Intelligent Systems – Agents and Reasoning

Handout Lecture 3

1 Topics of the Lesson

Propositional Logic.

- Sequent calculus
  - Soundness Theorem
  - Completeness Theorem
    - Search procedure
    - Proof of the theorem

2 Soundness Theorem

Theorem 1 (Soundness theorem for propositional sequent calculus)**.** *If a sequent $A_1, \ldots, A_m \to B_1, \ldots, B_n$ is provable, then it is valid, i.e. for every valuation v, $v \models (A_1 \wedge \cdots \wedge A_m) \supset (B_1 \vee \cdots \vee B_n)$.*

Proof. We can prove soundness by complete induction (see Section 4) on the construction of proof trees. Hence, we need to go through three main steps.

1. We have already proved that every axiom $\Gamma, A, \Delta \to \Lambda, A, \Theta$ is valid (see Lemma 1 in the handout of Lecture 2).

2. We prove, for every rule of our calculus, that it preserves validity: for every rule, if the premises correspond to valid formulas, also the conclusion correspond to a valid formula.

We present here two examples, leaving the rest to the student:

1. $$\frac{\Gamma, A, B, \Delta \to \Lambda}{\Gamma, A \wedge B, \Delta \to \Lambda} \ (\wedge\text{: left})$$

   We assume that the formula corresponding to $\Gamma, A, B, \Delta \to \Lambda$ is valid, that is,

   $$\models (\bigwedge \Gamma \wedge A \wedge B \wedge \bigwedge \Delta) \supset (\bigvee \Lambda).$$

   We have to prove that also the formula corresponding to $\Gamma, A \wedge B, \Delta \to \Lambda$ is valid, that is immediate, since the formula is, once again, $(\bigwedge \Gamma \wedge A \wedge B \wedge \bigwedge \Delta) \supset (\bigvee \Lambda)$.

2. $$\frac{\Gamma, \Delta \rightarrow A, \Lambda \qquad \Gamma, B, \Delta \rightarrow \Lambda}{\Gamma, A \supset B, \Delta \rightarrow \Lambda} \ (\supset\text{: left})$$

We assume

$$\models (\bigwedge \Gamma \wedge \bigwedge \Delta) \supset (A \vee \bigvee \Lambda) \text{ and } \models (\bigwedge \Gamma \wedge B \wedge \bigwedge \Delta) \supset \bigvee \Lambda$$

We have to prove

$$\models (\bigwedge \Gamma \wedge (A \supset B) \wedge \bigwedge \Delta) \supset \bigvee \Lambda.$$

Let v be an interpretation s.t. $v \models (\bigwedge \Gamma \wedge (A \supset B) \wedge \bigwedge \Delta)$. Then, in order for v to satisfy $(A \supset B)$, either $v \models \neg A$ or $v \models B$. Consider each case:

- $v \models \neg A$. Since $v \models (\bigwedge \Gamma \wedge \bigwedge \Delta)$ and $\models (\bigwedge \Gamma \wedge \bigwedge \Delta) \supset (A \vee \bigvee \Lambda)$, we have that $v \models (A \vee \bigvee \Lambda)$ but $v \models \neg A$. Hence it must be the case that $v \models \bigvee \Lambda$.
- $v \models B$. Since $v \models (\bigwedge \Gamma \wedge B \wedge \bigwedge \Delta)$ and $\models (\bigwedge \Gamma \wedge B \wedge \bigwedge \Delta) \supset (\bigvee \Lambda)$, we have that $v \models \bigvee \Lambda$.

So, in all the possible cases in which $v \models (\bigwedge \Gamma \wedge (A \supset B) \wedge \bigwedge \Delta)$, we have that $v \models \bigvee \Lambda$. Hence $\models (\bigwedge \Gamma \wedge (A \supset B) \wedge \bigwedge \Delta) \supset \bigvee \Lambda$.

