Admissible Rules in Logic and Algebra

George Metcalfe

Mathematics Institute
University of Bern

Pisa Summer Workshop on Proof Theory

12-15 June 2012, Pisa

Derivability vs Admissibility

Consider a system defined by two rules:

$$Nat(0) \qquad \text{and} \qquad Nat(x) \rhd Nat(s(x)).$$

The following rule is **derivable**:

$$Nat(x) \rhd Nat(s(s(x))).$$

However, this rule is only **admissible**:

$$Nat(s(x)) \rhd Nat(x).$$

But what if we add to the system:

$$Nat(s(-1)) \qquad ???$$

Derivability vs Admissibility

Consider a system defined by two rules:

$$Nat(0) \qquad \text{and} \qquad Nat(x) \rhd Nat(s(x)).$$

The following rule is **derivable**:

$$Nat(x) \rhd Nat(s(s(x))).$$

However, this rule is only **admissible**:

$$Nat(s(x)) \rhd Nat(x).$$

But what if we add to the system:

$$Nat(s(-1)) \qquad ???$$

Derivability vs Admissibility

Consider a system defined by two rules:

$$Nat(0) \qquad \text{and} \qquad Nat(x) \rhd Nat(s(x)).$$

The following rule is **derivable**:

$$Nat(x) \rhd Nat(s(s(x))).$$

However, this rule is only **admissible**:

$$Nat(s(x)) \rhd Nat(x).$$

But what if we add to the system:

$$Nat(s(-1)) \qquad ???$$

Derivability vs Admissibility

Consider a system defined by two rules:

$$Nat(0) \qquad \text{and} \qquad Nat(x) \rhd Nat(s(x)).$$

The following rule is **derivable**:

$$Nat(x) \rhd Nat(s(s(x))).$$

However, this rule is only **admissible**:

$$Nat(s(x)) \rhd Nat(x).$$

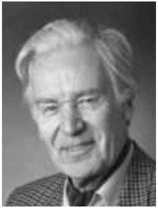
But what if we add to the system:

$$Nat(s(-1)) \qquad ???$$

Background: Lorenzen's Idea

The notion of an **admissible rule** was introduced explicitly by Paul Lorenzen in the 1950s in the context of **intuitionistic propositional logic** IPC.

P. Lorenzen. Einführung in die operative Logik und Mathematik. Springer, 1955.



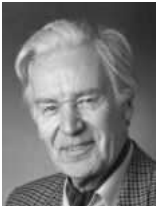
Lorenzen calls a rule R *admissible* in a system S, if adding R to the primitive rules of S does not enlarge the set of theorems.

His "operative interpretation" is that a rule R is admissible in S if every application of R can be eliminated from the extended calculus.

P. Schroeder-Heister. Lorenzen's operative justification of intuitionistic logic. *One Hundred Years of Intuitionism (1907-2007)*. Birkhäuser, 2008.

The notion of an **admissible rule** was introduced explicitly by Paul Lorenzen in the 1950s in the context of **intuitionistic propositional logic** IPC.

P. Lorenzen. Einführung in die operative Logik und Mathematik. Springer, 1955.



Lorenzen calls a rule R *admissible* in a system S, if adding R to the primitive rules of S does not enlarge the set of theorems.

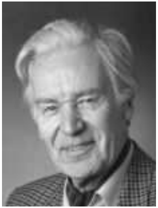
His "operative interpretation" is that a rule R is admissible in S if every application of R can be eliminated from the extended calculus.

P. Schroeder-Heister. Lorenzen's operative justification of intuitionistic logic. *One Hundred Years of Intuitionism (1907-2007)*. Birkhäuser, 2008.

Background: Lorenzen's Idea

The notion of an **admissible rule** was introduced explicitly by Paul Lorenzen in the 1950s in the context of **intuitionistic propositional logic** IPC.

P. Lorenzen. Einführung in die operative Logik und Mathematik. Springer, 1955.



Lorenzen calls a rule R *admissible* in a system S, if adding R to the primitive rules of S does not enlarge the set of theorems.

His "operative interpretation" is that a rule R is admissible in S if every application of R can be eliminated from the extended calculus.

P. Schroeder-Heister. Lorenzen's operative justification of intuitionistic logic. *One Hundred Years of Intuitionism (1907-2007)*. Birkhäuser, 2008.

Background: Admissibility in IPC

Examples of admissible but not derivable rules of IPC include the "independence of premises" rule

$$\{\neg p \to (q \lor r)\} \rhd (\neg p \to q) \lor (\neg p \to r)$$

and the "disjunction property"

$$\{p \lor q\} \rhd \{p, q\}.$$

It was shown by Vladimir Rybakov (among other things) that the set of admissible rules of IPC is decidable but not finitely axiomatizable.

Admissibility of Logical Inference Rules.
V. Rybakov. Elsevier, 1997.

Background: Admissibility in IPC

Examples of admissible but not derivable rules of IPC include the "independence of premises" rule

$$\{\neg p \to (q \lor r)\} \rhd (\neg p \to q) \lor (\neg p \to r)$$

and the "disjunction property"

$$\{p \lor q\} \rhd \{p, q\}.$$

It was shown by Vladimir Rybakov (among other things) that the set of admissible rules of IPC is decidable but not finitely axiomatizable.

Admissibility of Logical Inference Rules.
V. Rybakov. Elsevier, 1997.

Background: Admissibility in IPC

Examples of admissible but not derivable rules of IPC include the "independence of premises" rule

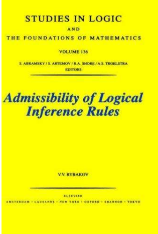
$$\{\neg p \to (q \vee r)\} \rhd (\neg p \to q) \vee (\neg p \to r)$$

and the "disjunction property"

$$\{p \vee q\} \rhd \{p, q\}.$$

It was shown by Vladimir Rybakov (among other things) that the set of admissible rules of IPC is decidable but not finitely axiomatizable.

Admissibility of Logical Inference Rules.
V. Rybakov. Elsevier, 1997.



Iemhoff and Rozière proved independently that the "Visser rules"

$$\{\bigwedge_{i=1}^{n}(p_i \to q_i) \to (p_{n+1} \lor p_{n+2})\} \rhd \bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(p_i \to q_i) \to p_j) \quad n = 2, 3, \ldots$$

plus the disjunction property provide a "basis" for admissibility in IPC.

P. Rozière. Regles admissibles en calcul propositionnel intuitionniste. Ph.D. thesis, Université Paris VII, 1992.

R. Iemhoff. On the admissible rules of intuitionistic propositional logic. *Journal of Symbolic Logic* 66(1):281–294, 2001.

Relationships between unification, admissibility, and projectivity in IPC have been studied and characterized by Ghilardi.

S. Ghilardi. Unification in intuitionistic logic. *Journal of Symbolic Logic* 64(2):859–880, 1999.

Background: The Visser Rules

Iemhoff and Rozière proved independently that the "Visser rules"

$$\{\bigwedge_{i=1}^{n}(p_i \to q_i) \to (p_{n+1} \vee p_{n+2})\} \rhd \bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(p_i \to q_i) \to p_j) \quad n = 2, 3, \ldots$$

plus the disjunction property provide a "basis" for admissibility in IPC.

P. Rozière. Regles admissibles en calcul propositionnel intuitionniste.
Ph.D. thesis, Université Paris VII, 1992.

R. Iemhoff. On the admissible rules of intuitionistic propositional logic.
Journal of Symbolic Logic 66(1):281–294, 2001.

Relationships between unification, admissibility, and projectivity in IPC have been studied and characterized by Ghilardi.

S. Ghilardi. Unification in intuitionistic logic.
Journal of Symbolic Logic 64(2):859–880, 1999.

Background: The Disjunctive Syllogism

In **relevant logics** such as R and RM, the "disjunctive syllogism"

$$\{\neg p,\ p \vee q\} \ \triangleright\ q$$

is admissible but not derivable.

A. R. Anderson and N. D. Belnap. *Entailment*.
Princeton University Press, 1975.

Background: Admissibility in Modal Logics

In the **modal logics** K and $\mathrm{K4}$, the rule

$$\{\Box p\} \ \triangleright \ p$$

is admissible and non-derivable, while Löb's rule

$$\{\Box p \to p\} \ \triangleright \ p$$

is admissible and non-derivable in K, but not admissible in $\mathrm{K4}$.

E. Jeřábek. Admissible Rules of Modal Logics. *Journal of Logic and Computation* 15:411–431, 2005.

In the **modal logics** K and $\mathrm{K4}$, the rule

$$\{\Box p\} \ \triangleright \ p$$

is admissible and non-derivable, while Löb's rule

$$\{\Box p \to p\} \ \triangleright \ p$$

is admissible and non-derivable in K, but not admissible in $\mathrm{K4}$.

E. Jeřábek. Admissible Rules of Modal Logics.
Journal of Logic and Computation 15:411–431, 2005.

Background: Admissibility in Algebra

For classes of algebraic structures, rules correspond to **clauses** (or **quasiequations**) and admissibility to validity in **free algebras**.

For example, the quasiequation

$$\{x + x \approx 0\} \rhd x \approx 0$$

- is not valid in all abelian groups (e.g., $\mathbf{Z_2}$)
- but is valid in all free abelian groups, since

$$\varphi + \varphi \approx 0 \text{ is valid in } \mathbf{Z} \quad \Longrightarrow \quad \varphi \approx 0 \text{ is valid in } \mathbf{Z}.$$

Background: Admissibility in Algebra

For classes of algebraic structures, rules correspond to **clauses** (or **quasiequations**) and admissibility to validity in **free algebras**.

For example, the quasiequation

$$\{x + x \approx 0\} \triangleright x \approx 0$$

- is not valid in all abelian groups (e.g., $\mathbb{Z}_2$)
- but is valid in all free abelian groups, since

$$\varphi + \varphi \approx 0 \text{ is valid in } \mathbb{Z} \quad \Longrightarrow \quad \varphi \approx 0 \text{ is valid in } \mathbb{Z}.$$

Background: Admissibility in Algebra

For classes of algebraic structures, rules correspond to **clauses** (or **quasiequations**) and admissibility to validity in **free algebras**.

For example, the quasiequation

$$\{x + x \approx 0\} \rhd x \approx 0$$

- is not valid in all abelian groups (e.g., $\mathbf{Z_2}$)
- but is valid in all free abelian groups, since

$$\varphi + \varphi \approx 0 \text{ is valid in } \mathbf{Z} \quad \Longrightarrow \quad \varphi \approx 0 \text{ is valid in } \mathbf{Z}.$$

Background: Admissibility in Algebra

For classes of algebraic structures, rules correspond to **clauses** (or **quasiequations**) and admissibility to validity in **free algebras**.

