Soundness and Completeness of Implicational Logic

Asta Halkjær From   Jørgen Villadsen

March 17, 2025

Contents

Abstract

This work is a formalization of soundness and completeness of the Bernays-Tarski axiom system for classical implicational logic. The completeness proof is constructive following the approach by László Kalmár, Elliott Mendelson and others. The result can be extended to full classical propositional logic by uncommenting a few lines for falsehood.

1 Formalization of the Bernays-Tarski Axiom System for Classical Implicational Logic

1.1 Syntax, Semantics and Axiom System

theory *Implicational-Logic* **imports** *Main* **begin**

datatype *form* =

Pro nat (‹·›) |
Imp form form (**infixr** ‹→› *55*)

primrec *semantics* (**infix** ‹$\models$› *50*) **where**

‹$I \models \cdot\ n = I\ n$› |
‹$I \models p \to q = (I \models p \longrightarrow I \models q)$›

inductive *Ax* (‹$\vdash$ -› *50*) **where**

Simp: ‹$\vdash p \to q \to p$› |
Tran: ‹$\vdash (p \to q) \to (q \to r) \to p \to r$› |
MP: ‹$\vdash p \to q \Longrightarrow \vdash p \Longrightarrow \vdash q$› |
PR: ‹$\vdash (p \to q) \to p \Longrightarrow \vdash p$›

1.2 Soundness and Derived Formulas

theorem *soundness*: ‹$\vdash p \Longrightarrow I \models p$›
⟨*proof*⟩

lemma *Swap*: ‹$\vdash (p \to q \to r) \to q \to p \to r$›
⟨*proof*⟩

lemma *Peirce*: ‹$\vdash ((p \to q) \to p) \to p$›
⟨*proof*⟩

lemma *Hilbert*: ‹$\vdash (p \to p \to q) \to p \to q$›
⟨*proof*⟩

lemma *Id*: ‹$\vdash p \to p$›
⟨*proof*⟩

lemma *Tran'*: ‹$\vdash (q \to r) \to (p \to q) \to p \to r$›
⟨*proof*⟩

lemma *Frege*: ‹$\vdash (p \to q \to r) \to (p \to q) \to p \to r$›
⟨*proof*⟩

lemma *Imp1*: ‹$\vdash (q \to s) \to ((q \to r) \to s) \to s$›
⟨*proof*⟩

lemma *Imp2*: ‹$\vdash ((r \to s) \to s) \to ((q \to r) \to s) \to s$›
⟨*proof*⟩

lemma *Imp3*: ‹$\vdash ((q \to s) \to s) \to (r \to s) \to (q \to r) \to s$›
⟨*proof*⟩

1.3 Completeness and Main Theorem

fun *pros* **where**
‹*pros* $(p \to q)$ = *remdups* (*pros* p @ *pros* q)› |
‹*pros* p = (*case* p *of* $(\cdot\ n) \Rightarrow [n]$ | - $\Rightarrow$ [])›

lemma *distinct-pros*: ‹$distinct\ (pros\ p)$›
⟨*proof*⟩

primrec *imply* (**infixr** ‹⇝› *56*) **where**
‹$[] \leadsto q = q$› |
‹$p\ \#\ ps \leadsto q = p \to ps \leadsto q$›

lemma *imply-append*: ‹$ps\ @\ qs \leadsto r = ps \leadsto qs \leadsto r$›
⟨*proof*⟩

abbreviation *Ax-assms* (**infix** ‹⊢› *50*) **where** ‹$ps \vdash q \equiv \vdash ps \leadsto q$›

lemma *imply-Cons*: ‹$ps \vdash q \Longrightarrow p\ \#\ ps \vdash q$›
⟨*proof*⟩

lemma *imply-head*: ‹$p\ \#\ ps \vdash p$›
⟨*proof*⟩

lemma *imply-mem*: ‹$p \in set\ ps \Longrightarrow ps \vdash p$›
⟨*proof*⟩

lemma *imply-MP*: ‹$\vdash ps \leadsto (p \to q) \to ps \leadsto p \to ps \leadsto q$›
⟨*proof*⟩

lemma *MP'*: ‹$ps \vdash p \to q \Longrightarrow ps \vdash p \Longrightarrow ps \vdash q$›
⟨*proof*⟩