3. Now we need to proceed with the induction step. That is, we prove soundness by induction on the construction of the proofs, using the notion of *depth* of the proof trees. We use a function d to formalise depth:

- If the proof tree T is composed just by an axiom, $d(T) = 1$.
- If the proof tree T is obtained applying a single-premise rule to a proof tree T_1, $d(T) = d(T_1) + 1$.
- If the proof tree T is obtained applying a double-premise rule to two proof trees T_1 and T_2, $d(T) = \max\{d(T_1), d(T_2)\} + 1$.

We prove soundness by complete induction on the depth of a generic proof tree T:

- If $d(T) = 1$, then it is an axiom, and all the axioms correspond to valid formulas (Lemma 1 in Lecture 2).
- If $d(T) = n$, with $n > 1$, we have two possible cases:
  - T is obtained applying a single-premise rule to a proof tree T_1, with $d(T_1) = n - 1$.
    By induction hypothesis, we assume that the sequent at the root of T_1 corresponds to a valid formula.
    Since every single-premise rule in our system preserves validity, we can conclude that also the sequent at the root of the tree T corresponds to a valid formula.

– T is obtained applying a double-premise rule to two proof trees T_1 and T_2, with depths strictly lower that n.

We assume by induction hypothesis that the sequents at the roots of the trees T_1 and T_2 correspond to valid formulas.

Since every double-premise rule in our system preserves validity, we can conclude that also the sequent at the root of the tree T corresponds to a valid formula.

Since we have proven that every proof tree of depth 1 corresponds to a valid formula and every step in the construction of a proof tree preserves validity, we have proven that every proof tree proves a sequent that corresponds to avalid formula.

□

Corollary 1. *If $\rightarrow A$ is provable, then A is a tautology.*

Proof. It is just a special case of Theorem 1.

□

Exercise 1. *Prove that the following rules preserve validity:*

i- $\dfrac{\Gamma, A, \Delta \rightarrow \Lambda, \Theta}{\Gamma, \Delta \rightarrow \Lambda, \neg A, \Theta}$ *($\neg$: right)*

ii- $\dfrac{\Gamma \rightarrow \Delta, A, B, \Lambda}{\Gamma \rightarrow \Delta, A \vee B, \Lambda}$ *($\vee$: right)*

3 Completeness Theorem

We need to prove the completeness of the sequent calculus, that is, for every formula A

$$\textit{If } \models A, \textit{ then } \vdash \rightarrow A$$

3.1 Search Procedure

We define a deterministic procedure to build proof trees. Such a procedure is a preliminary step to prove completeness.

The procedure is obtained combining the procedure *Search* and the procedure *Expand*.

- The overall procedure starts with a sequent and tries, step by step, to build a proof tree for it.
- The procedure *Search* check all the leaves of a tree of sequents: if the sequent in a leaf is not *finished*, the procedure *Expand* is activated.
- Each leaf of a tree is *finished* iff:
  - it is an axiom, or

– all propositions in it are propositional symbols.

- The overall procedure goes on building the proof tree until every leaf of the tree is finished

Procedure Search

procedure *search*($\Gamma \to \Delta$: *sequent*; **var** T : *tree*)**;**
  begin
    let T be the one-node tree labeled with $\Gamma \to \Delta$;
    while not all leaves of T are finished **do**
      $T_0 := T$;
      for each *leaf node of T_0 (in lexicographic order of tree addresses)* **do**
        if not *finished*(*node*) **then**
          expand(*node*,T)
        endif
      endfor
    endwhile;
    if *all leaves are axioms*
    then
      write ('$\Gamma \to \Delta$ *is a valid sequent*')
    else
      write ('$\Gamma \to \Delta$ *is falsifiable*')
    endif
  end

- The procedure *Expand* receives from the procedure *Search* an unfinished sequent and expands it into a subtree, trying to have only finished sequents in the leaves.
- The output of procedure *Expand* is added to the original tree.
- The procedure *Search* checks whether all the leaves are finished, and, if not, activates *Expand*.
- Once all the sequents in the leaves are finished, the procedure terminates.