For example, the quasiequation

$$\{x + x \approx 0\} \rhd x \approx 0$$

- is not valid in all abelian groups (e.g., $\mathbf{Z_2}$)
- but is valid in all free abelian groups, since

$$\varphi + \varphi \approx 0 \text{ is valid in } \mathbf{Z} \quad \Longrightarrow \quad \varphi \approx 0 \text{ is valid in } \mathbf{Z}.$$

The following clauses are valid in all **free lattices**, but not all lattices

- Whitman's condition

$$\{x \wedge y \preccurlyeq z \vee w\} \rhd \{x \wedge y \preccurlyeq z,\ x \wedge y \preccurlyeq w,\ x \preccurlyeq z \vee w,\ y \preccurlyeq z \vee w\}.$$

P. Whitman. Free lattices.
Annals of Mathematics 42: 325–329, 1941.

- Jónsson's semi-distributivity property

$$\{x \wedge y \approx x \wedge z\} \rhd x \wedge y \approx x \wedge (y \vee z).$$

B. Jónsson. Sublattices of a free lattice.
Canadian Journal of Mathematics 13: 256–264, 1961.

The following clauses are valid in all **free lattices**, but not all lattices

- Whitman's condition

$$\{x \wedge y \preccurlyeq z \vee w\} \rhd \{x \wedge y \preccurlyeq z,\ x \wedge y \preccurlyeq w,\ x \preccurlyeq z \vee w,\ y \preccurlyeq z \vee w\}.$$

P. Whitman. Free lattices.
Annals of Mathematics 42: 325–329, 1941.

- Jónsson's semi-distributivity property

$$\{x \wedge y \approx x \wedge z\} \rhd x \wedge y \approx x \wedge (y \vee z).$$

B. Jónsson. Sublattices of a free lattice.
Canadian Journal of Mathematics 13: 256–264, 1961.

The following clauses are valid in all **free lattices**, but not all lattices

- Whitman's condition

$$\{x \wedge y \preccurlyeq z \vee w\} \rhd \{x \wedge y \preccurlyeq z,\ x \wedge y \preccurlyeq w,\ x \preccurlyeq z \vee w,\ y \preccurlyeq z \vee w\}.$$

P. Whitman. Free lattices.
Annals of Mathematics 42: 325–329, 1941.

- Jónsson's semi-distributivity property

$$\{x \wedge y \approx x \wedge z\} \rhd x \wedge y \approx x \wedge (y \vee z).$$

B. Jónsson. Sublattices of a free lattice.
Canadian Journal of Mathematics 13: 256–264, 1961.

Background: Admissibility in Proof Theory

In Gentzen's sequent calculus for propositional classical logic, applications of the **cut rule** can be *eliminated* from derivations.

G. Gentzen. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift* 39:176–210,405–431,1935.

Saying that $\{\varphi_1, \ldots, \varphi_n\} \rhd \psi$ is derivable in a sequent calculus when $\frac{\Rightarrow \varphi_1 \ \ldots \ \Rightarrow \varphi_n}{\Rightarrow \psi}$ is derivable, it follows that the **transitivity rule**

$$\{p \to q,\ q \to r\} \ \rhd \ r$$

is admissible but not derivable in the cut-free system.

Background: Admissibility in Proof Theory

In Gentzen's sequent calculus for propositional classical logic, applications of the **cut rule** can be *eliminated* from derivations.

> G. Gentzen. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift* 39:176–210,405–431,1935.

Saying that $\{\varphi_1, \ldots, \varphi_n\} \rhd \psi$ is derivable in a sequent calculus when $\frac{\Rightarrow \varphi_1 \; \ldots \; \Rightarrow \varphi_n}{\Rightarrow \psi}$ is derivable, it follows that the **transitivity rule**

$$\{p \to q,\ q \to r\} \ \rhd\ r$$

is admissible but not derivable in the cut-free system.

Background: Admissibility in Proof Theory

In Gentzen's sequent calculus for propositional classical logic, applications of the **cut rule** can be *eliminated* from derivations.

G. Gentzen. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift* 39:176–210,405–431,1935.

Saying that $\{\varphi_1, \ldots, \varphi_n\} \rhd \psi$ is derivable in a sequent calculus when $\frac{\Rightarrow \varphi_1 \ \ldots \ \Rightarrow \varphi_n}{\Rightarrow \psi}$ is derivable, it follows that the **transitivity rule**

$$\{p \to q,\ q \to r\} \ \rhd \ r$$

is admissible but not derivable in the cut-free system.

This Tutorial. . .

. . . consists of six parts:

(I) Admissibility in Logic

(II) An Algebraic Perspective

(III) Unification and Admissibility

(IV) Proof Theory for Admissible Rules

(V) A First-Order Framework

(VI) Eliminations and Applications.

This Tutorial...

...consists of six parts:

(I) Admissibility in Logic

(II) An Algebraic Perspective

(III) Unification and Admissibility

(IV) Proof Theory for Admissible Rules

(V) A First-Order Framework

(VI) Eliminations and Applications.

. . . consists of six parts:

(I) Admissibility in Logic

(II) An Algebraic Perspective

(III) Unification and Admissibility

(IV) Proof Theory for Admissible Rules

(V) A First-Order Framework

(VI) Eliminations and Applications.

This Tutorial. . .

. . . consists of six parts:

(I) Admissibility in Logic

(II) An Algebraic Perspective

(III) Unification and Admissibility

(IV) Proof Theory for Admissible Rules

(V) A First-Order Framework

(VI) Eliminations and Applications.

This Tutorial. . .

. . . consists of six parts:

(I) Admissibility in Logic

(II) An Algebraic Perspective

(III) Unification and Admissibility

(IV) Proof Theory for Admissible Rules

(V) A First-Order Framework

(VI) Eliminations and Applications.

This Tutorial. . .

. . . consists of six parts:

(I) Admissibility in Logic
(II) An Algebraic Perspective
(III) Unification and Admissibility
(IV) Proof Theory for Admissible Rules
(V) A First-Order Framework
(VI) Eliminations and Applications.

This Tutorial. . .

. . . consists of six parts:

(I) Admissibility in Logic
(II) An Algebraic Perspective
(III) Unification and Admissibility
(IV) Proof Theory for Admissible Rules
(V) A First-Order Framework
(VI) Eliminations and Applications.

Part I

Admissibility in Logic

Rules

We will make use of

- **propositional languages** $\mathcal{L}$ consisting of connectives such as $\wedge, \vee, \cdot, , \to, \neg, \bot, \top$ with specified finite arities
- finite sets (denoted Γ, Δ) of $\mathcal{L}$-**formulas** (denoted ψ, φ, χ) from $\mathrm{Fm}_{\mathcal{L}}$ built from a countably infinite set of variables (denoted p, q, r)
- endomorphisms on $\mathbf{Fm}_{\mathcal{L}}$ called $\mathcal{L}$-**substitutions** (denoted σ).

Definition

An $\mathcal{L}$-**rule** is an ordered pair of finite sets of $\mathcal{L}$-formulas, written

$$\Gamma \rhd \Delta,$$

called **single-conclusion** if $|\Delta| = 1$, **multiple-conclusion** in general.

Rules

We will make use of

- **propositional languages** $\mathcal{L}$ consisting of connectives such as $\wedge, \vee, \cdot, , \to, \neg, \bot, \top$ with specified finite arities
- finite sets (denoted Γ, Δ) of **$\mathcal{L}$-formulas** (denoted ψ, φ, χ) from $\mathrm{Fm}_{\mathcal{L}}$ built from a countably infinite set of variables (denoted p, q, r)
- endomorphisms on $\mathbf{Fm}_{\mathcal{L}}$ called **$\mathcal{L}$-substitutions** (denoted σ).

Definition

An **$\mathcal{L}$-rule** is an ordered pair of finite sets of $\mathcal{L}$-formulas, written

$$\Gamma \rhd \Delta,$$

called **single-conclusion** if $|\Delta| = 1$, **multiple-conclusion** in general.

Rules

We will make use of

- **propositional languages** $\mathcal{L}$ consisting of connectives such as $\wedge, \vee, \cdot, , \to, \neg, \bot, \top$ with specified finite arities
- finite sets (denoted Γ, Δ) of $\mathcal{L}$**-formulas** (denoted ψ, φ, χ) from $\mathrm{Fm}_{\mathcal{L}}$ built from a countably infinite set of variables (denoted p, q, r)
- endomorphisms on $\mathbf{Fm}_{\mathcal{L}}$ called $\mathcal{L}$**-substitutions** (denoted σ).

Definition

An $\mathcal{L}$**-rule** is an ordered pair of finite sets of $\mathcal{L}$-formulas, written

$$\Gamma \rhd \Delta,$$

called **single-conclusion** if $|\Delta| = 1$, **multiple-conclusion** in general.

We will make use of

- **propositional languages** $\mathcal{L}$ consisting of connectives such as $\wedge, \vee, \cdot, , \rightarrow, \neg, \bot, \top$ with specified finite arities
- finite sets (denoted Γ, Δ) of $\mathcal{L}$**-formulas** (denoted ψ, φ, χ) from $\mathrm{Fm}_{\mathcal{L}}$ built from a countably infinite set of variables (denoted p, q, r)
- endomorphisms on **Fm**$_{\mathcal{L}}$ called $\mathcal{L}$**-substitutions** (denoted σ).

Definition

An $\mathcal{L}$**-rule** is an ordered pair of finite sets of $\mathcal{L}$-formulas, written

$$\Gamma \rhd \Delta,$$

called **single-conclusion** if $|\Delta| = 1$, **multiple-conclusion** in general.

Rules

We will make use of

- **propositional languages** $\mathcal{L}$ consisting of connectives such as $\wedge, \vee, \cdot, , \to, \neg, \bot, \top$ with specified finite arities
- finite sets (denoted Γ, Δ) of $\mathcal{L}$**-formulas** (denoted ψ, φ, χ) from $\mathrm{Fm}_{\mathcal{L}}$ built from a countably infinite set of variables (denoted p, q, r)
- endomorphisms on $\mathbf{Fm}_{\mathcal{L}}$ called $\mathcal{L}$**-substitutions** (denoted σ).

Definition

An $\mathcal{L}$**-rule** is an ordered pair of finite sets of $\mathcal{L}$-formulas, written

$$\Gamma \rhd \Delta,$$

called **single-conclusion** if $|\Delta| = 1$, **multiple-conclusion** in general.

Definition

A **logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of single-conclusion $\mathcal{L}$-rules satisfying (writing $\Gamma \vdash_{\mathrm{L}} \Delta$ for $(\Gamma, \Delta) \in \mathrm{L}$):

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \varphi$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \psi$, then $\Gamma \vdash_{\mathrm{L}} \psi$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any $\mathcal{L}$-substitution σ (structurality).

An L-**theorem** is a formula φ such that $\emptyset \vdash_{\mathrm{L}} \varphi$ (abbreviated as $\vdash_{\mathrm{L}} \varphi$).

(Note: A **finitary structural consequence relation** is obtained by fixing for $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}_{\mathcal{L}}$: $\Gamma \vdash_{\mathrm{L}} \varphi$ iff $\Gamma' \vdash_{\mathrm{L}} \varphi$ for some finite $\Gamma' \subseteq \Gamma$.)

Definition

A **logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of single-conclusion $\mathcal{L}$-rules satisfying (writing $\Gamma \vdash_{\mathrm{L}} \Delta$ for $(\Gamma, \Delta) \in \mathrm{L}$):

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \varphi$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \psi$, then $\Gamma \vdash_{\mathrm{L}} \psi$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any $\mathcal{L}$-substitution σ (structurality).

