

Soundness and Completeness of Implicational Logic

Asta Halkjær From Jørgen Villadsen

April 18, 2024

Contents

1 Formalization of the Bernays-Tarski Axiom System for Classical Implicational Logic	1
1.1 Syntax, Semantics and Axiom System	1
1.2 Soundness and Derived Formulas	2
1.3 Completeness and Main Theorem	2
1.4 Reference	5
2 Formalization of Łukasiewicz’s Axiom System from 1924 for Classical Propositional Logic	6
2.1 Syntax, Semantics and Axiom System	6
2.2 Soundness and Derived Formulas	6
2.3 Completeness and Main Theorem	9
2.4 Reference	12

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 $\langle \models \rangle$ 50) where

$\langle I \models \cdot n = I n \rangle \mid$
 $\langle I \models p \rightarrow q = (I \models p \longrightarrow I \models q) \rangle$

inductive Ax ($\langle \vdash \rightarrow \rangle$ 50) where

Simp: $\langle \vdash p \rightarrow q \rightarrow p \rangle \mid$
Tran: $\langle \vdash (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow p \rightarrow r \rangle \mid$
MP: $\langle \vdash p \rightarrow q \Longrightarrow \vdash p \Longrightarrow \vdash q \rangle \mid$
PR: $\langle \vdash (p \rightarrow q) \rightarrow p \Longrightarrow \vdash p \rangle$

1.2 Soundness and Derived Formulas

theorem soundness: $\langle \vdash p \Longrightarrow I \models p \rangle$

by (*induct p rule: Ax.induct*) *auto*

lemma Swap: $\langle \vdash (p \rightarrow q \rightarrow r) \rightarrow q \rightarrow p \rightarrow r \rangle$

proof –

have $\langle \vdash q \rightarrow (q \rightarrow r) \rightarrow r \rangle$
using *MP PR Simp Tran* **by** *metis*
then show *?thesis*
using *MP Tran* **by** *meson*

qed

lemma Peirce: $\langle \vdash ((p \rightarrow q) \rightarrow p) \rightarrow p \rangle$

using *MP PR Simp Swap Tran* **by** *meson*

lemma Hilbert: $\langle \vdash (p \rightarrow p \rightarrow q) \rightarrow p \rightarrow q \rangle$

using *MP MP Tran Tran Peirce* .

lemma Id: $\langle \vdash p \rightarrow p \rangle$

using *MP Hilbert Simp* .

lemma Tran': $\langle \vdash (q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r \rangle$

using *MP Swap Tran* .

lemma Frege: $\langle \vdash (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r \rangle$

using *MP MP Tran MP MP Tran Swap Tran' MP Tran' Hilbert* .

lemma Imp1: $\langle \vdash (q \rightarrow s) \rightarrow ((q \rightarrow r) \rightarrow s) \rightarrow s \rangle$

using *MP Peirce Tran Tran'* **by** *meson*

lemma Imp2: $\langle \vdash ((r \rightarrow s) \rightarrow s) \rightarrow ((q \rightarrow r) \rightarrow s) \rightarrow s \rangle$

using *MP Tran MP Tran Simp* .

lemma Imp3: $\langle \vdash ((q \rightarrow s) \rightarrow s) \rightarrow (r \rightarrow s) \rightarrow (q \rightarrow r) \rightarrow s \rangle$

using *MP Swap Tran* **by** *meson*

1.3 Completeness and Main Theorem

fun *pros* **where**

⟨*pros* ($p \rightarrow q$) = *remdups* (*pros* p @ *pros* q)⟩ |
 ⟨*pros* p = (*case* p of (\cdot n) \Rightarrow [n] | $- \Rightarrow$ [])⟩

lemma *distinct-pros*: ⟨*distinct* (*pros* p)⟩
by (*induct* p) *simp-all*

primrec *imply* (**infixr** \rightsquigarrow 56) **where**

⟨[] \rightsquigarrow $q = q$ ⟩ |
 ⟨ $p \# ps \rightsquigarrow q = p \rightarrow ps \rightsquigarrow q$ ⟩

lemma *imply-append*: ⟨ ps @ $qs \rightsquigarrow r = ps \rightsquigarrow qs \rightsquigarrow r$ ⟩
by (*induct* ps) *simp-all*

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

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

proof –

assume $\langle ps \vdash q \rangle$
with *MP Simp* **have** $\langle \vdash p \rightarrow ps \rightsquigarrow q \rangle$.
then show *?thesis*
by *simp*

qed

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

by (*induct* ps) (*use* *MP Frege Simp imply.simps in metis*)+

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

by (*induct* ps) (*use* *imply-Cons imply-head in auto*)

lemma *imply-MP*: ⟨ $\vdash ps \rightsquigarrow (p \rightarrow q) \rightarrow ps \rightsquigarrow p \rightarrow ps \rightsquigarrow q$ ⟩

proof (*induct* ps)

case (*Cons* $r\ ps$)

then have $\langle \vdash (r \rightarrow ps \rightsquigarrow (p \rightarrow q)) \rightarrow (r \rightarrow ps \rightsquigarrow p) \rightarrow r \rightarrow ps \rightsquigarrow q \rangle$