lemma *imply-swap-append*: ‹$ps\ @\ qs \vdash r \Longrightarrow qs\ @\ ps \vdash r$›
⟨*proof*⟩

lemma *imply-deduct*: ‹$p\ \#\ ps \vdash q \Longrightarrow ps \vdash p \to q$›
⟨*proof*⟩

lemma *add-imply* [*simp*]: ‹$\vdash p \Longrightarrow ps \vdash p$›
⟨*proof*⟩

lemma *imply-weaken*: ‹$ps \vdash p \Longrightarrow set\ ps \subseteq set\ ps' \Longrightarrow ps' \vdash p$›
⟨*proof*⟩

abbreviation ‹$lift\ t\ s\ p \equiv if\ t\ then\ (p \to s) \to s\ else\ p \to s$›

abbreviation ‹$lifts\ I\ s \equiv map\ (\lambda n.\ lift\ (I\ n)\ s\ (\cdot\ n))$›

lemma *lifts-weaken*: ‹$lifts\ I\ s\ l \vdash p \Longrightarrow set\ l \subseteq set\ l' \Longrightarrow lifts\ I\ s\ l' \vdash p$›
⟨*proof*⟩

lemma *lifts-pros-lift*: ‹$lifts\ I\ s\ (pros\ p) \vdash lift\ (I \models p)\ s\ p$›
⟨*proof*⟩

lemma *lifts-pros*: $‹I \models p \Longrightarrow \mathit{lifts}\ I\ p\ (\mathit{pros}\ p) \vdash p›$
$\langle \mathit{proof} \rangle$

theorem *completeness*: $‹\forall I.\ I \models p \Longrightarrow \vdash p›$
$\langle \mathit{proof} \rangle$

theorem *main*: $‹(\vdash p) = (\forall I.\ I \models p)›$
$\langle \mathit{proof} \rangle$

1.4 Reference

Wikipedia https://en.wikipedia.org/wiki/Implicational_propositional_calculus
July 2022

end

2 Formalization of ukasiewicz's Axiom System from 1924 for Classical Propositional Logic

2.1 Syntax, Semantics and Axiom System

theory *Implicational-Logic-Appendix* **imports** *Main* **begin**

datatype *form* =
  Pro nat $(‹\cdot›)$ |
  Neg form $(‹\sim›)$ |
  Imp form form (**infixr** $‹\rightarrow›$ *55*)

primrec *semantics* (**infix** $‹\models›$ *50*) **where**
  $‹I \models \cdot\ n = I\ n›$ |
  $‹I \models \sim p = (\neg\ I \models p)›$ |
  $‹I \models p \rightarrow q = (I \models p \longrightarrow I \models q)›$

inductive *Ax* $(‹\vdash -›\ 50)$ **where**
  01: $‹\vdash (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow p \rightarrow r›$ |
  02: $‹\vdash (\sim p \rightarrow p) \rightarrow p›$ |
  03: $‹\vdash p \rightarrow \sim p \rightarrow q›$ |
  MP: $‹\vdash p \rightarrow q \Longrightarrow \vdash p \Longrightarrow \vdash q›$

2.2 Soundness and Derived Formulas

theorem *soundness*: $‹\vdash p \Longrightarrow I \models p›$
$\langle \mathit{proof} \rangle$

lemma *04*: $‹\vdash (((q \rightarrow r) \rightarrow p \rightarrow r) \rightarrow s) \rightarrow (p \rightarrow q) \rightarrow s›$
$\langle \mathit{proof} \rangle$

lemma *05*: $‹\vdash (p \rightarrow q \rightarrow r) \rightarrow (s \rightarrow q) \rightarrow p \rightarrow s \rightarrow r›$

