A \vdash G, B$	$\Rightarrow$R
$\neg$L	$H, \neg A \vdash G$ $H \vdash G, A$	$H \vdash G, \neg A$ $H, A \vdash G$	$\neg$R
axiom	$H, A \vdash G, A$		

Each rule emphasizes the formula to be decomposed and maintains the context of the remaining hypotheses (H) and alternative goals (G). The rule $\wedge$R decomposes a conjunction $A \wedge B$ in the conclusions and returns two subgoal sequents - one in which we have to prove A and another in which we have to prove B. In each subgoal sequent the remaining assumptions and conclusions remain unchanged.

To explain the correctness of this rule let us look at the evidence. Assume we have constructed a proof for $H \vdash G, A$ and one for $H \vdash G, B$. Then the first proof gives us some evidence a for the truth of A and the second some evidence b for the truth of B (each in the context of H and G). Then putting the evidence together gives us evidence for the truth of $A \wedge B$, because the semantics of the conjunction says that $A \wedge B$ is true if both A and B are. If we use an ML-like language to describe the composition of evidence, we could say that the pair (a, b) is the evidence for the truth of $A \wedge B$ and write the rule $\wedge$R together with the associated evidence construction as follows.

$\wedge$R	$H \vdash G, A \wedge B$	$\mathtt{ev} = (a, b)$
	$H \vdash G, A$	$\mathtt{ev} = a$
	$H \vdash G, B$	$\mathtt{ev} = b$

Note that while the rule describes a top-down decomposition of sequents into subgoal sequents it describes the evidence construction in a bottom up fashion. It states that in order to prove $A \wedge B$ one has to prove EMA and B and once these proofs are completed one can compose the evidences a and b into the evidence (a, b) for $A \wedge B$. Thus the evidence for the validity of a formula can only be constructed once the top-down proof is complete, i.e. if each branch is closed by the axiom rule.

axiom is used to prove the validity nof a sequent without generating further subgoals. If the hypotheses and the conclusions both contain the same formula A then the sequent is valid.

Although the correctness of this rule is obvious it is worth looking at the evidence constructed by it. Essentially we need to provide information about the assumption that has been used to prove

the conclusion. For that purpose we need to associate assumptions with labels.[2] So if we label the assumption A by a, then the label a is sufficient evidence for the validity of the conclusion A in the sequent $H, \mathrm{A} \vdash G, \mathrm{A}$. If we integrate that information into the rule, again using an ML-like notation then the `axiom` rule appears as follows.[3]

$$\texttt{axiom} \quad H, a{:}\mathrm{A} \vdash G, \mathrm{A} \qquad \texttt{ev} = a$$

Keep in mind, however, that the explicit construction of evidence is not a part of the original formulation of Gentzen systems and has been added to the rules only to provide additional information.

Now let us go through the remaining rules.

The rule $\wedge\texttt{L}$ decomposes a conjunction $A \wedge B$ in the assumptions and splits it into two separate assumptions A and B, leaving the rest unchanged. For the construction of evidence let us assume that x is a label for $A \wedge B$, a one for A, b one for B, and $g[a,b]$ the evidence for $H, \mathrm{A}, \mathrm{B} \vdash G$, indicating that A and B may be used in the proof of G. Then by construction x must contain the same information as a and b together and the evidence for $H, \mathrm{A} \wedge \mathrm{B} \vdash G$ must indicate that x is split into a pair (a,b) which then are used in constructing $g[a,b]$. Using ML-like notation this leads to the following formulation of the rule.

$$\begin{array}{lll} \wedge\texttt{L} & H, x{:}\mathrm{A} \wedge \mathrm{B} \vdash G & \texttt{ev} = \texttt{let } x\texttt{=}(a\texttt{,}b) \texttt{ in } g[a,b] \\ & H, a{:}\mathrm{A}, b{:}\mathrm{B} \vdash G & \texttt{ev} = g[a,b] \end{array}$$

The rule $\vee\texttt{R}$ decomposes a disjunction $A \vee B$ in the conclusion into two separate conclusions A and B, leaving the rest unchanged.