using *MP Frege Simp* **by** *meson*

then show *?case*

by *simp*

qed (*auto intro: Id*)

lemma *MP'*: ⟨ $ps \vdash p \rightarrow q \Longrightarrow ps \vdash p \Longrightarrow ps \vdash q$ ⟩

using *MP imply-MP* **by** *metis*

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

by (*induct* qs *arbitrary: ps*) (*simp, metis MP' imply-append imply-Cons imply-head imply.simps(2)*)

lemma *imply-deduct*: ⟨ $p \# ps \vdash q \Longrightarrow ps \vdash p \rightarrow q$ ⟩

using *imply-append imply-swap-append imply.simps* **by** *metis*

lemma *add-imp* [*simp*]: $\langle \vdash p \implies ps \vdash p \rangle$

proof –

note *MP*

moreover have $\langle \vdash p \rightarrow ps \rightsquigarrow p \rangle$

using *imply-head* **by** *simp*

moreover assume $\langle \vdash p \rangle$

ultimately show *?thesis* .

qed

lemma *imply-weaken*: $\langle ps \vdash p \implies \text{set } ps \subseteq \text{set } ps' \implies ps' \vdash p \rangle$

by (*induct ps arbitrary: p*) (*simp, metis MP' imply-deduct imply-mem insert-subset list.set(2)*)

abbreviation $\langle \text{lift } t \text{ s } p \equiv \text{if } t \text{ then } (p \rightarrow s) \rightarrow s \text{ else } p \rightarrow s \rangle$

abbreviation $\langle \text{lifts } I \text{ s } \equiv \text{map } (\lambda n. \text{lift } (I \ n) \text{ s } (\cdot \ n)) \rangle$

lemma *lifts-weaken*: $\langle \text{lifts } I \text{ s } l \vdash p \implies \text{set } l \subseteq \text{set } l' \implies \text{lifts } I \text{ s } l' \vdash p \rangle$

using *imply-weaken* **by** (*metis (no-types, lifting) image-mono set-map*)

lemma *lifts-pros-lift*: $\langle \text{lifts } I \text{ s } (\text{pros } p) \vdash \text{lift } (I \models p) \text{ s } p \rangle$

proof (*induct p*)

case (*Imp q r*)

consider $\langle \neg I \models q \mid \langle I \models r \rangle \mid \langle I \models q \rangle \langle \neg I \models r \rangle$

by *blast*

then show *?case*

proof *cases*

case 1

then have $\langle \text{lifts } I \text{ s } (\text{pros } (q \rightarrow r)) \vdash q \rightarrow s \rangle$

using *Imp(1) lifts-weaken* [**where** $l' = \langle \text{pros } (q \rightarrow r) \rangle$] **by** *simp*

then have $\langle \text{lifts } I \text{ s } (\text{pros } (q \rightarrow r)) \vdash ((q \rightarrow r) \rightarrow s) \rightarrow s \rangle$

using *Imp1 MP' add-imp* **by** *blast*

with 1 **show** *?thesis*

by *simp*

next

case 2

then have $\langle \text{lifts } I \text{ s } (\text{pros } (q \rightarrow r)) \vdash (r \rightarrow s) \rightarrow s \rangle$

using *Imp(2) lifts-weaken* [**where** $l' = \langle \text{pros } (q \rightarrow r) \rangle$] **by** *simp*

then have $\langle \text{lifts } I \text{ s } (\text{pros } (q \rightarrow r)) \vdash ((q \rightarrow r) \rightarrow s) \rightarrow s \rangle$

using *Imp2 MP' add-imp* **by** *blast*

with 2 **show** *?thesis*

by *simp*

next

case 3

then have $\langle \text{lifts } I \text{ s } (\text{pros } (q \rightarrow r)) \vdash (q \rightarrow s) \rightarrow s \rangle \langle \text{lifts } I \text{ s } (\text{pros } (q \rightarrow r)) \vdash r \rightarrow s \rangle$

using *Imp lifts-weaken* [**where** $l' = \langle \text{pros } (q \rightarrow r) \rangle$] **by** *simp-all*