⟨*proof*⟩

lemma *06*: ‹$\vdash (p \to q) \to ((p \to r) \to s) \to (q \to r) \to s$›
⟨*proof*⟩

lemma *07*: ‹$\vdash (t \to (p \to r) \to s) \to (p \to q) \to t \to (q \to r) \to s$›
⟨*proof*⟩

lemma *09*: ‹$\vdash ((\sim p \to q) \to r) \to p \to r$›
⟨*proof*⟩

lemma *10*: ‹$\vdash p \to ((\sim p \to p) \to p) \to (q \to p) \to p$›
⟨*proof*⟩

lemma *11*: ‹$\vdash (q \to (\sim p \to p) \to p) \to (\sim p \to p) \to p$›
⟨*proof*⟩

lemma *12*: ‹$\vdash t \to (\sim p \to p) \to p$›
⟨*proof*⟩

lemma *13*: ‹$\vdash (\sim p \to q) \to t \to (q \to p) \to p$›
⟨*proof*⟩

lemma *14*: ‹$\vdash ((t \to (q \to p) \to p) \to r) \to (\sim p \to q) \to r$›
⟨*proof*⟩

lemma *15*: ‹$\vdash (\sim p \to q) \to (q \to p) \to p$›
⟨*proof*⟩

lemma *16*: ‹$\vdash p \to p$›
⟨*proof*⟩

lemma *17*: ‹$\vdash p \to (q \to p) \to p$›
⟨*proof*⟩

lemma *18*: ‹$\vdash q \to p \to q$›
⟨*proof*⟩

lemma *19*: ‹$\vdash ((p \to q) \to r) \to q \to r$›
⟨*proof*⟩

lemma *20*: ‹$\vdash p \to (p \to q) \to q$›
⟨*proof*⟩

lemma *21*: ‹$\vdash (p \to q \to r) \to q \to p \to r$›
⟨*proof*⟩

lemma *22*: ‹$\vdash (q \to r) \to (p \to q) \to p \to r$›
⟨*proof*⟩

lemma *23*: ‹$\vdash ((q \to p \to r) \to s) \to (p \to q \to r) \to s$›
⟨*proof*⟩

lemma *24*: ‹$\vdash ((p \to q) \to p) \to p$›
⟨*proof*⟩

lemma *25*: ‹$\vdash ((p \to r) \to s) \to (p \to q) \to (q \to r) \to s$›
⟨*proof*⟩

lemma *26*: ‹$\vdash ((p \to q) \to r) \to (r \to p) \to p$›
⟨*proof*⟩

lemma *28*: ‹$\vdash (((r \to p) \to p) \to s) \to ((p \to q) \to r) \to s$›
⟨*proof*⟩

lemma *29*: ‹$\vdash ((p \to q) \to r) \to (p \to r) \to r$›
⟨*proof*⟩

lemma *31*: ‹$\vdash (p \to s) \to ((p \to q) \to r) \to (s \to r) \to r$›
⟨*proof*⟩

lemma *32*: ‹$\vdash ((p \to q) \to r) \to (p \to s) \to (s \to r) \to r$›
⟨*proof*⟩

lemma *33*: ‹$\vdash (p \to s) \to (s \to q \to p \to r) \to q \to p \to r$›
⟨*proof*⟩

lemma *34*: ‹$\vdash (s \to q \to p \to r) \to (p \to s) \to q \to p \to r$›
⟨*proof*⟩

lemma *35*: ‹$\vdash (p \to q \to r) \to (p \to q) \to p \to r$›
⟨*proof*⟩

lemma *36*: ‹$\vdash \sim p \to p \to q$›
⟨*proof*⟩

lemmas
Tran = *01* **and**
Clavius = *02* **and**
Expl = *03* **and**
Frege′ = *05* **and**
Clavius′ = *15* **and**
Id = *16* **and**
Simp = *18* **and**
Swap = *21* **and**
Tran′ = *22* **and**
Peirce = *24* **and**
Frege = *35* **and**

$Expl' = 36$

lemma *Neg1*: $\langle\vdash (q \to s) \to (\sim q \to s) \to s\rangle$
$\langle proof \rangle$

lemma *Neg2*: $\langle\vdash ((q \to s) \to s) \to \sim q \to s\rangle$
$\langle proof \rangle$

lemma *Imp1*: $\langle\vdash (q \to s) \to ((q \to r) \to s) \to s\rangle$
$\langle proof \rangle$

lemma *Imp2*: $\langle\vdash ((r \to s) \to s) \to ((q \to r) \to s) \to s\rangle$
$\langle proof \rangle$

lemma *Imp3*: $\langle\vdash ((q \to s) \to s) \to (r \to s) \to (q \to r) \to s\rangle$
$\langle proof \rangle$