For the construction of evidence let us assume that a is the evidence for the fact that A was proven in $H \vdash G, \mathrm{A}, \mathrm{B}$. In that case the evidence for $H, \vdash G, A \vee B$ would indicate that the left disjunct had been proved, denoted by $\texttt{inl}(a)$. If instead b is the evidence for the fact that B was proven, the evidence would be $\texttt{inr}(b)$. Thus there are two possible ways to construct evidence for $\vee\texttt{R}$, indicated by the rules below.

$$\begin{array}{lll} \vee\texttt{R} & H, \vdash G, A \vee B & \texttt{ev} = \texttt{inl}(a) \\ & H, \vdash G, \mathrm{A}, \mathrm{B} & \texttt{ev} = a \\ \vee\texttt{R} & H, \vdash G, A \vee B & \texttt{ev} = \texttt{inr}(b) \\ & H, \vdash G, \mathrm{A}, \mathrm{B} & \texttt{ev} = b \end{array}$$

Actually, making evidence construction for $\vee\texttt{R}$ deterministic is difficult in the presence of multiple conclusions, as the evidence itself cannot tell which of the conclusions was proven.[4] Fully deterministic construction of evidence for $\vee\texttt{R}$ is possible only in single-conclusioned Gentzen systems, which we will present next week.

[2] Labels are easier to track than, for instance, the position of the formula A in a list of assumptions. In a setting where sets of assumptions are used, labels are the only meaningful way to track formulas, especially when the formulas were generated by decomposing larger ones.

[3] Actually, this description is not entirely accurate but serves only as an illustration. The evidence tells us which assumption was used but not which of the conclusions was proved, so the information is to a certan degree incomplete. This issue affects all the rules that work explicitly on multiple conclusions – `axiom`, $\vee\texttt{R}$, $\Rightarrow\texttt{L}$ – and can only be properly resolved requiring sequents to have only a single conclusion. We will discuss this issue again when we introduce Refinement Logic

[4] A fully formal representation of evidence construction for multi-conclusioned Gentzen systems would require a nondeterministic *choice operator*. As we will later show this is due to the fact that multi-conclusioned Gentzen systems, like tableaux and truth tables, represent the so-called *classical* logic, which by nature is nonconstructive.

The rule $\vee\mathtt{L}$ decomposes a disjunction $A \vee B$ in the assumptions and returns two subgoal sequents – one in which we have to use A and another in which we have to use B, leaving the rest unchanged.

For the construction of evidence let us assume that that x is a label for $A \vee B$ in $H,\ A \vee B \vdash G$, that a one for A in the first and b one for B in the second subgoal, and that $g_1[a]$ and $g_2[b]$ are the evidences for the subgoals. Then by construction x must contain the same information as a or b together with an indicator, which of the two is the case. Thus the evidence for the main sequent must indicate that x is either $\mathrm{inl}(a)$ or $\mathrm{inr}(b)$ and that in the first case a is used in building $g_1[a]$ and in the second case b is used in building $g_2[b]$. Using ML-like notation this leads to the following formulation of the rule.

$$\begin{array}{lll} \wedge\mathtt{R} & H, x{:}A \vee B \vdash G & \mathtt{ev} = \mathtt{case}\ x\ \mathtt{of}\ \mathtt{inl}(a) \mapsto g_1[a]\ \mid\ \mathtt{inr}(b) \mapsto g_2[b] \\ & H, a{:}\mathbf{A} \vdash G & \mathtt{ev} = g_1[a] \\ & H, b{:}\mathbf{B} \vdash G & \mathtt{ev} = g_2[b] \end{array}$$

The rule $\Rightarrow\mathtt{R}$ decomposes an implication $A \Rightarrow B$ in the conclusion into the assumption A and the conclusion B, leaving the rest unchanged.

If a is a label for A and $b[a]$ the evidence for B in $H,\ \mathbf{A} \vdash G,\ \mathbf{B}$, then this means that we have found a generic way to turn evidence for A into evidence for B. In other words, the evidence for $A \Rightarrow B$, which means that whenever we have proof for A the we can construct one for B, must be a function that takes evidence a for A and computes the evidence $b[a]$ for B from that. Using ML-like notation we get

$$\begin{array}{lll} \Rightarrow\mathtt{R} & H, \vdash G, A \Rightarrow B & \mathtt{ev} = \mathtt{fun}\ a\ \rightarrow\ b[a] \\ & H, a{:}\mathbf{A} \vdash G, \mathbf{B} & \mathtt{ev} = b[a] \end{array}$$

The rule $\Rightarrow\mathtt{L}$, as explained before, decomposes an implication $A \Rightarrow B$ in the assumptions and generates two subgoal sequents – one in which A becomes an additional conclusion and one in which we may use B to prove the goal.