An L-**theorem** is a formula φ such that $\emptyset \vdash_{\mathrm{L}} \varphi$ (abbreviated as $\vdash_{\mathrm{L}} \varphi$).

(Note: A **finitary structural consequence relation** is obtained by fixing for $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}_{\mathcal{L}}$: $\Gamma \vdash_{\mathrm{L}} \varphi$ iff $\Gamma' \vdash_{\mathrm{L}} \varphi$ for some finite $\Gamma' \subseteq \Gamma$.)

Logics and Consequence

Definition

A **logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of single-conclusion $\mathcal{L}$-rules satisfying (writing $\Gamma \vdash_{\mathrm{L}} \Delta$ for $(\Gamma, \Delta) \in \mathrm{L}$):

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \varphi$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \psi$, then $\Gamma \vdash_{\mathrm{L}} \psi$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any $\mathcal{L}$-substitution σ (structurality).

An L-**theorem** is a formula φ such that $\emptyset \vdash_{\mathrm{L}} \varphi$ (abbreviated as $\vdash_{\mathrm{L}} \varphi$).

(Note: A **finitary structural consequence relation** is obtained by fixing for $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}_{\mathcal{L}}$: $\Gamma \vdash_{\mathrm{L}} \varphi$ iff $\Gamma' \vdash_{\mathrm{L}} \varphi$ for some finite $\Gamma' \subseteq \Gamma$.)

Logics and Consequence

Definition

A **logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of single-conclusion $\mathcal{L}$-rules satisfying (writing $\Gamma \vdash_{\mathrm{L}} \Delta$ for $(\Gamma, \Delta) \in \mathrm{L}$):

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \varphi$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \psi$, then $\Gamma \vdash_{\mathrm{L}} \psi$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any $\mathcal{L}$-substitution σ (structurality).

An L-**theorem** is a formula φ such that $\emptyset \vdash_{\mathrm{L}} \varphi$ (abbreviated as $\vdash_{\mathrm{L}} \varphi$).

(Note: A **finitary structural consequence relation** is obtained by fixing for $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}_{\mathcal{L}}$: $\Gamma \vdash_{\mathrm{L}} \varphi$ iff $\Gamma' \vdash_{\mathrm{L}} \varphi$ for some finite $\Gamma' \subseteq \Gamma$.)

Logics and Consequence

Definition

A **logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of single-conclusion $\mathcal{L}$-rules satisfying (writing $\Gamma \vdash_{\mathrm{L}} \Delta$ for $(\Gamma, \Delta) \in \mathrm{L}$):

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \varphi$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \psi$, then $\Gamma \vdash_{\mathrm{L}} \psi$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any $\mathcal{L}$-substitution σ (structurality).

An L-**theorem** is a formula φ such that $\emptyset \vdash_{\mathrm{L}} \varphi$ (abbreviated as $\vdash_{\mathrm{L}} \varphi$).

(Note: A **finitary structural consequence relation** is obtained by fixing for $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}_{\mathcal{L}}$: $\Gamma \vdash_{\mathrm{L}} \varphi$ iff $\Gamma' \vdash_{\mathrm{L}} \varphi$ for some finite $\Gamma' \subseteq \Gamma$.)

Logics and Consequence

Definition

A **logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of single-conclusion $\mathcal{L}$-rules satisfying (writing $\Gamma \vdash_{\mathrm{L}} \Delta$ for $(\Gamma, \Delta) \in \mathrm{L}$):

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \varphi$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \psi$, then $\Gamma \vdash_{\mathrm{L}} \psi$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any $\mathcal{L}$-substitution σ (structurality).

An L-**theorem** is a formula φ such that $\emptyset \vdash_{\mathrm{L}} \varphi$ (abbreviated as $\vdash_{\mathrm{L}} \varphi$).

(Note: A **finitary structural consequence relation** is obtained by fixing for $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}_{\mathcal{L}}$: $\Gamma \vdash_{\mathrm{L}} \varphi$ iff $\Gamma' \vdash_{\mathrm{L}} \varphi$ for some finite $\Gamma' \subseteq \Gamma$.)

Definition

A **logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of single-conclusion $\mathcal{L}$-rules satisfying (writing $\Gamma \vdash_{\mathrm{L}} \Delta$ for $(\Gamma, \Delta) \in \mathrm{L}$):

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \varphi$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \psi$, then $\Gamma \vdash_{\mathrm{L}} \psi$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \varphi$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any $\mathcal{L}$-substitution σ (structurality).

An L-**theorem** is a formula φ such that $\emptyset \vdash_{\mathrm{L}} \varphi$ (abbreviated as $\vdash_{\mathrm{L}} \varphi$).

(Note: A **finitary structural consequence relation** is obtained by fixing for $\Gamma \cup \{\varphi\} \subseteq \mathrm{Fm}_{\mathcal{L}}$: $\Gamma \vdash_{\mathrm{L}} \varphi$ iff $\Gamma' \vdash_{\mathrm{L}} \varphi$ for some finite $\Gamma' \subseteq \Gamma$.)

Multiple-Conclusion Consequence

Definition

An **m-logic** L on $\mathbf{Fm}_{\mathcal{L}}$ is a set of $\mathcal{L}$-rules satisfying:

- $\{\varphi\} \vdash_{\mathrm{L}} \varphi$ (reflexivity)
- if $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \cup \Gamma' \vdash_{\mathrm{L}} \Delta' \cup \Delta$ (monotonicity)
- if $\Gamma \vdash_{\mathrm{L}} \{\varphi\} \cup \Delta$ and $\Gamma \cup \{\varphi\} \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \Delta$ (transitivity)
- if $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\Delta$ for any $\mathcal{L}$-substitution σ (structurality).

Derivable and Admissible Rules

Definition

For a logic L on $\mathbf{Fm}_{\mathcal{L}}$, an $\mathcal{L}$-rule $\Gamma \rhd \Delta$ is

- L-**derivable**, written $\Gamma \vdash_{\mathrm{L}} \Delta$, if $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$.
- L-**admissible**, written $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, if for every $\mathcal{L}$-substitution σ:

$$\vdash_{\mathrm{L}} \sigma\varphi \quad \text{for all } \varphi \in \Gamma \qquad \Longrightarrow \qquad \vdash_{\mathrm{L}} \sigma\psi \quad \text{for some } \psi \in \Delta.$$

(Note: $\vdash_{\mathrm{L}}$ and $\mathrel{|\!\!\sim}_{\mathrm{L}}$ are m-logics.)

Derivable and Admissible Rules

Definition

For a logic L on $\mathbf{Fm}_{\mathcal{L}}$, an $\mathcal{L}$-rule $\Gamma \rhd \Delta$ is

- L**-derivable**, written $\Gamma \vdash_{\mathrm{L}} \Delta$, if $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$.
- L**-admissible**, written $\Gamma \mid\!\sim_{\mathrm{L}} \Delta$, if for every $\mathcal{L}$-substitution σ:

$$\vdash_{\mathrm{L}} \sigma\varphi \ \text{ for all } \varphi \in \Gamma \quad \Longrightarrow \quad \vdash_{\mathrm{L}} \sigma\psi \ \text{ for some } \psi \in \Delta.$$

(Note: $\vdash_{\mathrm{L}}$ and $\mid\!\sim_{\mathrm{L}}$ are m-logics.)

Derivable and Admissible Rules

Definition

For a logic L on $\mathbf{Fm}_{\mathcal{L}}$, an $\mathcal{L}$-rule $\Gamma \rhd \Delta$ is

- L-**derivable**, written $\Gamma \vdash_{\mathrm{L}} \Delta$, if $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$.
- L-**admissible**, written $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, if for every $\mathcal{L}$-substitution σ:

$$\vdash_{\mathrm{L}} \sigma\varphi \text{ for all } \varphi \in \Gamma \quad \Longrightarrow \quad \vdash_{\mathrm{L}} \sigma\psi \text{ for some } \psi \in \Delta.$$

(Note: $\vdash_{\mathrm{L}}$ and $\mathrel{|\!\!\sim}_{\mathrm{L}}$ are m-logics.)

Derivable and Admissible Rules

Definition

For a logic L on $\mathbf{Fm}_{\mathcal{L}}$, an $\mathcal{L}$-rule $\Gamma \rhd \Delta$ is

- L-**derivable**, written $\Gamma \vdash_{\mathrm{L}} \Delta$, if $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$.
- L-**admissible**, written $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, if for every $\mathcal{L}$-substitution σ:

$$\vdash_{\mathrm{L}} \sigma\varphi \ \text{ for all } \varphi \in \Gamma \quad \Longrightarrow \quad \vdash_{\mathrm{L}} \sigma\psi \ \text{ for some } \psi \in \Delta.$$

(Note: $\vdash_{\mathrm{L}}$ and $\mathrel{|\!\!\sim}_{\mathrm{L}}$ are m-logics.)

Example: Łukasiewicz Logic

Łukasiewicz logic Ł with negation $\neg$ and implication $\to$ can be defined via an axiom system or, semantically, via a matrix with truth values $[0, 1]$, designated truth value 1, and interpretations:

$$\neg x = 1 - x \qquad \text{and} \qquad x \to y = \min(1, 1 - x + y).$$

Then for example:

$$\{p \to \neg p, \ \neg p \to p\} \not\vdash_{Ł} q, \qquad \text{but} \quad \{p \to \neg p, \ \neg p \to p\} \mathrel{|\!\sim}_{Ł} q.$$

Also, defining $\varphi \cdot \psi = \neg(\varphi \to \neg\psi)$ (so that $x \cdot y = \max(0, x + y - 1)$):

$$\{p \to (p \cdot p)\} \not\vdash_{Ł} \{p, \neg p\}, \qquad \text{but} \quad \{p \to (p \cdot p)\} \mathrel{|\!\sim}_{Ł} \{p, \neg p\}.$$

Example: Łukasiewicz Logic

Łukasiewicz logic Ł with negation $\neg$ and implication $\to$ can be defined via an axiom system or, semantically, via a matrix with truth values $[0,1]$, designated truth value 1, and interpretations:

$$\neg x = 1 - x \qquad \text{and} \qquad x \to y = \min(1, 1 - x + y).$$

Then for example:

$$\{p \to \neg p,\ \neg p \to p\} \not\vdash_{Ł} q, \qquad \text{but} \qquad \{p \to \neg p,\ \neg p \to p\} \mathrel{|\!\!\sim}_{Ł} q.$$

Also, defining $\varphi \cdot \psi = \neg(\varphi \to \neg\psi)$ (so that $x \cdot y = \max(0, x + y - 1)$):

$$\{p \to (p \cdot p)\} \not\vdash_{Ł} \{p, \neg p\}, \qquad \text{but} \qquad \{p \to (p \cdot p)\} \mathrel{|\!\!\sim}_{Ł} \{p, \neg p\}.$$

Example: Łukasiewicz Logic

Łukasiewicz logic Ł with negation $\neg$ and implication $\to$ can be defined via an axiom system or, semantically, via a matrix with truth values $[0,1]$, designated truth value 1, and interpretations:

$$\neg x = 1 - x \qquad \text{and} \qquad x \to y = \min(1, 1 - x + y).$$

Then for example:

$$\{p \to \neg p,\ \neg p \to p\} \not\vdash_{\text{Ł}} q, \qquad \text{but} \quad \{p \to \neg p,\ \neg p \to p\} \mathrel{|\!\!\sim}_{\text{Ł}} q.$$