then have $\langle \text{lifts } I \text{ s } (\text{pros } (q \rightarrow r)) \vdash (q \rightarrow r) \rightarrow s \rangle$

```

    using Imp3 MP' add-imply by blast
  with 3 show ?thesis
    by simp
qed
qed (auto intro: Id Ax.intros)

lemma lifts-pros:  $\langle I \models p \implies \text{lifts } I p (\text{pros } p) \vdash p \rangle$ 
proof -
  assume  $\langle I \models p \rangle$ 
  then have  $\langle \text{lifts } I p (\text{pros } p) \vdash (p \rightarrow p) \rightarrow p \rangle$ 
    using lifts-pros-lift[of I p p] by simp
  then show ?thesis
    using Id MP' add-imply by blast
qed

theorem completeness:  $\langle \forall I. I \models p \implies \vdash p \rangle$ 
proof -
  let ?A =  $\langle \lambda l I. \text{lifts } I p l \vdash p \rangle$ 
  let ?B =  $\langle \lambda l. \forall I. ?A l I \wedge \text{distinct } l \rangle$ 
  assume  $\langle \forall I. I \models p \rangle$ 
  moreover have  $\langle ?B l \implies (\bigwedge n l. ?B (n \# l) \implies ?B l) \implies ?B [] \rangle$  for l
    by (induct l) blast+
  moreover have  $\langle ?B (n \# l) \implies ?B l \rangle$  for n l
  proof -
    assume *:  $\langle ?B (n \# l) \rangle$ 
    show  $\langle ?B l \rangle$ 
  proof
    fix I
    from * have  $\langle ?A (n \# l) (I(n := \text{True})) \rangle \langle ?A (n \# l) (I(n := \text{False})) \rangle$ 
      by blast+
    moreover from * have  $\langle \forall m \in \text{set } l. \forall t. (I(n := t)) m = I m \rangle$ 
      by simp
    ultimately have  $\langle ((\cdot n \rightarrow p) \rightarrow p) \# \text{lifts } I p l \vdash p \rangle \langle (\cdot n \rightarrow p) \# \text{lifts } I p l$ 
      by (simp-all cong: map-cong)
    then have  $\langle ?A l I \rangle$ 
      using MP' imply-deduct by blast
    moreover from * have  $\langle \text{distinct } (n \# l) \rangle$ 
      by blast
    ultimately show  $\langle ?A l I \wedge \text{distinct } l \rangle$ 
      by simp
  qed
qed
qed
ultimately have  $\langle ?B [] \rangle$ 
  using lifts-pros distinct-pros by blast
then show ?thesis
  by simp
qed

```

theorem main: $\langle \vdash p = (\forall I. I \models p) \rangle$
 using *soundness completeness by blast*

1.4 Reference

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

end

2 Formalization of Łukasiewicz'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 $\langle \cdot \rangle$ |
Neg form $\langle \sim \rangle$ |
Imp form form (**infix** $\langle \rightarrow \rangle$ 55)

primrec *semantics* (**infix** $\langle \models \rangle$ 50) **where**

$\langle I \models \cdot \ n = I \ n \rangle$ |
 $\langle I \models \sim \ p = (\neg I \models p) \rangle$ |
 $\langle I \models p \rightarrow q = (I \models p \longrightarrow I \models q) \rangle$

inductive *Ax* $\langle \vdash \rightarrow 50 \rangle$ **where**

01: $\langle \vdash (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow p \rightarrow r \rangle$ |
02: $\langle \vdash (\sim p \rightarrow p) \rightarrow p \rangle$ |
03: $\langle \vdash p \rightarrow \sim p \rightarrow q \rangle$ |
MP: $\langle \vdash p \rightarrow q \Longrightarrow \vdash p \Longrightarrow \vdash q \rangle$

2.2 Soundness and Derived Formulas

theorem *soundness:* $\langle \vdash p \Longrightarrow I \models p \rangle$
 by (*induct p rule: Ax.induct*) *simp-all*

lemma *04:* $\langle \vdash (((q \rightarrow r) \rightarrow p \rightarrow r) \rightarrow s) \rightarrow (p \rightarrow q) \rightarrow s \rangle$
 using *MP 01 01* .

lemma *05:* $\langle \vdash (p \rightarrow q \rightarrow r) \rightarrow (s \rightarrow q) \rightarrow p \rightarrow s \rightarrow r \rangle$
 using *MP 04 04* .

lemma *06:* $\langle \vdash (p \rightarrow q) \rightarrow ((p \rightarrow r) \rightarrow s) \rightarrow (q \rightarrow r) \rightarrow s \rangle$
 using *MP 04 01* .

lemma *07:* $\langle \vdash (t \rightarrow (p \rightarrow r) \rightarrow s) \rightarrow (p \rightarrow q) \rightarrow t \rightarrow (q \rightarrow r) \rightarrow s \rangle$
 using *MP 05 06* .

lemma 09: $\langle \vdash ((\sim p \rightarrow q) \rightarrow r) \rightarrow p \rightarrow r \rangle$
using *MP 01 03* .

lemma 10: $\langle \vdash p \rightarrow ((\sim p \rightarrow p) \rightarrow p) \rightarrow (q \rightarrow p) \rightarrow p \rangle$
using *MP 09 06* .

lemma 11: $\langle \vdash (q \rightarrow (\sim p \rightarrow p) \rightarrow p) \rightarrow (\sim p \rightarrow p) \rightarrow p \rangle$
using *MP MP 10 02 02* .

lemma 12: $\langle \vdash t \rightarrow (\sim p \rightarrow p) \rightarrow p \rangle$
using *MP 09 11* .

lemma 13: $\langle \vdash (\sim p \rightarrow q) \rightarrow t \rightarrow (q \rightarrow p) \rightarrow p \rangle$
using *MP 07 12* .

lemma 14: $\langle \vdash ((t \rightarrow (q \rightarrow p) \rightarrow p) \rightarrow r) \rightarrow (\sim p \rightarrow q) \rightarrow r \rangle$
using *MP 01 13* .