2.3 Completeness and Main Theorem

primrec *pros* **where**
$\langle pros\ (\cdot\ n) = [n]\rangle$ |
$\langle pros\ (\sim p) = pros\ p\rangle$ |
$\langle pros\ (p \to q) = remdups\ (pros\ p\ @\ pros\ q)\rangle$

lemma *distinct-pros*: $\langle distinct\ (pros\ p)\rangle$
$\langle proof \rangle$

primrec *imply* (**infixr** $\langle\rightsquigarrow\rangle$ *56*) **where**
$\langle [] \rightsquigarrow q = q\rangle$ |
$\langle p\ \#\ ps \rightsquigarrow q = p \to ps \rightsquigarrow q\rangle$

lemma *imply-append*: $\langle ps\ @\ qs \rightsquigarrow r = ps \rightsquigarrow qs \rightsquigarrow r\rangle$
$\langle proof \rangle$

abbreviation *Ax-assms* (**infix** $\langle\vdash\rangle$ *50*) **where** $\langle ps \vdash q \equiv \vdash ps \rightsquigarrow q\rangle$

lemma *imply-Cons*: $\langle ps \vdash q \Longrightarrow p\ \#\ ps \vdash q\rangle$
$\langle proof \rangle$

lemma *imply-head*: $\langle p\ \#\ ps \vdash p\rangle$
$\langle proof \rangle$

lemma *imply-mem*: $\langle p \in set\ ps \Longrightarrow ps \vdash p\rangle$
$\langle proof \rangle$

lemma *imply-MP*: $\langle\vdash ps \rightsquigarrow (p \to q) \to ps \rightsquigarrow p \to ps \rightsquigarrow q\rangle$
$\langle proof \rangle$

lemma *MP'*: $\langle ps \vdash p \to q \Longrightarrow ps \vdash p \Longrightarrow ps \vdash q\rangle$

⟨*proof*⟩

lemma *imply-swap-append*: ‹ps @ $qs \vdash r \Longrightarrow qs$ @ $ps \vdash r$›
⟨*proof*⟩

lemma *imply-deduct*: ‹p # $ps \vdash q \Longrightarrow ps \vdash p \to q$›
⟨*proof*⟩

lemma *add-imply* [*simp*]: ‹$\vdash p \Longrightarrow ps \vdash p$›
⟨*proof*⟩

lemma *imply-weaken*: ‹$ps \vdash p \Longrightarrow set\ ps \subseteq set\ ps' \Longrightarrow ps' \vdash p$›
⟨*proof*⟩

abbreviation ‹$lift\ t\ s\ p \equiv if\ t\ then\ (p \to s) \to s\ else\ p \to s$›

abbreviation ‹$lifts\ I\ s \equiv map\ (\lambda n.\ lift\ (I\ n)\ s\ (\cdot\ n))$›

lemma *lifts-weaken*: ‹$lifts\ I\ s\ l \vdash p \Longrightarrow set\ l \subseteq set\ l' \Longrightarrow lifts\ I\ s\ l' \vdash p$›
⟨*proof*⟩

lemma *lifts-pros-lift*: ‹$lifts\ I\ s\ (pros\ p) \vdash lift\ (I \models p)\ s\ p$›
⟨*proof*⟩

lemma *lifts-pros*: ‹$I \models p \Longrightarrow lifts\ I\ p\ (pros\ p) \vdash p$›
⟨*proof*⟩

theorem *completeness*: ‹$\forall I.\ I \models p \Longrightarrow \vdash p$›
⟨*proof*⟩

theorem *main*: ‹$(\vdash p) = (\forall I.\ I \models p)$›
⟨*proof*⟩

2.4 Reference

Numbered lemmas are from Jan ukasiewicz: Elements of Mathematical Logic (English Tr. 1963)

end
theory *Implicational-Logic-Sequent-Calculus* **imports** *Main* **begin**

datatype *form* =
Pro nat (‹·›) |
Imp form form (**infixr** ‹→› *100*)

primrec *semantics* (**infix** ‹⊨› *50*) **where**
‹$I \models \cdot n = I\ n$› |
‹$I \models p \to q = (I \models p \longrightarrow I \models q)$›

abbreviation *sc* (‹⟦-⟧›) **where** ‹$[\![I]\!]\ X\ Y \equiv (\forall p \in set\ X.\ I \models p) \longrightarrow (\exists q \in set$