Also, defining $\varphi \cdot \psi = \neg(\varphi \to \neg\psi)$ (so that $x \cdot y = \max(0, x + y - 1)$):

$$\{p \to (p \cdot p)\} \not\vdash_{\text{Ł}} \{p, \neg p\}, \qquad \text{but} \quad \{p \to (p \cdot p)\} \mathrel{|\!\!\sim}_{\text{Ł}} \{p, \neg p\}.$$

Example: Łukasiewicz Logic

Łukasiewicz logic Ł with negation $\neg$ and implication $\to$ can be defined via an axiom system or, semantically, via a matrix with truth values $[0,1]$, designated truth value 1, and interpretations:

$$\neg x = 1 - x \qquad \text{and} \qquad x \to y = \min(1, 1 - x + y).$$

Then for example:

$$\{p \to \neg p,\ \neg p \to p\} \nvdash_{Ł} q, \qquad \text{but} \quad \{p \to \neg p,\ \neg p \to p\} \mathrel{|\!\!\sim}_{Ł} q.$$

Also, defining $\varphi \cdot \psi = \neg(\varphi \to \neg\psi)$ (so that $x \cdot y = \max(0, x + y - 1)$):

$$\{p \to (p \cdot p)\} \nvdash_{Ł} \{p, \neg p\}, \qquad \text{but} \quad \{p \to (p \cdot p)\} \mathrel{|\!\!\sim}_{Ł} \{p, \neg p\}.$$

Example: Łukasiewicz Logic

Łukasiewicz logic Ł with negation $\neg$ and implication $\to$ can be defined via an axiom system or, semantically, via a matrix with truth values $[0, 1]$, designated truth value 1, and interpretations:

$$\neg x = 1 - x \qquad \text{and} \qquad x \to y = \min(1, 1 - x + y).$$

Then for example:

$$\{p \to \neg p,\ \neg p \to p\} \not\vdash_{\text{Ł}} q, \qquad \text{but} \quad \{p \to \neg p,\ \neg p \to p\} \mathrel{|\!\!\sim}_{\text{Ł}} q.$$

Also, defining $\varphi \cdot \psi = \neg(\varphi \to \neg\psi)$ (so that $x \cdot y = \max(0, x + y - 1)$):

$$\{p \to (p \cdot p)\} \not\vdash_{\text{Ł}} \{p, \neg p\}, \qquad \text{but} \quad \{p \to (p \cdot p)\} \mathrel{|\!\!\sim}_{\text{Ł}} \{p, \neg p\}.$$

Another Characterization of Admissibility (1)

Lemma

The following are equivalent for any logic L *on* $\mathbf{Fm}_{\mathcal{L}}$:

(1) $\Gamma \triangleright \varphi$ *is* L*-admissible.*

(2) *For the smallest logic* L^* *on* $\mathbf{Fm}_{\mathcal{L}}$ *containing* L *and* $(\Gamma, \{\varphi\})$:

$$\vdash_{\mathrm{L}} \psi \quad \text{iff} \quad \vdash_{\mathrm{L}^*} \psi.$$

Another Characterization of Admissibility (1)

Lemma

The following are equivalent for any logic L *on* $\mathbf{Fm}_{\mathcal{L}}$*:*

(1) $\Gamma \rhd \varphi$ *is* L*-admissible.*

(2) *For the smallest logic* L^* *on* $\mathbf{Fm}_{\mathcal{L}}$ *containing* L *and* $(\Gamma, \{\varphi\})$*:*

$$\vdash_{\mathrm{L}} \psi \qquad \text{iff} \qquad \vdash_{\mathrm{L}^*} \psi.$$

Another Characterization of Admissibility (2)

Lemma

The following are equivalent for any logic L *on* $\mathbf{Fm}_{\mathcal{L}}$*:*

(1) $\Gamma \rhd \Delta$ *is* L*-admissible.*

(2) *For the smallest m-logic* L^* *on* $\mathbf{Fm}_{\mathcal{L}}$ *containing* L *and* (Γ, Δ)*:*

$$\vdash_{\mathrm{L}} \{\psi_1, \ldots, \psi_n\} \quad \text{iff} \quad \vdash_{\mathrm{L}^*} \{\psi_1, \ldots, \psi_n\}.$$

Another Characterization of Admissibility (2)

Lemma

The following are equivalent for any logic L *on* $\mathbf{Fm}_{\mathcal{L}}$*:*

(1) $\Gamma \rhd \Delta$ *is* L*-admissible.*

(2) *For the smallest m-logic* L^* *on* $\mathbf{Fm}_{\mathcal{L}}$ *containing* L *and* (Γ, Δ)*:*

$$\vdash_{\mathrm{L}} \{\psi_1, \ldots, \psi_n\} \qquad \text{iff} \qquad \vdash_{\mathrm{L}^*} \{\psi_1, \ldots, \psi_n\}.$$

Definition

A logic L on $\mathbf{Fm}_{\mathcal{L}}$ is

- **structurally complete** if for all single-conclusion $\mathcal{L}$-rules $\Gamma \rhd \varphi$

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \Leftrightarrow \quad \Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \varphi$$

  (or, any logic extending L has new theorems)

- **universally complete** if for all $\mathcal{L}$-rules $\Gamma \rhd \Delta$

$$\Gamma \vdash_{\mathrm{L}} \Delta \quad \Leftrightarrow \quad \Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$$

  (or, any m-logic L' extending L has new consequences $\vdash_{\mathrm{L}'} \Delta$).

W. A. Pogorzelski. Structural completeness of the propositional calculus. *Bulletin de L'Académie Polonaise des Sciences* 19: 349–351 (1971).

Structural and Universal Completeness

Definition

A logic L on $\mathbf{Fm}_{\mathcal{L}}$ is

- **structurally complete** if for all single-conclusion $\mathcal{L}$-rules $\Gamma \rhd \varphi$

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \Leftrightarrow \quad \Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \varphi$$

  (or, any logic extending L has new theorems)

- **universally complete** if for all $\mathcal{L}$-rules $\Gamma \rhd \Delta$

$$\Gamma \vdash_{\mathrm{L}} \Delta \quad \Leftrightarrow \quad \Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$$

  (or, any m-logic L' extending L has new consequences $\vdash_{\mathrm{L}'} \Delta$).

W. A. Pogorzelski. Structural completeness of the propositional calculus. *Bulletin de L'Académie Polonaise des Sciences* 19: 349–351 (1971).

Structural and Universal Completeness

Definition

A logic L on $\mathbf{Fm}_{\mathcal{L}}$ is

- **structurally complete** if for all single-conclusion $\mathcal{L}$-rules $\Gamma \rhd \varphi$

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \Leftrightarrow \quad \Gamma \mathrel{|\!\sim}_{\mathrm{L}} \varphi$$

  (or, any logic extending L has new theorems)

- **universally complete** if for all $\mathcal{L}$-rules $\Gamma \rhd \Delta$

$$\Gamma \vdash_{\mathrm{L}} \Delta \quad \Leftrightarrow \quad \Gamma \mathrel{|\!\sim}_{\mathrm{L}} \Delta$$

  (or, any m-logic L' extending L has new consequences $\vdash_{\mathrm{L}'} \Delta$).

W. A. Pogorzelski. Structural completeness of the propositional calculus. *Bulletin de L'Académie Polonaise des Sciences* 19: 349–351 (1971).

Classical Logic is Structurally Complete

Theorem (Pogorzelski)

Classical propositional logic CPC *is structurally complete.*

Proof.

Suppose that $\Gamma \nvdash_{\mathrm{CPC}} \varphi$. Then for some classical evaluation e, we have $e(\psi) = 1$ for all $\psi \in \Gamma$ and $e(\varphi) = 0$. Let:

$$\sigma p = \begin{cases} \top & \text{if } e(p) = 1 \\ \bot & \text{if } e(p) = 0. \end{cases}$$

It follows inductively that $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \top$ iff $e(\psi) = 1$, and $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \bot$ iff $e(\psi) = 1$. So $\vdash_{\mathrm{CPC}} \sigma\psi$ for all $\psi \in \Gamma$, but $\nvdash_{\mathrm{CPC}} \sigma\varphi$. I.e., $\Gamma \not\mid\!\sim_{\mathrm{CPC}} \varphi$. $\square$

Theorem (Pogorzelski)

Classical propositional logic CPC *is structurally complete.*

Proof.

Suppose that $\Gamma \not\vdash_{\mathrm{CPC}} \varphi$. Then for some classical evaluation e, we have $e(\psi) = 1$ for all $\psi \in \Gamma$ and $e(\varphi) = 0$. Let:

$$\sigma p = \begin{cases} \top & \text{if } e(p) = 1 \\ \bot & \text{if } e(p) = 0. \end{cases}$$

It follows inductively that $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \top$ iff $e(\psi) = 1$, and $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \bot$ iff $e(\psi) = 1$. So $\vdash_{\mathrm{CPC}} \sigma\psi$ for all $\psi \in \Gamma$, but $\not\vdash_{\mathrm{CPC}} \sigma\varphi$. I.e., $\Gamma \not\mid\!\sim_{\mathrm{CPC}} \varphi$. $\square$

Classical Logic is Structurally Complete

Theorem (Pogorzelski)

Classical propositional logic CPC *is structurally complete.*

Proof.

Suppose that $\Gamma \nvdash_{\mathrm{CPC}} \varphi$. Then for some classical evaluation e, we have $e(\psi) = 1$ for all $\psi \in \Gamma$ and $e(\varphi) = 0$. Let:

$$\sigma p = \begin{cases} \top & \text{if } e(p) = 1 \\ \bot & \text{if } e(p) = 0. \end{cases}$$

It follows inductively that $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \top$ iff $e(\psi) = 1$, and $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \bot$ iff $e(\psi) = 1$. So $\vdash_{\mathrm{CPC}} \sigma\psi$ for all $\psi \in \Gamma$, but $\nvdash_{\mathrm{CPC}} \sigma\varphi$. I.e., $\Gamma \not\mid\!\sim_{\mathrm{CPC}} \varphi$. □

Classical Logic is Structurally Complete

Theorem (Pogorzelski)

Classical propositional logic CPC *is structurally complete.*

Proof.

Suppose that $\Gamma \nvdash_{\mathrm{CPC}} \varphi$. Then for some classical evaluation e, we have $e(\psi) = 1$ for all $\psi \in \Gamma$ and $e(\varphi) = 0$. Let:

$$\sigma p = \begin{cases} \top & \text{if } e(p) = 1 \\ \bot & \text{if } e(p) = 0. \end{cases}$$

It follows inductively that $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \top$ iff $e(\psi) = 1$, and $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \bot$ iff $e(\psi) = 1$. So $\vdash_{\mathrm{CPC}} \sigma\psi$ for all $\psi \in \Gamma$, but $\nvdash_{\mathrm{CPC}} \sigma\varphi$. I.e., $\Gamma \not\mid\!\sim_{\mathrm{CPC}} \varphi$. □

Classical Logic is Structurally Complete

Theorem (Pogorzelski)

Classical propositional logic CPC *is structurally complete.*

Proof.