lemma 15: $\langle \vdash (\sim p \rightarrow q) \rightarrow (q \rightarrow p) \rightarrow p \rangle$
using *MP 14 02* .

lemma 16: $\langle \vdash p \rightarrow p \rangle$
using *MP 09 02* .

lemma 17: $\langle \vdash p \rightarrow (q \rightarrow p) \rightarrow p \rangle$
using *MP 09 15* .

lemma 18: $\langle \vdash q \rightarrow p \rightarrow q \rangle$
using *MP MP 05 17 03* .

lemma 19: $\langle \vdash ((p \rightarrow q) \rightarrow r) \rightarrow q \rightarrow r \rangle$
using *MP 01 18* .

lemma 20: $\langle \vdash p \rightarrow (p \rightarrow q) \rightarrow q \rangle$
using *MP 19 15* .

lemma 21: $\langle \vdash (p \rightarrow q \rightarrow r) \rightarrow q \rightarrow p \rightarrow r \rangle$
using *MP 05 20* .

lemma 22: $\langle \vdash (q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r \rangle$
using *MP 21 01* .

lemma 23: $\langle \vdash ((q \rightarrow p \rightarrow r) \rightarrow s) \rightarrow (p \rightarrow q \rightarrow r) \rightarrow s \rangle$
using *MP 01 21* .

lemma 24: $\langle \vdash ((p \rightarrow q) \rightarrow p) \rightarrow p \rangle$
using *MP MP 23 15 03* .

lemma 25: $\langle \vdash ((p \rightarrow r) \rightarrow s) \rightarrow (p \rightarrow q) \rightarrow (q \rightarrow r) \rightarrow s \rangle$
using *MP 21 06* .

lemma 26: $\langle \vdash ((p \rightarrow q) \rightarrow r) \rightarrow (r \rightarrow p) \rightarrow p \rangle$
using *MP 25 24* .

lemma 28: $\langle \vdash (((r \rightarrow p) \rightarrow p) \rightarrow s) \rightarrow ((p \rightarrow q) \rightarrow r) \rightarrow s \rangle$
using *MP 01 26* .

lemma 29: $\langle \vdash ((p \rightarrow q) \rightarrow r) \rightarrow (p \rightarrow r) \rightarrow r \rangle$
using *MP 28 26* .

lemma 31: $\langle \vdash (p \rightarrow s) \rightarrow ((p \rightarrow q) \rightarrow r) \rightarrow (s \rightarrow r) \rightarrow r \rangle$
using *MP 07 29* .

lemma 32: $\langle \vdash ((p \rightarrow q) \rightarrow r) \rightarrow (p \rightarrow s) \rightarrow (s \rightarrow r) \rightarrow r \rangle$
using *MP 21 31* .

lemma 33: $\langle \vdash (p \rightarrow s) \rightarrow (s \rightarrow q \rightarrow p \rightarrow r) \rightarrow q \rightarrow p \rightarrow r \rangle$
using *MP 32 18* .

lemma 34: $\langle \vdash (s \rightarrow q \rightarrow p \rightarrow r) \rightarrow (p \rightarrow s) \rightarrow q \rightarrow p \rightarrow r \rangle$
using *MP 21 33* .

lemma 35: $\langle \vdash (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r \rangle$
using *MP 34 22* .

lemma 36: $\langle \vdash \sim p \rightarrow p \rightarrow q \rangle$
using *MP 21 03* .

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: $\langle \vdash (q \rightarrow s) \rightarrow (\sim q \rightarrow s) \rightarrow s \rangle$
using *MP Clavius' Expl' Frege' Swap by meson*

lemma Neg2: $\langle \vdash ((q \rightarrow s) \rightarrow s) \rightarrow \sim q \rightarrow s \rangle$
using *MP Tran MP Swap Expl* .

lemma *Imp1*: $\langle \vdash (q \rightarrow s) \rightarrow ((q \rightarrow r) \rightarrow s) \rightarrow s \rangle$
using *MP Peirce Tran Tran'* **by** *meson*

lemma *Imp2*: $\langle \vdash ((r \rightarrow s) \rightarrow s) \rightarrow ((q \rightarrow r) \rightarrow s) \rightarrow s \rangle$
using *MP Tran MP Tran Simp* .

lemma *Imp3*: $\langle \vdash ((q \rightarrow s) \rightarrow s) \rightarrow (r \rightarrow s) \rightarrow (q \rightarrow r) \rightarrow s \rangle$
using *MP Swap Tran* **by** *meson*

2.3 Completeness and Main Theorem

primrec *pros* **where**

$\langle \text{pros } (\cdot n) = [n] \rangle |$
 $\langle \text{pros } (\sim p) = \text{pros } p \rangle |$
 $\langle \text{pros } (p \rightarrow q) = \text{remdups } (\text{pros } p @ \text{pros } q) \rangle$

lemma *distinct-pros*: $\langle \text{distinct } (\text{pros } p) \rangle$
by *(induct p) simp-all*

primrec *imply* (**infix** $\langle \rightsquigarrow \rangle$ 56) **where**

$\langle [] \rightsquigarrow q = q \rangle |$
 $\langle p \# ps \rightsquigarrow q = p \rightarrow ps \rightsquigarrow q \rangle$

lemma *imply-append*: $\langle ps @ qs \rightsquigarrow r = ps \rightsquigarrow qs \rightsquigarrow r \rangle$
by *(induct ps) simp-all*