Procedure Expand

procedure *expand*(*node* : *tree-address*; **var** T : *tree*)**;**
  begin
    let $A_1, \ldots, A_m \to B_1, \ldots, B_n$ *be the label of node*;
    let S be the one-node tree labeled with $A_1, \ldots, A_m \to B_1, \ldots, B_n$;
      for $i := 1$ **to** m **do**
        if *nonatomic*(A_i) **then**
          $S :=$ *the new tree obtained from S by applying to the descendant of A_i in every nonaxiom leaf of S the left rule applicable to A_i*;
        endif
      endfor;
      for $i := 1$ **to** n **do**
        if *nonatomic*(B_i) **then**

S := *the new tree obtained from* S *by applying to the descendant of* B_i *in every nonaxiom leaf of* S *the right rule applicable to* B_i;
endif
endfor;
T := *the new tree obtained from* T *by substituting node by* S *in* T
end

Exercise 2. *Build the proof trees of the following sequents using the procedures* Search *and* Expand*.*

- $P \wedge R, P \supset Q \to Q, R \vee \neg T$
- $Q \supset R, \neg R \to Q \wedge T$

3.2 Proof of the Completeness Theorem

We need to prove completeness.

Theorem 2 (Completeness)**.** *The search procedure terminates for every finite input sequent. If the input sequent* $\Gamma \to \Delta$ *is valid, the search procedure produces a proof tree for* $\Gamma \to \Delta$*.*

The theorem can be divided into two parts:

1. The search procedure always terminates.
2. Every valid sequent is provable.

3.2.1 Termination

To prove termination, we go through the following steps:

- We define the *complexity* of a sequent to be the number of logical connectives occurring in the sequent.

For example, the complexity of a sequent composed only of propositional symbols $P_1, \ldots P_n \to P_{n+1}, \ldots, P_m$ is 0 while the complexity of the dequent $(P_1 \wedge \neg P_2) \vee P_3, \neg P_4 \to P_1, P_3 \vee \neg P_5$ is 6.

- For every call to procedure *expand*, the complexity of every upper sequent involved in applying a rule is strictly smaller than the complexity of the lower sequent.

This can easily been proven noticing that for every rule of the sequent calculus the complexity of the premises (the sequents above the derivation line) is always lower than the complexity of the conclusion (the sequent below the derivation line). E.g., in the rule

$$\frac{\Gamma, A, B, \Delta \to \Lambda}{\Gamma, A \wedge B, \Delta \to \Lambda} \ (\wedge\text{: left})$$

The complexity of the conclusion is n iff the complexity of the premise is $n-1$.

- In every execution of the search procedure, either all leaves will become axioms or their complexity will become 0, hence the *while* loop in the the search procedure will always terminate.

3.2.2 Every valid sequent is provable

The proof of completeness needs Lemma 3, that, in turn, needs Lemma 1 and Lemma 2 to be proved.

Lemma 1. *If the sequent labeling a finished leaf is not an axiom, it is falsifiable.*

Proof. If the sequent labeling a finished leaf is not an axiom, it contains only propositional symbols. Define $v(P) = \mathbf{T}$ if P appears on the left of the sequent and $v(P) = \mathbf{F}$ if P appears on the right of the sequent. Then v falsifies the sequent. □

Lemma 2. *For every inference rule and every valuation v, v makes the conclusion of the rule true if and only if v makes all premises of that rule true.*

Proof. For example, consider the rule ($\vee$: left):

$$\frac{\Gamma, A, \Delta \rightarrow \Lambda \qquad \Gamma, B, \Delta \rightarrow \Lambda}{\Gamma, A \vee B, \Delta \rightarrow \Lambda} \ (\vee\text{: left})$$