Suppose that $\Gamma \not\vdash_{\mathrm{CPC}} \varphi$. Then for some classical evaluation e, we have $e(\psi) = 1$ for all $\psi \in \Gamma$ and $e(\varphi) = 0$. Let:

$$\sigma p = \begin{cases} \top & \text{if } e(p) = 1 \\ \bot & \text{if } e(p) = 0. \end{cases}$$

It follows inductively that $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \top$ iff $e(\psi) = 1$, and $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \bot$ iff $e(\psi) = 1$. So $\vdash_{\mathrm{CPC}} \sigma\psi$ for all $\psi \in \Gamma$, but $\not\vdash_{\mathrm{CPC}} \sigma\varphi$. I.e., $\Gamma \not\mathrel{|\!\sim}_{\mathrm{CPC}} \varphi$. □

Classical Logic is Structurally Complete

Theorem (Pogorzelski)

Classical propositional logic CPC *is structurally complete.*

Proof.

Suppose that $\Gamma \not\vdash_{\mathrm{CPC}} \varphi$. Then for some classical evaluation e, we have $e(\psi) = 1$ for all $\psi \in \Gamma$ and $e(\varphi) = 0$. Let:

$$\sigma p = \begin{cases} \top & \text{if } e(p) = 1 \\ \bot & \text{if } e(p) = 0. \end{cases}$$

It follows inductively that $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \top$ iff $e(\psi) = 1$, and $\vdash_{\mathrm{CPC}} \sigma\psi \leftrightarrow \bot$ iff $e(\psi) = 1$. So $\vdash_{\mathrm{CPC}} \sigma\psi$ for all $\psi \in \Gamma$, but $\not\vdash_{\mathrm{CPC}} \sigma\varphi$. I.e., $\Gamma \not\mid\!\sim_{\mathrm{CPC}} \varphi$. $\square$

Definition

An L-**unifier** of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is an $\mathcal{L}$-substitution σ such that

$$\vdash_{\mathrm{L}} \sigma\varphi \quad \text{for all } \varphi \in \Gamma.$$

In this case, Γ is said to be L-**unifiable**.

Notice that for a non-trivial logic L:

$$\Gamma \text{ is L-unifiable} \qquad \text{iff} \qquad \Gamma \rhd \emptyset \text{ is not L-admissible.}$$

Definition

An L-**unifier** of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is an $\mathcal{L}$-substitution σ such that

$$\vdash_{\mathrm{L}} \sigma\varphi \qquad \text{for all } \varphi \in \Gamma.$$

In this case, Γ is said to be L-**unifiable**.

Notice that for a non-trivial logic L:

$$\Gamma \text{ is L-unifiable} \qquad \text{iff} \qquad \Gamma \rhd \emptyset \;\text{ is not L-admissible.}$$

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L**-exact**.

Lemma

If Γ is L-exact, then $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. $\square$

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L-**exact**.

Lemma

If Γ is L-exact, then $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. $\square$

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L**-exact**.

Lemma

If Γ is L-exact, then $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. $\square$

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L-**exact**.

Lemma

If Γ is L-exact, then $\Gamma \mathrel{|\!\!\!\sim}_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mathrel{|\!\!\!\sim}_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mathrel{|\!\!\!\sim}_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. $\square$

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L-**exact**.

Lemma

If Γ is L-exact, then $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. □

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L-**exact**.

Lemma

If Γ is L-exact, then $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. □

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L**-exact**.

Lemma

If Γ is L-exact, then $\Gamma \mid\!\sim_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mid\!\sim_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mid\!\sim_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. □

Exact Unifiers

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **exact** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \varphi \quad \text{iff} \quad \vdash_{\mathrm{L}} \sigma\varphi.$$

In this case, Γ is said to be L**-exact**.

Lemma

If Γ is L-exact, then $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$ iff $\Gamma \vdash_{\mathrm{L}} \Delta$.

Proof.

($\Leftarrow$) If $\Gamma \vdash_{\mathrm{L}} \Delta$, then $\Gamma \vdash_{\mathrm{L}} \varphi$ for some $\varphi \in \Delta$. So $\sigma\Gamma \vdash_{\mathrm{L}} \sigma\varphi$ for any substitution σ, and if $\vdash_{\mathrm{L}} \sigma\psi$ for each $\psi \in \Gamma$, then $\vdash_{\mathrm{L}} \sigma\varphi$. I.e., $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$.

($\Rightarrow$) Let σ be an exact L-unifier of Γ. If $\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$, then $\vdash_{\mathrm{L}} \sigma\varphi$ for some $\varphi \in \Delta$. So $\Gamma \vdash_{\mathrm{L}} \varphi$ and $\Gamma \vdash_{\mathrm{L}} \Delta$ as required. $\square$

Suppose that L has an implication $\to$ satisfying $\{p, p \to q\} \vdash_{\mathrm{L}} q$.

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **projective** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \sigma\varphi \to \varphi \quad \text{and} \quad \Gamma \vdash_{\mathrm{L}} \varphi \to \sigma\varphi.$$

In this case, Γ is said to be L-**projective**.

Notice that every L-projective unifier of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is

- L-exact
- a **most general** L**-unifier** of Γ
- an L'-projective unifier of Γ for each logic L' extending L.

Projective Unifiers

Suppose that L has an implication $\to$ satisfying $\{p, p \to q\} \vdash_{\mathrm{L}} q$.

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **projective** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \sigma\varphi \to \varphi \quad \text{and} \quad \Gamma \vdash_{\mathrm{L}} \varphi \to \sigma\varphi.$$

In this case, Γ is said to be L**-projective**.

Notice that every L-projective unifier of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is

- L-exact
- a **most general** L**-unifier** of Γ
- an L'-projective unifier of Γ for each logic L' extending L.

Projective Unifiers

Suppose that L has an implication $\to$ satisfying $\{p, p \to q\} \vdash_{\mathrm{L}} q$.

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **projective** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \sigma\varphi \to \varphi \quad \text{and} \quad \Gamma \vdash_{\mathrm{L}} \varphi \to \sigma\varphi.$$

In this case, Γ is said to be L**-projective**.

Notice that every L-projective unifier of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is

- L-exact
- a **most general** L**-unifier** of Γ
- an L'-projective unifier of Γ for each logic L' extending L.

Projective Unifiers

Suppose that L has an implication $\to$ satisfying $\{p, p \to q\} \vdash_{\mathrm{L}} q$.

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **projective** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \sigma\varphi \to \varphi \quad \text{and} \quad \Gamma \vdash_{\mathrm{L}} \varphi \to \sigma\varphi.$$

In this case, Γ is said to be L-**projective**.

Notice that every L-projective unifier of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is

- L-exact
- a **most general** L**-unifier** of Γ
- an L'-projective unifier of Γ for each logic L' extending L.

Projective Unifiers

Suppose that L has an implication $\to$ satisfying $\{p, p \to q\} \vdash_{\mathrm{L}} q$.

Definition

An L-unifier σ of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is called **projective** if for all $\varphi \in \mathrm{Fm}_{\mathcal{L}}$:

$$\Gamma \vdash_{\mathrm{L}} \sigma\varphi \to \varphi \quad \text{and} \quad \Gamma \vdash_{\mathrm{L}} \varphi \to \sigma\varphi.$$

In this case, Γ is said to be L-**projective**.

Notice that every L-projective unifier of $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ is

- L-exact
- a **most general** L**-unifier** of Γ
- an L'-projective unifier of Γ for each logic L' extending L.

Definition

A logic L on $\mathbf{Fm}_{\mathcal{L}}$ is called

- **hereditarily structurally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is structurally complete.
- **hereditarily universally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is universally complete.

Notice that if Γ is L-projective, then for any logic L' extending L:

$$\Gamma \mathrel{|\!\sim}_{\mathrm{L}'} \Delta \quad \text{iff} \quad \Gamma \vdash_{\mathrm{L}'} \Delta.$$

In particular, if *all* finite $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ are

- L-*exact*, then L is universally complete
- L-*projective*, then L is hereditarily universally complete.

Hereditary Structural Completeness

Definition

A logic L on $\mathbf{Fm}_{\mathcal{L}}$ is called

- **hereditarily structurally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is structurally complete.
- **hereditarily universally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is universally complete.

Notice that if Γ is L-projective, then for any logic L' extending L:

$$\Gamma \mid\!\sim_{\mathrm{L}'} \Delta \qquad \text{iff} \qquad \Gamma \vdash_{\mathrm{L}'} \Delta.$$

In particular, if *all* finite $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ are

- L-*exact*, then L is universally complete
- L-*projective*, then L is hereditarily universally complete.

Definition

A logic L on $\mathbf{Fm}_{\mathcal{L}}$ is called

- **hereditarily structurally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is structurally complete.
- **hereditarily universally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is universally complete.

Notice that if Γ is L-projective, then for any logic L' extending L:

$$\Gamma \mathrel{|\!\!\sim}_{\mathrm{L}'} \Delta \qquad \text{iff} \qquad \Gamma \vdash_{\mathrm{L}'} \Delta.$$

In particular, if *all* finite $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ are

- L-*exact*, then L is universally complete
- L-*projective*, then L is hereditarily universally complete.

Hereditary Structural Completeness

Definition

A logic L on $\mathbf{Fm}_{\mathcal{L}}$ is called

- **hereditarily structurally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is structurally complete.
- **hereditarily universally complete** if each logic on $\mathbf{Fm}_{\mathcal{L}}$ extending L is universally complete.

Notice that if Γ is L-projective, then for any logic L' extending L:

$$\Gamma \mid\!\sim_{\mathrm{L}'} \Delta \qquad \text{iff} \qquad \Gamma \vdash_{\mathrm{L}'} \Delta.$$

In particular, if *all* finite $\Gamma \subseteq \mathrm{Fm}_{\mathcal{L}}$ are

- L-*exact*, then L is universally complete
- L-*projective*, then L is hereditarily universally complete.

Prucnal's Trick

For each **implication-conjunction** formula φ of IPC, define

$$\sigma(p) = \varphi \to p \quad \text{for each variable } p.$$

Then inductively, for each implication-conjunction formula ψ:

$$\vdash_{\mathrm{IPC}} \sigma\psi \to (\varphi \to \psi) \quad \text{and} \quad \vdash_{\mathrm{IPC}} (\varphi \to \psi) \to \sigma\psi.$$

But then $\vdash_{\mathrm{IPC}} \sigma\varphi$ and, using properties of IPC:

$$\varphi \vdash_{\mathrm{IPC}} \sigma\psi \to \psi \quad \text{and} \quad \varphi \vdash_{\mathrm{IPC}} \psi \to \sigma\psi.$$

So σ is a **projective** $\mathrm{IPC}_{\neg,\to}$**-unifier** of φ.

Theorem (Prucnal)

The $\{\to, \land\}$ fragment of IPC is hereditarily universally complete.

T. Prucnal. On the structural completeness of some pure implicational propositional calculi. *Studia Logica* 32(1): 45–50, 1973.