For the construction of evidence let us assume that f is a label for $A \Rightarrow B$ in $H,\ A \Rightarrow B \vdash G$, that a is the evidence for A in the first subgoal, that b is a label for B in the second subgoal, and that $g[b]$ is the evidence for G in that subgoal. Then building the evidence for G from the label f for $A \Rightarrow B$ in the main sequent is simply a matter of putting the evidence together: $f(a)$ will be evidence for B and can thus take the place of b in $g[b]$, resulting in the evidence $g[f(a)/b]$ for G.Using ML-like notation we get

$$\begin{array}{lll} \Rightarrow\mathtt{L} & H, f{:}A \Rightarrow B \vdash G & \mathtt{ev} = g[f(a)/b] \\ & H, \vdash G, \mathbf{A} & \mathtt{ev} = a \\ & H, b{:}\mathbf{B} \vdash G & \mathtt{ev} = g[b] \end{array}$$

This leaves us with the rules for negation.

The rule $\neg\mathtt{R}$ is similar to $\Rightarrow\mathtt{R}$, because negation $\neg A$ is usually viewed as a method for deriving a contradiction from A, The rule decomposes a negation $\neg A$ in the conclusion by generating the assumption A and removing the conclusion, leaving the rest unchanged.

If a is a label for A and $g[a]$ the evidence for G in $H,\ \mathbf{A} \vdash G$, then the evidence for $\neg A$ must be a function of the form $\mathrm{fun}\ a \rightarrow g[a]$, indicating that every evidence for A can be converted into evidence for G, the other conclusion (even if there is none, which is equivalent to falsehood).

$$\neg\mathtt{R}\quad \frac{H, \vdash G, \neg A}{H, a{:}\mathrm{A} \vdash G}\qquad \begin{array}{l}\mathtt{ev} = \mathtt{fun}\ \ a \ \rightarrow\ g[a]\\ \mathtt{ev} = g[a]\end{array}$$

Similarly, the rule $\neg\mathtt{L}$ decomposes a negation $\neg A$ in the assumptions and generates a subgoal sequent in which A becomes an additional conclusion.

For the construction of evidence let us assume that f is a label for $\neg A$ in $H, \neg A \vdash G$ and that a is the evidence for A in the subgoal. This means that applying f to a will result in a contradiction (we have evidence for A and evidence for $\neg A$, i.e. a method that can derive a contradiction from A). As a consequence any evidence that utilizes this information is evidence for G. We denote this evidence by a generic expression $\mathtt{any}[f(a)]$.

$$\neg\mathtt{L}\quad \frac{H, f{:}\neg A \vdash G}{H, \vdash G, \mathrm{A}}\qquad \begin{array}{r}\mathtt{ev} = \mathtt{any}[f(a)]\\ \mathtt{ev} = a\end{array}$$

The above constructions illustrate why the Gentzen proof rules are correct and that Gentzen systems can prove a formula by providing evidence for its truth.[5] They are not part of the original proof system but can easily be added to it in order to construct proof terms that can be checked by an independent proof checker. In proof systems capable of dealing with first-order or higher logics the evidence construction also has an interesting side-effect: it allows the extraction of programs from proofs of formulas that state the existence of certain algorithms and thus the construction of verified programs through theorem proving.

One issue that remains is to show that the sequent proof method is consistent and complete. Given the above illustrations it seems obvious that the rules are correct but that is not enough to be sure that we haven't overlooked a subtle mistake. Besides, we also need to show completeness.

We could go on and repeat what we did for tableaux: we extend the notion of boolean valuations to sequents. $H \vdash G$ is true under v_0 iff there is an $\mathrm{Y} \in G$ such that val(Y,v_0) = t, whenever val(X,v_0) = t for all $\mathrm{X} \in H$. Similarly we could now define the notions of satisfiability and tautology, and then go on to prove that a sequent $H \vdash G$ is a tautology if and only if there is a sequent proof for it. This method would, however, be quite time consuming.

There is a much simpler way, because Gentzen systems and tableau are quite similar. When we prove a formula in either system we can more or less use the same kind and order of rules. When we write the proofs side by side as in the initial example, we can also see that each sequent in the Gentzen proof contains exactly those formulas of the tableau that have not yet been decomposed ... with the once labelled F on the right and the ones labelled T on the left of the $\vdash$. Thus in a sense, Gentzen systems and the tableau method are dual to each other and we can prove consistency and completeness of Gentzen systems by showing we can translate each sequent proof into a tableau proof and vice versa.

This will be addressed in the next lecture.

[5] These notes provide much more detail than we discussed in class and go probably way beyond the level of 4th year undergraduate studies. I couldn't resist elaborating the material completely once I got started.