Y. *I* ⊨ *q*)›

inductive *SC* (**infix** ‹≫› *50*) **where**
Imp-L: ‹*p* → *q* # *X* ≫ *Y*› **if** ‹*X* ≫ *p* # *Y*› **and** ‹*q* # *X* ≫ *Y*› |
Imp-R: ‹*X* ≫ *p* → *q* # *Y*› **if** ‹*p* # *X* ≫ *q* # *Y*› |
Set-L: ‹*X′* ≫ *Y*› **if** ‹*X* ≫ *Y*› **and** ‹*set X′* = *set X*› |
Set-R: ‹*X* ≫ *Y′*› **if** ‹*X* ≫ *Y*› **and** ‹*set Y′* = *set Y*› |
Basic: ‹*p* # - ≫ *p* # -›

function *mp* **where**
‹*mp A B* (*p* → *q* # *C*) [] = (*mp A B C* [*p*] ∧ *mp A B* (*q* # *C*) [])› |
‹*mp A B C* (*p* → *q* # *D*) = *mp A B* (*p* # *C*) (*q* # *D*)› |
‹*mp A B* [] [] = (*set A* ∩ *set B* ≠ {})› |
‹*mp A B* (·*n* # *C*) [] = *mp* (*n* # *A*) *B C* []› |
‹*mp A B C* (·*n* # *D*) = *mp A* (*n* # *B*) *C D*›
⟨*proof*⟩

termination
⟨*proof*⟩

lemma *main*: ‹(∀ *I*. ⟦*I*⟧ (*map* · *A* @ *C*) (*map* · *B* @ *D*)) ⟷ *mp A B C D*›
⟨*proof*⟩

definition *prover* (‹⊢›) **where** ‹⊢ *p* ≡ *mp* [] [] [] [*p*]›

theorem *prover-correct*: ‹⊢ *p* ⟷ (∀ *I*. *I* ⊨ *p*)›
⟨*proof*⟩

export-code ⊢ **in** *SML*

lemma *MP*: ‹*mp A B C D* ⟹ *set X* ⊇ *set* (*map* · *A* @ *C*) ⟹ *set Y* ⊇ *set* (*map* · *B* @ *D*) ⟹ *X* ≫ *Y*›
⟨*proof*⟩

theorem *OK*: ‹(∀ *I*. ⟦*I*⟧ *X Y*) ⟷ *X* ≫ *Y*›
⟨*proof*⟩

corollary ‹[] ≫ [*p*] ⟷ (∀ *I*. *I* ⊨ *p*)›
⟨*proof*⟩

proposition ‹[] ≫ [*p* → *p*]›
⟨*proof*⟩

proposition ‹[] ≫ [*p* → (*p* → *q*) → *q*]›
⟨*proof*⟩

proposition ‹[] ≫ [*p* → *q* → *q* → *p*]›
⟨*proof*⟩

proposition ‹[] ≫ [(p → q) → p → q]›
⟨proof⟩

proposition ‹[] ≫ [p → p → q → q]›
⟨proof⟩

proposition ‹[] ≫ [(p → p → q) → p → q]›
⟨proof⟩

proposition ‹[] ≫ [p → q → p]›
⟨proof⟩

proposition ‹[] ≫ [(p → r) → (r → q) → p → q]›
⟨proof⟩

proposition ‹[] ≫ [((p → q) → p) → p]›
⟨proof⟩

end
theory *Implicational-Logic-Natural-Deduction* **imports** *Main* **begin**

datatype *form* =
  Pro nat (‹·›) |
  Imp form form (**infixr** ‹→› *100*)

primrec *semantics* (**infix** ‹⊨› *50*) **where**
  ‹I ⊨ ·n = I n› |
  ‹I ⊨ p → q = (I ⊨ p ⟶ I ⊨ q)›

inductive *Calculus* (**infix** ‹⇝› *50*) **where**
  Assm: ‹p ∈ set A ⟹ A ⇝ p› |
  ImpI: ‹p # A ⇝ q ⟹ A ⇝ p → q› |
  ImpE: ‹A ⇝ p → q ⟹ A ⇝ p ⟹ A ⇝ q› |
  ImpC: ‹p → - # A ⇝ p ⟹ A ⇝ p›