Prucnal's Trick

For each **implication-conjunction** formula φ of IPC, define

$$\sigma(p) = \varphi \to p \quad \text{for each variable } p.$$

Then inductively, for each implication-conjunction formula ψ:

$$\vdash_{\mathrm{IPC}} \sigma\psi \to (\varphi \to \psi) \quad \text{and} \quad \vdash_{\mathrm{IPC}} (\varphi \to \psi) \to \sigma\psi.$$

But then $\vdash_{\mathrm{IPC}} \sigma\varphi$ and, using properties of IPC:

$$\varphi \vdash_{\mathrm{IPC}} \sigma\psi \to \psi \quad \text{and} \quad \varphi \vdash_{\mathrm{IPC}} \psi \to \sigma\psi.$$

So σ is a **projective** $\mathrm{IPC}_{\neg,\to}$**-unifier** of φ.

Theorem (Prucnal)

The $\{\to, \wedge\}$ fragment of IPC is hereditarily universally complete.

T. Prucnal. On the structural completeness of some pure implicational propositional calculi. *Studia Logica* 32(1): 45–50, 1973.

Prucnal's Trick

For each **implication-conjunction** formula φ of IPC, define

$$\sigma(p) = \varphi \to p \quad \text{for each variable } p.$$

Then inductively, for each implication-conjunction formula ψ:

$$\vdash_{\mathrm{IPC}} \sigma\psi \to (\varphi \to \psi) \quad \text{and} \quad \vdash_{\mathrm{IPC}} (\varphi \to \psi) \to \sigma\psi.$$

But then $\vdash_{\mathrm{IPC}} \sigma\varphi$ and, using properties of IPC:

$$\varphi \vdash_{\mathrm{IPC}} \sigma\psi \to \psi \quad \text{and} \quad \varphi \vdash_{\mathrm{IPC}} \psi \to \sigma\psi.$$

So σ is a **projective** $\mathrm{IPC}_{\neg,\to}$**-unifier** of φ.

Theorem (Prucnal)

The $\{\to, \wedge\}$ fragment of IPC is hereditarily universally complete.

T. Prucnal. On the structural completeness of some pure implicational propositional calculi. *Studia Logica* 32(1): 45–50, 1973.

Prucnal's Trick

For each **implication-conjunction** formula φ of IPC, define

$$\sigma(p) = \varphi \to p \quad \text{for each variable } p.$$

Then inductively, for each implication-conjunction formula ψ:

$$\vdash_{\mathrm{IPC}} \sigma\psi \to (\varphi \to \psi) \quad \text{and} \quad \vdash_{\mathrm{IPC}} (\varphi \to \psi) \to \sigma\psi.$$

But then $\vdash_{\mathrm{IPC}} \sigma\varphi$ and, using properties of IPC:

$$\varphi \vdash_{\mathrm{IPC}} \sigma\psi \to \psi \quad \text{and} \quad \varphi \vdash_{\mathrm{IPC}} \psi \to \sigma\psi.$$

So σ is a **projective** $\mathrm{IPC}_{\neg,\to}$**-unifier** of φ.

Theorem (Prucnal)

The $\{\to, \wedge\}$ fragment of IPC is hereditarily universally complete.

T. Prucnal. On the structural completeness of some pure implicational propositional calculi. *Studia Logica* 32(1): 45–50, 1973.

Prucnal's Trick

For each **implication-conjunction** formula φ of IPC, define

$$\sigma(p) = \varphi \to p \quad \text{for each variable } p.$$

Then inductively, for each implication-conjunction formula ψ:

$$\vdash_{\mathrm{IPC}} \sigma\psi \to (\varphi \to \psi) \quad \text{and} \quad \vdash_{\mathrm{IPC}} (\varphi \to \psi) \to \sigma\psi.$$

But then $\vdash_{\mathrm{IPC}} \sigma\varphi$ and, using properties of IPC:

$$\varphi \vdash_{\mathrm{IPC}} \sigma\psi \to \psi \quad \text{and} \quad \varphi \vdash_{\mathrm{IPC}} \psi \to \sigma\psi.$$

So σ is a **projective** $\mathrm{IPC}_{\neg,\to}$**-unifier** of φ.

Theorem (Prucnal)

The $\{\to, \land\}$ fragment of IPC is hereditarily universally complete.

T. Prucnal. On the structural completeness of some pure implicational propositional calculi. *Studia Logica* 32(1): 45–50, 1973.

Prucnal's Trick

For each **implication-conjunction** formula φ of IPC, define

$$\sigma(p) = \varphi \to p \quad \text{for each variable } p.$$

Then inductively, for each implication-conjunction formula ψ:

$$\vdash_{\mathrm{IPC}} \sigma\psi \to (\varphi \to \psi) \quad \text{and} \quad \vdash_{\mathrm{IPC}} (\varphi \to \psi) \to \sigma\psi.$$

But then $\vdash_{\mathrm{IPC}} \sigma\varphi$ and, using properties of IPC:

$$\varphi \vdash_{\mathrm{IPC}} \sigma\psi \to \psi \quad \text{and} \quad \varphi \vdash_{\mathrm{IPC}} \psi \to \sigma\psi.$$

So σ is a **projective** $\mathrm{IPC}_{\neg,\to}$**-unifier** of φ.

Theorem (Prucnal)

The $\{\to, \wedge\}$ *fragment of* IPC *is hereditarily universally complete.*

T. Prucnal. On the structural completeness of some pure implicational propositional calculi. *Studia Logica* 32(1): 45–50, 1973.

Wroński's Observation

In fact, the $\{\to\}$, $\{\to,\land\}$, and $\{\to,\land,\neg\}$ fragments of *all* **intermediate logics** are hereditarily universally complete.

However, this is not the case for the $\{\to,\neg\}$ fragments; e.g.

$$\{p \to \neg q,\ (\neg\neg p \to p) \to r,\ (\neg\neg q \to q) \to r\} \rhd r$$

is admissible but not derivable in the $\{\to,\neg\}$ fragment of IPC.

A. Wroński. On factoring by compact congruences in algebras of certain varieties related to the intuitionistic logic. *Bulletin of the Section of Logic* 15(2) (1986) 48–51.

Wroński's Observation

In fact, the $\{\to\}$, $\{\to, \land\}$, and $\{\to, \land, \neg\}$ fragments of *all* **intermediate logics** are hereditarily universally complete.

However, this is not the case for the $\{\to, \neg\}$ fragments; e.g.

$$\{p \to \neg q,\ (\neg\neg p \to p) \to r,\ (\neg\neg q \to q) \to r\} \rhd r$$

is admissible but not derivable in the $\{\to, \neg\}$ fragment of IPC.

A. Wroński. On factoring by compact congruences in algebras of certain varieties related to the intuitionistic logic. *Bulletin of the Section of Logic* 15(2) (1986) 48–51.

Other Cases

- Hereditarily structurally complete intermediate logics and transitive modal logics have been fully characterized by Citkin and Rybakov.

  Admissibility of Logical Inference Rules. V. Rybakov. Elsevier, 1997.

- Medvedev's logic is the only known intermediate logic that is structurally complete but not hereditarily structurally complete.

  Structural completeness of Medvedev's propositional calculus. T. Prucnal. *Reports on Mathematical Logic* 6:103–105, 1976.

- Prucnal's trick extends to establish hereditary universal completeness for certain fragments of relevant logics.

  A structurally complete fragment of relevant logic. J. K. Slaney and R. K. Meyer. *Notre Dame Journal of Formal Logic* 33:561–566, 1992.

Other Cases

- Hereditarily structurally complete intermediate logics and transitive modal logics have been fully characterized by Citkin and Rybakov.

  Admissibility of Logical Inference Rules. V. Rybakov. Elsevier, 1997.

- Medvedev's logic is the only known intermediate logic that is structurally complete but not hereditarily structurally complete.

  Structural completeness of Medvedev's propositional calculus. T. Prucnal. *Reports on Mathematical Logic* 6:103–105, 1976.

- Prucnal's trick extends to establish hereditary universal completeness for certain fragments of relevant logics.

  A structurally complete fragment of relevant logic. J. K. Slaney and R. K. Meyer. *Notre Dame Journal of Formal Logic* 33:561–566, 1992.

Other Cases

- Hereditarily structurally complete intermediate logics and transitive modal logics have been fully characterized by Citkin and Rybakov.

  Admissibility of Logical Inference Rules. V. Rybakov. Elsevier, 1997.

- Medvedev's logic is the only known intermediate logic that is structurally complete but not hereditarily structurally complete.

  Structural completeness of Medvedev's propositional calculus. T. Prucnal. *Reports on Mathematical Logic* 6:103–105, 1976.

- Prucnal's trick extends to establish hereditary universal completeness for certain fragments of relevant logics.

  A structurally complete fragment of relevant logic. J. K. Slaney and R. K. Meyer. *Notre Dame Journal of Formal Logic* 33:561–566, 1992.

Further Questions

- How can we characterize admissible rules when structural or universal completeness *fails?*
- How can admissibility be characterized *algebraically*?
- Are admissible rules *useful* for anything?

Further Questions

- How can we characterize admissible rules when structural or universal completeness *fails?*
- How can admissibility be characterized *algebraically*?
- Are admissible rules *useful* for anything?

Further Questions

- How can we characterize admissible rules when structural or universal completeness *fails?*
- How can admissibility be characterized *algebraically*?
- Are admissible rules *useful* for anything?

Part II

An Algebraic Perspective

The Idea

For classes of algebraic structures:

- rules correspond to **clauses**
- single-conclusion rules correspond to **quasiequations**
- logics correspond to **quasivarieties**
- admissibility corresponds to validity in **free algebras**.

The Idea

For classes of algebraic structures:

- rules correspond to **clauses**
- single-conclusion rules correspond to **quasiequations**
- logics correspond to **quasivarieties**
- admissibility corresponds to validity in **free algebras**.

The Idea

For classes of algebraic structures:

- rules correspond to **clauses**
- single-conclusion rules correspond to **quasiequations**
- logics correspond to **quasivarieties**
- admissibility corresponds to validity in **free algebras**.

The Idea

For classes of algebraic structures:

- rules correspond to **clauses**
- single-conclusion rules correspond to **quasiequations**
- logics correspond to **quasivarieties**
- admissibility corresponds to validity in **free algebras**.

The Idea

For classes of algebraic structures:

- rules correspond to **clauses**
- single-conclusion rules correspond to **quasiequations**
- logics correspond to **quasivarieties**
- admissibility corresponds to validity in **free algebras**.

We fix an **algebraic language** $\mathcal{L}$ and consider classes of $\mathcal{L}$**-algebras**.

$\mathbf{Fm}_{\mathcal{L}}(X)$ denotes the **formula algebra of** $\mathcal{L}$ over a set of variables X, writing just $\mathbf{Fm}_{\mathcal{L}}$ when X is countably infinite.

An $\mathcal{L}$**-equation** is an ordered pair of $\mathcal{L}$-formulas, written $\varphi \approx \psi$.

Formulas and Equations

We fix an **algebraic language** $\mathcal{L}$ and consider classes of $\mathcal{L}$**-algebras**.

$\mathbf{Fm}_{\mathcal{L}}(X)$ denotes the **formula algebra of** $\mathcal{L}$ over a set of variables X, writing just $\mathbf{Fm}_{\mathcal{L}}$ when X is countably infinite.

An $\mathcal{L}$**-equation** is an ordered pair of $\mathcal{L}$-formulas, written $\varphi \approx \psi$.