Assume that a valuation v satisfies both $\bigwedge \Gamma \wedge A \wedge \bigwedge \Delta \supset \bigvee \Lambda$ and $\bigwedge \Gamma \wedge B \wedge \bigwedge \Delta \supset \bigvee \Lambda$, but assume that v does not satisfy the formula correspond to the sequent $\Gamma, A \vee B, \Delta \rightarrow \Lambda$, that is

$$v \models \bigwedge \Gamma \wedge (A \vee B) \wedge \bigwedge \Delta \wedge \neg \bigvee \Lambda$$

Since $v \models \bigwedge \Gamma \wedge (A \vee B) \wedge \bigwedge \Delta$, from the definition of disjunction we have two possibilities: either

$$v \models \bigwedge \Gamma \wedge A \wedge \bigwedge \Delta \text{ or } v \models \bigwedge \Gamma \wedge B \wedge \bigwedge \Delta$$

In both cases we have to conclude, by the definition of implication, that $v \models \bigvee \Lambda$. So, $v \models \bigvee \Lambda$ and $v \models \neg \bigvee \Lambda$ at the same time: contradiction.

In the other direction, assume that v satisfies the conclusion, that is $v \models \bigwedge \Gamma \wedge (A \vee B) \wedge \bigwedge \Delta \supset \bigvee \Lambda$, but falsifies one of the premises. W.l.o.g., assume

$$v \models \bigwedge \Gamma \wedge A \wedge \bigwedge \Delta \wedge \neg \bigvee \Lambda$$

Hence $v \models \neg \bigvee \Lambda$ and $v \models \bigwedge \Gamma \wedge A \wedge \bigwedge \Delta$.

By the definition of disjunction, $v \models \bigwedge \Gamma \wedge A \wedge \bigwedge \Delta$ implies $v \models \bigwedge \Gamma \wedge (A \vee B) \wedge \bigwedge \Delta$, which in turn implies $v \models \bigvee \Lambda$. Again, contradiction.

We have to make the same kind of proof for every rule of sequent calculus. □

Lemma 3. *Given any deduction tree T, a valuation v falsifies the sequent $\Gamma \rightarrow \Delta$ labeling the root of T if and only if v falsifies some sequent labeling a leaf of T.*

Proof. It is easily provable by complete induction in the depth of a tree (the function d defined in the proof of Theorem 1).

Consider a tree T:

- Lemma 1 tells us that if $d(T) = 1$, Lemma 3 holds.
- Lemma 2 tells us that if Lemma 3 holds for all the trees of depth $m \leq n$, it holds also for the trees of depth $n + 1$.

□

Given Lemma 3, the proof of completeness is straightforward.

- If the sequent $\Gamma \to \Delta$ is valid, no valuation falsifies it.
- So by lemma 3, no valuation falsifies a leaf of the tree produced by *Search*.
- By lemma 1, all leafs are axioms, i.e. it is a proof tree.

Exercise 3. *Prove for the following rule that for every valuation v, v makes the conclusion true if and only if it makes both the premises true.*

$$\frac{\Gamma, \Delta \to A, \Lambda \qquad \Gamma, B, \Delta \to \Lambda}{\Gamma, A \supset B, \Delta \to \Lambda}\ (\supset\text{: } \textit{left})$$

4 Notes: Proofs by Induction

Assume you have a class C of objects, and you want to prove that a property P holds for all the members of such a class.

Let $f : C \mapsto \mathbb{N}$ be a surjective function that associates to every member of C a natural number.

Schema of a Proof by Induction.

1. Prove that the property P holds for each element $x \in C$ such that $f(x) = 1$;
2. Prove that for every $n \in \mathbb{N}$, if the property P holds for every $x \in C$ such that $f(x) = n$, then it holds for every $y \in C$ such that $f(y) = n + 1$

In this way we prove that the property P holds for all the elements of the class C.

Schema of a Proof by Complete Induction.

1. Prove that the property P holds for each element $x \in C$ such that $f(x) = 1$;
2. Prove that for every $n \in \mathbb{N}$, if the property P holds for every $x \in C$ such that $f(x) \leq n$, then it holds for every $y \in C$ such that $f(y) = n + 1$

In this way we prove that the property P holds for all the elements of the class C.