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

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

proof –

assume $\langle ps \vdash q \rangle$
with *MP Simp* **have** $\langle \vdash p \rightarrow ps \rightsquigarrow q \rangle$.
then show *?thesis*
by *simp*

qed

lemma *imply-head*: $\langle p \# ps \vdash p \rangle$
by *(induct ps) (use MP Frege Simp imply.simps in metis)+*

lemma *imply-mem*: $\langle p \in \text{set } ps \implies ps \vdash p \rangle$
by *(induct ps) (use imply-Cons imply-head in auto)*

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

proof *(induct ps)*

case *(Cons r ps)*

then have $\langle \vdash (r \rightarrow ps \rightsquigarrow (p \rightarrow q)) \rightarrow (r \rightarrow ps \rightsquigarrow p) \rightarrow r \rightarrow ps \rightsquigarrow q \rangle$
using *MP Frege Simp* **by** *meson*

then show *?case*

by *simp*
qed (*auto intro: Id*)

lemma *MP'*: $\langle ps \vdash p \rightarrow q \implies ps \vdash p \implies ps \vdash q \rangle$
 using *MP imply-MP by metis*

lemma *imply-swap-append*: $\langle ps @ qs \vdash r \implies qs @ ps \vdash r \rangle$
 by (*induct qs arbitrary: ps*) (*simp, metis MP' imply-append imply-Cons imply-head imply.simps(2)*)

lemma *imply-deduct*: $\langle p \# ps \vdash q \implies ps \vdash p \rightarrow q \rangle$
 using *imply-append imply-swap-append imply.simps by metis*

lemma *add-imply [simp]*: $\langle \vdash p \implies ps \vdash p \rangle$

proof –
 note *MP*
 moreover have $\langle \vdash p \rightarrow ps \rightsquigarrow p \rangle$
 using *imply-head by simp*
 moreover assume $\langle \vdash p \rangle$
 ultimately show *?thesis* .

qed

lemma *imply-weaken*: $\langle ps \vdash p \implies \text{set } ps \subseteq \text{set } ps' \implies ps' \vdash p \rangle$
 by (*induct ps arbitrary: p*) (*simp, metis MP' imply-deduct imply-mem insert-subset list.set(2)*)

abbreviation $\langle \text{lift } t \text{ s } p \equiv \text{if } t \text{ then } (p \rightarrow s) \rightarrow s \text{ else } p \rightarrow s \rangle$

abbreviation $\langle \text{lifts } I \text{ s } \equiv \text{map } (\lambda n. \text{lift } (I \ n) \text{ s } (\cdot \ n)) \rangle$

lemma *lifts-weaken*: $\langle \text{lifts } I \text{ s } l \vdash p \implies \text{set } l \subseteq \text{set } l' \implies \text{lifts } I \text{ s } l' \vdash p \rangle$
 using *imply-weaken by (metis (no-types, lifting) image-mono set-map)*

lemma *lifts-pros-lift*: $\langle \text{lifts } I \text{ s } (\text{pros } p) \vdash \text{lift } (I \models p) \text{ s } p \rangle$

proof (*induct p*)
 case (*Neg q*)
 consider $\langle \neg I \models q \mid \langle I \models q \rangle$
 by *blast*
 then show *?case*
proof *cases*
 case 1
 then have $\langle \text{lifts } I \text{ s } (\text{pros } (\sim q)) \vdash q \rightarrow s \rangle$
 using *Neg by simp*
 then have $\langle \text{lifts } I \text{ s } (\text{pros } (\sim q)) \vdash (\sim q \rightarrow s) \rightarrow s \rangle$
 using *MP' Neg1 add-imply by blast*
 with 1 show *?thesis*
 by *simp*
 next
 case 2

```

then have ⟨lifts I s (pros (∼ q)) ⊢ (q → s) → s⟩
  using Neg by simp
then have ⟨lifts I s (pros (∼ q)) ⊢ ∼ q → s⟩
  using MP' Neg2 add-impby by blast
with 2 show ?thesis
  by simp
qed
next
case (Imp q r)
consider ⟨¬ I ⊢ q⟩ | ⟨I ⊢ r⟩ | ⟨I ⊢ q⟩ ⟨¬ I ⊢ r⟩
  by blast
then show ?case
proof cases
case 1
then have ⟨lifts I s (pros (q → r)) ⊢ q → s⟩
  using Imp(1) lifts-weaken[where l' = ⟨pros (q → r)⟩] by simp
then have ⟨lifts I s (pros (q → r)) ⊢ ((q → r) → s) → s⟩
  using Imp1 MP' add-impby by blast
with 1 show ?thesis
  by simp
next
case 2
then have ⟨lifts I s (pros (q → r)) ⊢ (r → s) → s⟩
  using Imp(2) lifts-weaken[where l' = ⟨pros (q → r)⟩] by simp
then have ⟨lifts I s (pros (q → r)) ⊢ ((q → r) → s) → s⟩
  using Imp2 MP' add-impby by blast
with 2 show ?thesis
  by simp
next
case 3
then have ⟨lifts I s (pros (q → r)) ⊢ (q → s) → s⟩ ⟨lifts I s (pros (q → r)) ⊢
r → s⟩
  using Imp lifts-weaken[where l' = ⟨pros (q → r)⟩] by simp-all
then have ⟨lifts I s (pros (q → r)) ⊢ (q → r) → s⟩
  using Imp3 MP' add-impby by blast
with 3 show ?thesis
  by simp
qed
qed (auto intro: Id)

lemma lifts-pros: ⟨I ⊢ p ⟹ lifts I p (pros p) ⊢ p⟩
proof –
assume ⟨I ⊢ p⟩
then have ⟨lifts I p (pros p) ⊢ (p → p) → p⟩
  using lifts-pros-lift[of I p p] by simp
then show ?thesis
  using Id MP' add-impby by blast
qed