We fix an **algebraic language** $\mathcal{L}$ and consider classes of $\mathcal{L}$**-algebras**.

$\mathbf{Fm}_{\mathcal{L}}(X)$ denotes the **formula algebra of** $\mathcal{L}$ over a set of variables X, writing just $\mathbf{Fm}_{\mathcal{L}}$ when X is countably infinite.

An $\mathcal{L}$**-equation** is an ordered pair of $\mathcal{L}$-formulas, written $\varphi \approx \psi$.

Clauses

An $\mathcal{L}$**-clause** is an ordered pair of finite sets of $\mathcal{L}$-equations, written

$$\Gamma \;\triangleright\; \Delta,$$

called an $\mathcal{L}$**-quasiequation** if $|\Delta| = 1$ and a **positive** $\mathcal{L}$**-clause** if $\Gamma = \emptyset$.

Clauses

An $\mathcal{L}$**-clause** is an ordered pair of finite sets of $\mathcal{L}$-equations, written

$$\Gamma \ \triangleright \ \Delta,$$

called an $\mathcal{L}$**-quasiequation** if $|\Delta| = 1$ and a **positive** $\mathcal{L}$**-clause** if $\Gamma = \emptyset$.

Validity

An $\mathcal{L}$-clause $\Gamma \rhd \Delta$ is **valid** in an $\mathcal{L}$-algebra **A**, written

$$\Gamma \models_{\mathbf{A}} \Delta,$$

if for every homomorphism $g \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{A}$,

$$\begin{array}{ccc} g(\varphi) = g(\psi) & \Longrightarrow & g(\varphi') = g(\psi') \\ \text{for all } \varphi \approx \psi \in \Gamma & & \text{for some } \varphi' \approx \psi' \in \Delta. \end{array}$$

An $\mathcal{L}$-clause $\Gamma \rhd \Delta$ is **valid** in a class K of $\mathcal{L}$-algebras, written $\Gamma \vdash_{\mathrm{L}} \Delta$, if $\Gamma \models_{\mathbf{A}} \Delta$ for each $\mathbf{A} \in$ K.

Validity

An $\mathcal{L}$-clause $\Gamma \rhd \Delta$ is **valid** in an $\mathcal{L}$-algebra **A**, written

$$\Gamma \models_{\mathbf{A}} \Delta,$$

if for every homomorphism $g \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{A}$,

$$\begin{array}{ccc} g(\varphi) = g(\psi) & \Longrightarrow & g(\varphi') = g(\psi') \\ \text{for all } \varphi \approx \psi \in \Gamma & & \text{for some } \varphi' \approx \psi' \in \Delta. \end{array}$$

An $\mathcal{L}$-clause $\Gamma \rhd \Delta$ is **valid** in a class K of $\mathcal{L}$-algebras, written $\Gamma \vdash_{\mathrm{L}} \Delta$, if $\Gamma \models_{\mathbf{A}} \Delta$ for each $\mathbf{A} \in$ K.

Validity

An $\mathcal{L}$-clause $\Gamma \rhd \Delta$ is **valid** in an $\mathcal{L}$-algebra **A**, written

$$\Gamma \models_{\mathbf{A}} \Delta,$$

if for every homomorphism $g \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{A}$,

$$\begin{array}{ccc} g(\varphi) = g(\psi) & \Longrightarrow & g(\varphi') = g(\psi') \\ \text{for all } \varphi \approx \psi \in \Gamma & & \text{for some } \varphi' \approx \psi' \in \Delta. \end{array}$$

An $\mathcal{L}$-clause $\Gamma \rhd \Delta$ is **valid** in a class K of $\mathcal{L}$-algebras, written $\Gamma \vdash_{\mathrm{L}} \Delta$, if $\Gamma \models_{\mathbf{A}} \Delta$ for each $\mathbf{A} \in$ K.

Admissibility

A K-**unifier** of a set of $\mathcal{L}$-equations Γ is a homomorphism $\sigma\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$ such that:

$$\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi) \quad \text{for all } \varphi \approx \psi \in \Gamma.$$

An $\mathcal{L}$-**clause** $\Gamma \rhd \Delta$ is K-**admissible** if

$$\sigma \text{ is a K-unifier of } \Gamma \quad \Longrightarrow \quad \sigma \text{ is a K-unifier of some } \varphi \approx \psi \in \Delta.$$

Admissibility

A K-**unifier** of a set of $\mathcal{L}$-equations Γ is a homomorphism $\sigma\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$ such that:

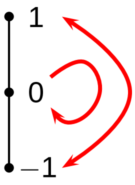
$$\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi) \quad \text{for all } \varphi \approx \psi \in \Gamma.$$

An $\mathcal{L}$**-clause** $\Gamma \rhd \Delta$ is K-**admissible** if

$$\sigma \text{ is a K-unifier of } \Gamma \quad \Longrightarrow \quad \sigma \text{ is a K-unifier of some } \varphi \approx \psi \in \Delta.$$

Example: A Three Element Algebra

Consider the Kleene lattice $\mathbf{C_3} = \langle \{-1, 0, 1\}, \wedge, \vee, \neg \rangle$ described by:



Then (since no formula is constantly 0)

$$\{p \approx \neg p\} \rhd p \approx q \quad \text{is } \mathbf{C_3}\text{-admissible, but } \{p \approx \neg p\} \not\models_{\mathbf{C_3}} p \approx q.$$

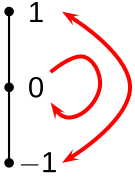
Also the following quasiequation is $\mathbf{C_3}$-admissible, but not $\mathbf{C_3}$-valid:

$$\{\neg p \leq p,\ p \wedge \neg q \leq \neg p \vee q\} \rhd \neg q \leq q$$

where $\varphi \leq \psi$ stands for $\varphi \wedge \psi \approx \varphi$.

Example: A Three Element Algebra

Consider the Kleene lattice $\mathbf{C_3} = \langle \{-1, 0, 1\}, \wedge, \vee, \neg \rangle$ described by:



Then (since no formula is constantly 0)

$$\{p \approx \neg p\} \ \triangleright\ p \approx q \quad \text{is } \mathbf{C_3}\text{-admissible, but } \{p \approx \neg p\} \not\models_{\mathbf{C_3}} p \approx q.$$

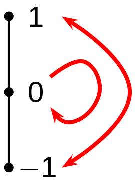
Also the following quasiequation is $\mathbf{C_3}$-admissible, but not $\mathbf{C_3}$-valid:

$$\{\neg p \leq p,\ p \wedge \neg q \leq \neg p \vee q\} \ \triangleright\ \neg q \leq q$$

where $\varphi \leq \psi$ stands for $\varphi \wedge \psi \approx \varphi$.

Example: A Three Element Algebra

Consider the Kleene lattice $\mathbf{C_3} = \langle\{-1, 0, 1\}, \wedge, \vee, \neg\rangle$ described by:



Then (since no formula is constantly 0)

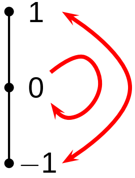
$$\{p \approx \neg p\} \vartriangleright p \approx q \quad \text{is } \mathbf{C_3}\text{-admissible, but } \{p \approx \neg p\} \not\models_{\mathbf{C_3}} p \approx q.$$

Also the following quasiequation is $\mathbf{C_3}$-admissible, but not $\mathbf{C_3}$-valid:

$$\{\neg p \leq p,\ p \wedge \neg q \leq \neg p \vee q\} \vartriangleright \neg q \leq q$$

where $\varphi \leq \psi$ stands for $\varphi \wedge \psi \approx \varphi$.

Consider the Kleene lattice $\mathbf{C_3} = \langle \{-1, 0, 1\}, \wedge, \vee, \neg \rangle$ described by:



Then (since no formula is constantly 0)

$$\{p \approx \neg p\} \rhd p \approx q \quad \text{is } \mathbf{C_3}\text{-admissible, but } \{p \approx \neg p\} \not\models_{\mathbf{C_3}} p \approx q.$$

Also the following quasiequation is $\mathbf{C_3}$-admissible, but not $\mathbf{C_3}$-valid:

$$\{\neg p \leq p,\ p \wedge \neg q \leq \neg p \vee q\} \rhd \neg q \leq q$$

where $\varphi \leq \psi$ stands for $\varphi \wedge \psi \approx \varphi$.

The Congruence $\sim_K$

For a class of $\mathcal{L}$-algebras K and $\mathrm{Fm}_{\mathcal{L}}(X) \neq \emptyset$, define for $\varphi, \psi \in \mathrm{Fm}_{\mathcal{L}}(X)$:

$$\varphi \sim_K \psi \qquad \text{iff} \qquad \models_K \varphi \approx \psi.$$

Then $\sim_K$ is a **congruence** on $\mathbf{Fm}_{\mathcal{L}}(X)$, i.e., an equivalence relation satisfying for each n-ary function symbol f of $\mathcal{L}$:

$$\begin{array}{c} \varphi_1 \sim_K \psi_1 \\ \vdots \\ \varphi_n \sim_K \psi_n \end{array} \quad \Longrightarrow \quad f(\varphi_1, \ldots, \varphi_n) \sim_K f(\psi_1, \ldots, \psi_n).$$

The Congruence $\sim_K$

For a class of $\mathcal{L}$-algebras K and $\mathrm{Fm}_{\mathcal{L}}(X) \neq \emptyset$, define for $\varphi, \psi \in \mathrm{Fm}_{\mathcal{L}}(X)$:

$$\varphi \sim_K \psi \qquad \text{iff} \qquad \models_K \varphi \approx \psi.$$

Then $\sim_K$ is a **congruence** on $\mathbf{Fm}_{\mathcal{L}}(X)$, i.e., an equivalence relation satisfying for each n-ary function symbol f of $\mathcal{L}$:

$$\begin{array}{c} \varphi_1 \sim_K \psi_1 \\ \vdots \\ \varphi_n \sim_K \psi_n \end{array} \quad \Longrightarrow \quad f(\varphi_1, \ldots, \varphi_n) \sim_K f(\psi_1, \ldots, \psi_n).$$

The Congruence $\sim_K$

For a class of $\mathcal{L}$-algebras K and $\mathrm{Fm}_{\mathcal{L}}(X) \neq \emptyset$, define for $\varphi, \psi \in \mathrm{Fm}_{\mathcal{L}}(X)$:

$$\varphi \sim_K \psi \qquad \text{iff} \qquad \models_K \varphi \approx \psi.$$

Then $\sim_K$ is a **congruence** on $\mathbf{Fm}_{\mathcal{L}}(X)$, i.e., an equivalence relation satisfying for each n-ary function symbol f of $\mathcal{L}$:

$$\begin{array}{c} \varphi_1 \sim_K \psi_1 \\ \vdots \\ \varphi_n \sim_K \psi_n \end{array} \quad \Longrightarrow \quad f(\varphi_1, \ldots, \varphi_n) \sim_K f(\psi_1, \ldots, \psi_n).$$

Free Algebras

The *quotient algebra*

$$\mathbf{F}_{\mathsf{K}}(X) = \mathbf{Fm}_{\mathcal{L}}(X) \,/\, \sim_{\mathsf{K}}$$

with universe $\mathrm{F}_{\mathsf{K}}(X) = \{[\varphi]_{\sim_{\mathsf{K}}} \mid \varphi \in \mathrm{Fm}_{\mathcal{L}}(X)\}$ and operations

$$f([\varphi_1]_{\sim_{\mathsf{K}}}, \ldots, [\varphi_n]_{\sim_{\mathsf{K}}}) = [f(\varphi_1, \ldots, \varphi_n)]_{\sim_{\mathsf{K}}}$$

is called the X-**generated free algebra** of K.