abbreviation *natural-deduction* (‹⊢ -› [*100*] *100*) **where** ‹⊢ p ≡ [] ⇝ p›

theorem *soundness*: ‹A ⇝ p ⟹ ∀ p ∈ set A. I ⊨ p ⟹ I ⊨ p›
  ⟨proof⟩

lemma *weaken'*: ‹A ⇝ p ⟹ set A = set B ⟹ B ⇝ p›
⟨proof⟩

lemma *weak*: ‹A ⇝ p ⟹ q # A ⇝ p›
⟨proof⟩

lemma *add-assumptions*: ‹⊢ p ⟹ A ⇝ p›
  ⟨proof⟩

lemma *weaken*: ‹$A \leadsto p \Longrightarrow set\ A \subseteq set\ B \Longrightarrow B \leadsto p$›
⟨*proof*⟩

lemma *deduct*: ‹$A \leadsto p \to q \Longrightarrow p \# A \leadsto q$›
⟨*proof*⟩

lemma *Peirce*: ‹$\vdash ((p \to q) \to p) \to p$›
⟨*proof*⟩

lemma *Simp*: ‹$\vdash p \to q \to p$›
⟨*proof*⟩

lemma *Tran*: ‹$\vdash (p \to q) \to (q \to r) \to p \to r$›
⟨*proof*⟩

lemma *Swap*: ‹$\vdash (p \to q \to r) \to q \to p \to r$›
⟨*proof*⟩

lemma *Tran'*: ‹$\vdash (q \to r) \to (p \to q) \to p \to r$›
⟨*proof*⟩

lemma *Imp1*: ‹$\vdash (q \to s) \to ((q \to r) \to s) \to s$›
⟨*proof*⟩

lemma *Imp2*: ‹$\vdash ((r \to s) \to s) \to ((q \to r) \to s) \to s$›
⟨*proof*⟩

lemma *Imp3*: ‹$\vdash ((q \to s) \to s) \to (r \to s) \to (q \to r) \to s$›
⟨*proof*⟩

fun *pros* **where**
‹$pros\ (p \to q) = remdups\ (pros\ p\ @\ pros\ q)$› |
‹$pros\ p = (case\ p\ of\ \cdot n \Rightarrow [n]\ |\ \text{-} \Rightarrow [])$›

lemma *distinct-pros*: ‹$distinct\ (pros\ p)$›
⟨*proof*⟩

abbreviation ‹$lift\ t\ s\ p \equiv if\ t\ then\ (p \to s) \to s\ else\ p \to s$›

abbreviation ‹$lifts\ I\ s \equiv map\ (\lambda n.\ lift\ (I\ n)\ s\ (\cdot n))$›

lemma *lifts-weaken*: ‹$lifts\ I\ s\ l \leadsto p \Longrightarrow set\ l \subseteq set\ l' \Longrightarrow lifts\ I\ s\ l' \leadsto p$›
⟨*proof*⟩

lemma *lifts-pros-lift*: ‹$lifts\ I\ s\ (pros\ p) \leadsto lift\ (I \models p)\ s\ p$›
⟨*proof*⟩

lemma *lifts-pros*: ‹$I \models p \Longrightarrow lifts\ I\ p\ (pros\ p) \leadsto p$›
⟨*proof*⟩

theorem *completeness*: ‹$\forall I.\ I \models p \Longrightarrow \vdash p$›
⟨*proof*⟩

primrec *chain* **where**
‹$chain\ p\ []\ = p$› |
‹$chain\ p\ (q\ \#\ A) = q \to chain\ p\ A$›

lemma *chain-rev*: ‹$B \rightsquigarrow chain\ p\ A \Longrightarrow rev\ A\ @\ B \rightsquigarrow p$›
⟨*proof*⟩

lemma *chain-deduct*: ‹$\vdash chain\ p\ A \Longrightarrow A \rightsquigarrow p$›
⟨*proof*⟩

lemma *chain-semantics*: ‹$I \models chain\ p\ A = ((\forall p \in set\ A.\ I \models p) \longrightarrow I \models p)$›
⟨*proof*⟩

theorem *main*: ‹$A \rightsquigarrow p = (\forall I.\ (\forall p \in set\ A.\ I \models p) \longrightarrow I \models p)$›
⟨*proof*⟩

end