```

```

theorem completeness:  $\langle \forall I. I \models p \implies \vdash p \rangle$ 
proof –
  let  $?A = \langle \lambda l I. \text{lifts } I p l \vdash p \rangle$ 
  let  $?B = \langle \lambda l. \forall I. ?A l I \wedge \text{distinct } l \rangle$ 
  assume  $\langle \forall I. I \models p \rangle$ 
  moreover have  $\langle ?B l \implies (\bigwedge n l. ?B (n \# l) \implies ?B l) \implies ?B [] \rangle$  for  $l$ 
    by (induct l) blast+
  moreover have  $\langle ?B (n \# l) \implies ?B l \rangle$  for  $n l$ 
  proof –
    assume  $*$ :  $\langle ?B (n \# l) \rangle$ 
    show  $\langle ?B l \rangle$ 
    proof
      fix  $I$ 
      from  $*$  have  $\langle ?A (n \# l) (I(n := True)) \rangle \langle ?A (n \# l) (I(n := False)) \rangle$ 
        by blast+
      moreover from  $*$  have  $\langle \forall m \in \text{set } l. \forall t. (I(n := t)) m = I m \rangle$ 
        by simp
      ultimately have  $\langle ((\cdot n \rightarrow p) \rightarrow p) \# \text{lifts } I p l \vdash p \rangle \langle (\cdot n \rightarrow p) \# \text{lifts } I p l$ 
         $\vdash p \rangle$ 
        by (simp-all cong: map-cong)
      then have  $\langle ?A l I \rangle$ 
        using MP' imply-deduct by blast
      moreover from  $*$  have  $\langle \text{distinct } (n \# l) \rangle$ 
        by blast
      ultimately show  $\langle ?A l I \wedge \text{distinct } l \rangle$ 
        by simp
    qed
  qed
  ultimately have  $\langle ?B [] \rangle$ 
    using lifts-pros distinct-pros by blast
  then show  $?thesis$ 
    by simp
qed

theorem main:  $\langle (\vdash p) = (\forall I. I \models p) \rangle$ 
  using soundness completeness by blast

```

2.4 Reference

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

end

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

datatype *form* =

Pro nat $(\langle \cdot \rangle)$ |
Imp form form (**infix** $\langle \rightarrow \rangle$ 100)

primrec *semantics* (**infix** $\langle \models \rangle$ 50) **where**

$\langle I \models \cdot n = I n \rangle \mid$
 $\langle I \models p \rightarrow q = (I \models p \longrightarrow I \models q) \rangle$

abbreviation $sc \ (\langle \llbracket - \rrbracket \rangle)$ **where** $\langle \llbracket I \rrbracket X Y \equiv (\forall p \in set X. I \models p) \longrightarrow (\exists q \in set Y. I \models q) \rangle$

inductive SC (**infix** $\langle \gg \rangle$ 50) **where**

Imp-L: $\langle p \rightarrow q \# X \gg Y \rangle$ **if** $\langle X \gg p \# Y \rangle$ **and** $\langle q \# X \gg Y \rangle \mid$
Imp-R: $\langle X \gg p \rightarrow q \# Y \rangle$ **if** $\langle p \# X \gg q \# Y \rangle \mid$
Set-L: $\langle X' \gg Y \rangle$ **if** $\langle X \gg Y \rangle$ **and** $\langle set X' = set X \rangle \mid$
Set-R: $\langle X \gg Y' \rangle$ **if** $\langle X \gg Y \rangle$ **and** $\langle set Y' = set Y \rangle \mid$
Basic: $\langle p \# - \gg p \# - \rangle$

function mp **where**

$\langle mp A B (p \rightarrow q \# C) [] = (mp A B C [p] \wedge mp A B (q \# C) []) \rangle \mid$
 $\langle mp A B C (p \rightarrow q \# D) = mp A B (p \# C) (q \# D) \rangle \mid$
 $\langle mp A B [] [] = (set A \cap set B \neq \{\}) \rangle \mid$
 $\langle mp A B (\cdot n \# C) [] = mp (n \# A) B C [] \rangle \mid$
 $\langle mp A B C (\cdot n \# D) = mp A (n \# B) C D \rangle$
by *pat-completeness simp-all*

