

Logik für Informatiker Logic for computer scientists

Till Mossakowski

WiSe 2013/14

Proofs for Boolean Logic

Logical consequence

- ① Q is a **logical consequence** of P_1, \dots, P_n , if all worlds that make P_1, \dots, P_n true also make Q true.
- ② Q is a **tautological consequence** of P_1, \dots, P_n , if all valuations of atomic formulas with truth values that make P_1, \dots, P_n true also make Q true.
- ③ Q is a **TW-logical consequence** of P_1, \dots, P_n , if all worlds from Tarski's world that make P_1, \dots, P_n true also make Q true.

The difference lies in the **set of worlds** that is considered:

- ① **all worlds** (whatever this exactly means ...)
- ② all **valuations** of atomic formulas with truth values
 (= **rows** in the truth table)
- ③ all **block worlds from Tarski's world**

Proofs

- With proofs, we try to show (tauto)logical consequence
- Truth-table method can lead to very large tables, proofs are often shorter
- Proofs are also available for consequence in full first-order logic, not only for tautological consequence

Limits of the truth-table method

- 1 truth-table method leads to **exponentially growing** tables
 - 20 atomic sentences \Rightarrow more than 1.000.000 rows
- 2 truth-table method cannot be extended to **first-order logic**
 - **model checking** can overcome the first limitation (up to 1.000.000 atomic sentences)
 - **proofs** can overcome both limitations

Proofs

- A proof consists of a sequence of **proof steps**
- Each proof step is known to be valid and should
 - be significant but easily understood, in **informal** proofs,
 - follow some **proof rule**, in **formal** proofs.
- Some valid patterns of inference that generally go unmentioned in informal (but not in formal) proofs:
 - From $P \wedge Q$, infer P .
 - From P and Q , infer $P \wedge Q$.
 - From P , infer $P \vee Q$.

Proof by cases (disjunction elimination)

To prove S from $P_1 \vee \dots \vee P_n$, prove S from each of P_1, \dots, P_n .

Claim: there are irrational numbers b and c such that b^c is rational.

Proof: $\sqrt{2}^{\sqrt{2}}$ is either rational or irrational.

Case 1: If $\sqrt{2}^{\sqrt{2}}$ is rational: take $b = c = \sqrt{2}$.

Case 2: If $\sqrt{2}^{\sqrt{2}}$ is irrational: take $b = \sqrt{2}^{\sqrt{2}}$ and $c = \sqrt{2}$.

$$\text{Then } b^c = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{(\sqrt{2} \cdot \sqrt{2})} = \sqrt{2}^2 = 2.$$

Proof by contradiction

To prove $\neg S$, assume S and prove a contradiction \perp .

(\perp may be inferred from P and $\neg P$.)

Assume $Cube(c) \vee Dodec(c)$ and $Tet(b)$.

Claim: $\neg(b = c)$.

Proof: Let us assume $b = c$.

Case 1: If $Cube(c)$, then by $b = c$, also $Cube(b)$, which contradicts $Tet(b)$.

Case 2: $Dodec(c)$ similarly contradicts $Tet(b)$.

In both case, we arrive at a contradiction. Hence, our assumption $b = c$ cannot be true, thus $\neg(b = c)$.

Arguments with inconsistent premises

A proof of a contradiction \perp from premises P_1, \dots, P_n (without additional assumptions) shows that the premises are **inconsistent**. An argument with inconsistent premises is always **valid**, but more importantly, always **unsound**.

Home(max) \vee Home(claire)

\neg Home(max)

\neg Home(claire)

Home(max) \wedge Happy(carl)

Arguments without premises

A proof without any premises shows that its conclusion is a **logical truth**.

Example: $\neg(P \wedge \neg P)$.

Formal Proofs and Boolean Logic

Formal proofs in Fitch

- Well-defined set of **formal proof rules**
- Formal proofs in Fitch can be **mechanically checked**
- For each connective, there is
 - an **introduction rule**, e.g. “from P , infer $P \vee Q$ ”.
 - an **elimination rule**, e.g. “from $P \wedge Q$, infer P ”.

Conjunction Elimination (\wedge Elim)

$$\triangleright \left| \begin{array}{l} P_1 \wedge \dots \wedge P_i \wedge \dots \wedge P_n \\ \vdots \\ P_i \end{array} \right.$$

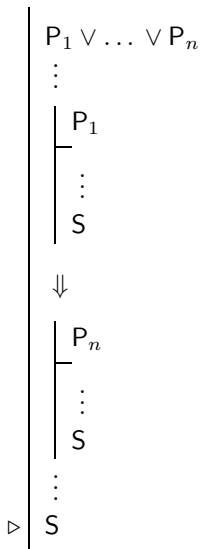
Conjunction Introduction (\wedge Intro)

$$\begin{array}{l} | \\ | \\ | \\ | \\ | \\ | \\ \triangleright | \end{array} \begin{array}{l} P_1 \\ \Downarrow \\ P_n \\ \vdots \\ P_1 \wedge \dots \wedge P_n \end{array}$$

Disjunction Introduction (\vee Intro)

$$\triangleright \left| \begin{array}{l} P_i \\ \vdots \\ P_1 \vee \dots \vee P_i \vee \dots \vee P_n \end{array} \right.$$

Disjunction Elimination (\vee Elim)



The proper use of subproofs

1. $(B \wedge A) \vee (A \wedge C)$	
2. $B \wedge A$	
3. B	\wedge Elim: 2
4. A	\wedge Elim: 2
5. $A \wedge C$	
6. A	\wedge Elim: 5
7. A	\vee Elim: 1, 2–4, 5–6
8. $A \wedge B$	\wedge Intro: 7, 3

The proper use of subproofs (cont'd)

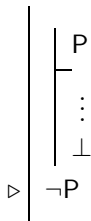
- In justifying a step of a subproof, you may cite any **earlier step** contained in the main proof, or in any subproof whose assumption is **still in force**. You may **never** cite individual steps inside a subproof that has **already ended**.
- Fitch enforces this automatically by not permitting the citation of individual steps inside subproofs that have ended.

\perp Introduction (\perp Intro)

$$\begin{array}{|l} P \\ \vdots \\ \neg P \\ \vdots \\ \perp \end{array}$$

\triangleright

Negation Introduction (\neg Intro)



Negation Elimination (\neg Elim)

$$\begin{array}{|l} \neg\neg P \\ \vdots \\ P \end{array}$$

▷

\perp Elimination (\perp Elim)

$$\begin{array}{c} \vdots \\ \perp \\ \vdots \\ \text{P} \end{array}$$

▷

Strategies and tactics in Fitch

- 1 **Understand** what the sentences are saying.
- 2 **Decide** whether you think the conclusion follows from the premises.
- 3 If you think it does not follow, or are not sure, try to find a **counterexample**.
- 4 If you think it does follow, try to give an **informal proof**.
- 5 If a **formal proof** is called for, use the **informal proof to guide** you in finding one.
- 6 In giving consequence proofs, both formal and informal, don't forget the tactic of **working backwards**.
- 7 In working backwards, though, always check that your **intermediate goals are consequences** of the available information.

Strategies in Fitch, cont'd

- Always try to **match** the situation in your proof with the **rules** in the book (see book appendix for a complete list)
- Look at the **main connective** in a **premise**, apply the corresponding **elimination rule** (forwards)
- Or: look at the **main connective** in the **conclusion**, apply the corresponding **introduction rule** (backwards)