termination

by (*relation* $\langle measure \ (\lambda(-, -, C, D). 2 * (\sum p \leftarrow C @ D. size p) + size (C @ D)) \rangle$) *simp-all*

lemma *main*: $\langle (\forall I. \llbracket I \rrbracket (map \cdot A @ C) (map \cdot B @ D)) \longleftrightarrow mp A B C D \rangle$

by (*induct rule: mp.induct*) (*auto 5 2*)

definition *prover* ($\langle \vdash \rangle$) **where** $\langle \vdash p \equiv mp [] [] [p] \rangle$

theorem *prover-correct*: $\langle \vdash p \longleftrightarrow (\forall I. I \models p) \rangle$

unfolding *prover-def* **by** (*simp flip: main*)

export-code \vdash **in** *SML*

lemma *MP*: $\langle mp A B C D \Longrightarrow set X \supseteq set (map \cdot A @ C) \Longrightarrow set Y \supseteq set (map \cdot B @ D) \Longrightarrow X \gg Y \rangle$

proof (*induct A B C D arbitrary: X Y rule: mp.induct[case-names Imp-L Imp-R Basic]*)

case (*Imp-L A B p q C*)

have

$\langle set (map \cdot A @ C) \subseteq set X \rangle$

$\langle set (map \cdot B) \subseteq set Y \rangle$

using *Imp-L(4,5)* **by** *auto*

moreover from this have

$\langle set (map \cdot A @ C) \subseteq set (q \# X) \rangle$

$\langle set (map \cdot B) \subseteq set (p \# Y) \rangle$

by *auto*

ultimately have $\langle p \rightarrow q \# X \gg Y \rangle$

using *Imp-L(1-3) SC.Imp-L* **by** *simp*
then show *?case*
using *Imp-L(4) Set-L* **by** *fastforce*
next
case (*Imp-R A B C p q D*)
have
 $\langle \text{set } (\text{map } \cdot A @ C) \subseteq \text{set } (p \# X) \rangle$
 $\langle \text{set } (\text{map } \cdot B @ D) \subseteq \text{set } (q \# Y) \rangle$
using *Imp-R(3,4)* **by** *auto*
then have $\langle X \gg p \rightarrow q \# Y \rangle$
using *Imp-R(1,2) SC.Imp-R* **by** *simp*
then show *?case*
using *Imp-R(4) Set-R* **by** *fastforce*
next
case (*Basic A B*)
obtain *n* **where**
 $\langle n \in \text{set } A \rangle$
 $\langle n \in \text{set } B \rangle$
using *Basic(1)* **by** *auto*
then have
 $\langle \text{set } (\cdot n \# X) = \text{set } X \rangle$
 $\langle \text{set } (\cdot n \# Y) = \text{set } Y \rangle$
using *Basic(2,3)* **by** *auto*
then show *?case*
using *Set-L Set-R SC.Basic* **by** *metis*
qed *simp-all*

theorem *OK*: $\langle (\forall I. \llbracket I \rrbracket X Y) \longleftrightarrow X \gg Y \rangle$
by (*rule, use MP main[of $\langle \llbracket \cdot \rrbracket \rangle$ - $\langle \llbracket \cdot \rrbracket \rangle$ -]* **in** *simp, induct rule: SC.induct*) *auto*

corollary $\langle \llbracket \cdot \rrbracket \gg [p] \longleftrightarrow (\forall I. I \models p) \rangle$
using *OK* **by** *force*

proposition $\langle \llbracket \cdot \rrbracket \gg [p \rightarrow p] \rangle$
proof –
from *Imp-R* **have** *?thesis* **if** $\langle [p] \gg [p] \rangle$
using *that* **by** *force*
with *Basic* **show** *?thesis*
by *force*
qed

proposition $\langle \llbracket \cdot \rrbracket \gg [p \rightarrow (p \rightarrow q) \rightarrow q] \rangle$
proof –
from *Imp-R* **have** *?thesis* **if** $\langle [p] \gg [(p \rightarrow q) \rightarrow q] \rangle$
using *that* **by** *force*
with *Imp-R* **have** *?thesis* **if** $\langle [p \rightarrow q, p] \gg [q] \rangle$
using *that* **by** *force*
with *Imp-L* **have** *?thesis* **if** $\langle [p] \gg [p, q] \rangle$ **and** $\langle [q, p] \gg [q] \rangle$
using *that* **by** *force*

with *Basic show* *?thesis*
 by *force*
 qed

proposition $\langle [] \gg [p \rightarrow q \rightarrow q \rightarrow p] \rangle$
proof –
 from *Imp-R* have *?thesis* if $\langle [p] \gg [q \rightarrow q \rightarrow p] \rangle$
 using *that by force*
 with *Imp-R* have *?thesis* if $\langle [q, p] \gg [q \rightarrow p] \rangle$
 using *that by force*
 with *Imp-R* have *?thesis* if $\langle [q, q, p] \gg [p] \rangle$
 using *that by force*
 with *Set-L* have *?thesis* if $\langle [p, q] \gg [p] \rangle$
 using *that by force*
 with *Basic show* *?thesis*
 by *force*
 qed

proposition $\langle [] \gg [(p \rightarrow q) \rightarrow p \rightarrow q] \rangle$
proof –
 from *Imp-R* have *?thesis* if $\langle [p \rightarrow q] \gg [p \rightarrow q] \rangle$
 using *that by force*
 with *Basic show* *?thesis*
 by *force*
 qed

proposition $\langle [] \gg [p \rightarrow p \rightarrow q \rightarrow q] \rangle$
proof –
 from *Imp-R* have *?thesis* if $\langle [p] \gg [p \rightarrow q \rightarrow q] \rangle$
 using *that by force*
 with *Imp-R* have *?thesis* if $\langle [p, p] \gg [q \rightarrow q] \rangle$
 using *that by force*
 with *Imp-R* have *?thesis* if $\langle [q, p, p] \gg [q] \rangle$
 using *that by force*
 with *Basic show* *?thesis*
 by *force*
 qed

proposition $\langle [] \gg [(p \rightarrow p \rightarrow q) \rightarrow p \rightarrow q] \rangle$
proof –
 from *Imp-R* have *?thesis* if $\langle [p \rightarrow p \rightarrow q] \gg [p \rightarrow q] \rangle$
 using *that by force*
 with *Imp-R* have *?thesis* if $\langle [p, p \rightarrow p \rightarrow q] \gg [q] \rangle$
 using *that by force*
 with *Set-L* have *?thesis* if $\langle [p \rightarrow p \rightarrow q, p] \gg [q] \rangle$
 using *that by force*
 with *Imp-L* have *?thesis* if $\langle [p] \gg [p, q] \rangle$ and $\langle [p \rightarrow q, p] \gg [q] \rangle$
 using *that by force*
 with *Imp-L* have *?thesis* if $\langle [p] \gg [p, q] \rangle$ and $\langle [q, p] \gg [q] \rangle$ and $\langle [p] \gg [p, q] \rangle$

using *that by force*
 with *Basic show ?thesis*
 by *force*
 qed

proposition $\langle [] \gg [p \rightarrow q \rightarrow p] \rangle$
proof –
 from *Imp-R have ?thesis if* $\langle [p] \gg [q \rightarrow p] \rangle$
 using *that by force*
 with *Imp-R have ?thesis if* $\langle [q, p] \gg [p] \rangle$
 using *that by force*
 with *Set-L have ?thesis if* $\langle [p, q] \gg [p] \rangle$
 using *that by force*
 with *Basic show ?thesis*
 by *force*
 qed

proposition $\langle [] \gg [(p \rightarrow r) \rightarrow (r \rightarrow q) \rightarrow p \rightarrow q] \rangle$
proof –
 from *Imp-R have ?thesis if* $\langle [p \rightarrow r] \gg [(r \rightarrow q) \rightarrow p \rightarrow q] \rangle$
 using *that by force*
 with *Imp-R have ?thesis if* $\langle [r \rightarrow q, p \rightarrow r] \gg [p \rightarrow q] \rangle$
 using *that by force*
 with *Imp-R have ?thesis if* $\langle [p, r \rightarrow q, p \rightarrow r] \gg [q] \rangle$
 using *that by force*
 with *Set-L have ?thesis if* $\langle [r \rightarrow q, p \rightarrow r, p] \gg [q] \rangle$
 using *that by force*
 with *Imp-L have ?thesis if* $\langle [p \rightarrow r, p] \gg [r, q] \rangle$ **and** $\langle [q, p \rightarrow r, p] \gg [q] \rangle$
 using *that by force*
 with *Basic have ?thesis if* $\langle [p \rightarrow r, p] \gg [r, q] \rangle$
 using *that by force*
 with *Imp-L have ?thesis if* $\langle [p] \gg [p, r, q] \rangle$ **and** $\langle [r, p] \gg [r, q] \rangle$
 using *that by force*
 with *Basic show ?thesis*
 by *force*
 qed

proposition $\langle [] \gg [((p \rightarrow q) \rightarrow p) \rightarrow p] \rangle$
proof –
 from *Imp-R have ?thesis if* $\langle [(p \rightarrow q) \rightarrow p] \gg [p] \rangle$
 using *that by force*
 with *Imp-L have ?thesis if* $\langle [] \gg [p \rightarrow q, p] \rangle$ **and** $\langle [p] \gg [p] \rangle$
 using *that by force*
 with *Basic have ?thesis if* $\langle [] \gg [p \rightarrow q, p] \rangle$
 using *that by force*
 with *Imp-R have ?thesis if* $\langle [p] \gg [q, p] \rangle$
 using *that by force*
 with *Set-R have ?thesis if* $\langle [p] \gg [p, q] \rangle$
 using *that by force*

with *Basic* **show** *?thesis*
by *force*
qed
end