In particular, $\mathbf{F}_{\mathsf{K}}(\omega)$ is the **free algebra on countably infinitely many generators** of K.

The *quotient algebra*

$$\mathbf{F}_{\mathsf{K}}(X) \;=\; \mathbf{Fm}_{\mathcal{L}}(X) \,/ \sim_{\mathsf{K}}$$

with universe $\mathrm{F}_{\mathsf{K}}(X) = \{[\varphi]_{\sim_{\mathsf{K}}} \mid \varphi \in \mathrm{Fm}_{\mathcal{L}}(X)\}$ and operations

$$f([\varphi_1]_{\sim_{\mathsf{K}}}, \ldots, [\varphi_n]_{\sim_{\mathsf{K}}}) = [f(\varphi_1, \ldots, \varphi_n)]_{\sim_{\mathsf{K}}}$$

is called the X-**generated free algebra** of K.

In particular, $\mathbf{F}_{\mathsf{K}}(\omega)$ is the **free algebra on countably infinitely many generators** of K.

Free Algebras

The *quotient algebra*

$$\mathbf{F}_{\mathsf{K}}(X) \;=\; \mathbf{Fm}_{\mathcal{L}}(X) \,/\, \sim_{\mathsf{K}}$$

with universe $\mathrm{F}_{\mathsf{K}}(X) = \{[\varphi]_{\sim_{\mathsf{K}}} \mid \varphi \in \mathrm{Fm}_{\mathcal{L}}(X)\}$ and operations

$$f([\varphi_1]_{\sim_{\mathsf{K}}}, \ldots, [\varphi_n]_{\sim_{\mathsf{K}}}) = [f(\varphi_1, \ldots, \varphi_n)]_{\sim_{\mathsf{K}}}$$

is called the X-**generated free algebra** of K.

In particular, $\mathbf{F}_{\mathsf{K}}(\omega)$ is the **free algebra on countably infinitely many generators** of K.

Free Algebras

The *quotient algebra*

$$\mathbf{F}_{\mathsf{K}}(X) \;=\; \mathbf{Fm}_{\mathcal{L}}(X)\,/\sim_{\mathsf{K}}$$

with universe $\mathrm{F}_{\mathsf{K}}(X) = \{[\varphi]_{\sim_{\mathsf{K}}} \mid \varphi \in \mathrm{Fm}_{\mathcal{L}}(X)\}$ and operations

$$f([\varphi_1]_{\sim_{\mathsf{K}}}, \ldots, [\varphi_n]_{\sim_{\mathsf{K}}}) = [f(\varphi_1, \ldots, \varphi_n)]_{\sim_{\mathsf{K}}}$$

is called the X-**generated free algebra** of K.

In particular, $\mathbf{F}_{\mathsf{K}}(\omega)$ is the **free algebra on countably infinitely many generators** of K.

Free Algebras

The *quotient algebra*

$$\mathbf{F}_{\mathsf{K}}(X) \;=\; \mathbf{Fm}_{\mathcal{L}}(X)\,/\,\sim_{\mathsf{K}}$$

with universe $\mathrm{F}_{\mathsf{K}}(X) = \{[\varphi]_{\sim_{\mathsf{K}}} \mid \varphi \in \mathrm{Fm}_{\mathcal{L}}(X)\}$ and operations

$$f([\varphi_1]_{\sim_{\mathsf{K}}}, \ldots, [\varphi_n]_{\sim_{\mathsf{K}}}) = [f(\varphi_1, \ldots, \varphi_n)]_{\sim_{\mathsf{K}}}$$

is called the X-**generated free algebra** of K.

In particular, $\mathbf{F}_{\mathsf{K}}(\omega)$ is the **free algebra on countably infinitely many generators** of K.

The **canonical homomorphism**

$$h_{\mathsf{K}} \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$$

maps each $\mathcal{L}$-formula to its equivalence class in the free algebra,

$$h_{\mathsf{K}}(\varphi) = [\varphi]_{\sim_{\mathsf{K}}}.$$

Then for each $\mathcal{L}$-equation $\varphi \approx \psi$:

$$\models_{\mathsf{K}} \varphi \approx \psi \quad \text{iff} \quad \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \varphi \approx \psi \quad \text{iff} \quad h_{\mathsf{K}}(\varphi) = h_{\mathsf{K}}(\psi). \qquad (1)$$

S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Graduate Texts in Mathematics 78. Springer, 1981.

The **canonical homomorphism**

$$h_{\mathsf{K}} \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$$

maps each $\mathcal{L}$-formula to its equivalence class in the free algebra,

$$h_{\mathsf{K}}(\varphi) = [\varphi]_{\sim_{\mathsf{K}}}.$$

Then for each $\mathcal{L}$-equation $\varphi \approx \psi$:

$$\models_{\mathsf{K}} \varphi \approx \psi \quad \text{iff} \quad \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \varphi \approx \psi \quad \text{iff} \quad h_{\mathsf{K}}(\varphi) = h_{\mathsf{K}}(\psi). \qquad (1)$$

S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Graduate Texts in Mathematics 78. Springer, 1981.

Free Algebras and Admissibility

Lemma

$\Gamma \rhd \Delta$ *is* K*-admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Free Algebras and Admissibility

Lemma

$\Gamma \rhd \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Free Algebras and Admissibility

Lemma

$\Gamma \rhd \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Free Algebras and Admissibility

Lemma

$\Gamma \triangleright \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \triangleright \Delta$ is K-admissible and consider a homomorphism $g\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Free Algebras and Admissibility

Lemma

$\Gamma \rhd \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Free Algebras and Admissibility

Lemma

$\Gamma \rhd \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Free Algebras and Admissibility

Lemma

$\Gamma \rhd \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma \colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Lemma

$\Gamma \rhd \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Lemma

$\Gamma \rhd \Delta$ *is* K-*admissible iff* $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \Delta$.

Proof.

($\Rightarrow$) Suppose that $\Gamma \rhd \Delta$ is K-admissible and consider a homomorphism $g\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{F}_{\mathsf{K}}(\omega)$ such that $g(\varphi) = g(\psi)$ for all $\varphi \approx \psi \in \Gamma$.

Any map sending each variable p to a member of the equivalence class $g(p)$ extends to a homomorphism $\sigma\colon \mathbf{Fm}_{\mathcal{L}} \to \mathbf{Fm}_{\mathcal{L}}$.

Since $h_{\mathsf{K}}(\sigma(p)) = g(p)$ for each variable p, it follows that $h_{\mathsf{K}} \circ \sigma = g$.

So for each $\varphi \approx \psi \in \Gamma$, also $h_{\mathsf{K}}(\sigma(\varphi)) = h_{\mathsf{K}}(\sigma(\psi))$ and by (1), $\models_{\mathsf{K}} \sigma(\varphi) \approx \sigma(\psi)$. Hence $\models_{\mathsf{K}} \sigma(\varphi') \approx \sigma(\psi')$ for some $\varphi' \approx \psi' \in \Delta$.

But then again by (1), $g(\varphi') = h_{\mathsf{K}}(\sigma(\varphi')) = h_{\mathsf{K}}(\sigma(\psi')) = g(\psi')$.

($\Leftarrow$) Very similar (in fact, a bit easier). $\square$

Varieties

A **variety** is a class of $\mathcal{L}$-algebras axiomatized by equations.

Varieties are also (Birkhoff's theorem) the classes of $\mathcal{L}$-algebras closed under taking **homomorphic images**, **subalgebras**, and **products**.

$\mathbb{V}(\mathsf{K}) = \mathbb{HSP}(\mathsf{K})$ is the **smallest variety containing** K, and

$$\mathbb{V}(\mathsf{K}_1) = \mathbb{V}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-equations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbb{V}(\mathsf{K}) \quad = \quad \mathbb{V}(\mathbf{F}_{\mathsf{K}}(\omega)).$$

Varieties

A **variety** is a class of $\mathcal{L}$-algebras axiomatized by equations.

Varieties are also (Birkhoff's theorem) the classes of $\mathcal{L}$-algebras closed under taking **homomorphic images**, **subalgebras**, and **products**.

$\mathbb{V}(\mathsf{K}) = \mathbb{HSP}(\mathsf{K})$ is the **smallest variety containing** K, and

$$\mathbb{V}(\mathsf{K}_1) = \mathbb{V}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-equations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbb{V}(\mathsf{K}) \quad = \quad \mathbb{V}(\mathbf{F}_{\mathsf{K}}(\omega)).$$

Varieties

A **variety** is a class of $\mathcal{L}$-algebras axiomatized by equations.

Varieties are also (Birkhoff's theorem) the classes of $\mathcal{L}$-algebras closed under taking **homomorphic images**, **subalgebras**, and **products**.

$\mathbb{V}(\mathsf{K}) = \mathbb{HSP}(\mathsf{K})$ is the **smallest variety containing** K, and

$$\mathbb{V}(\mathsf{K}_1) = \mathbb{V}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-equations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbb{V}(\mathsf{K}) \quad = \quad \mathbb{V}(\mathbf{F}_{\mathsf{K}}(\omega)).$$

Varieties

A **variety** is a class of $\mathcal{L}$-algebras axiomatized by equations.

Varieties are also (Birkhoff's theorem) the classes of $\mathcal{L}$-algebras closed under taking **homomorphic images**, **subalgebras**, and **products**.

$\mathbb{V}(\mathsf{K}) = \mathbb{HSP}(\mathsf{K})$ is the **smallest variety containing** K, and

$$\mathbb{V}(\mathsf{K}_1) = \mathbb{V}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-equations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbb{V}(\mathsf{K}) \quad = \quad \mathbb{V}(\mathbf{F}_{\mathsf{K}}(\omega)).$$

Varieties

A **variety** is a class of $\mathcal{L}$-algebras axiomatized by equations.

Varieties are also (Birkhoff's theorem) the classes of $\mathcal{L}$-algebras closed under taking **homomorphic images**, **subalgebras**, and **products**.

$\mathbb{V}(\mathsf{K}) = \mathbb{HSP}(\mathsf{K})$ is the **smallest variety containing** K, and

$$\mathbb{V}(\mathsf{K}_1) = \mathbb{V}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-equations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbb{V}(\mathsf{K}) \quad = \quad \mathbb{V}(\mathbf{F}_{\mathsf{K}}(\omega)).$$

Quasivarieties

A **quasivariety** is a class of $\mathcal{L}$-algebras axiomatized by quasiequations.

Quasivarieties are also the classes of $\mathcal{L}$-algebras closed under taking **isomorphic images**, **subalgebras**, **products**, and **ultraproducts**.

$\mathbb{Q}(\mathsf{K}) = \mathbb{ISPP}_U(\mathsf{K})$ is the **smallest quasivariety containing** K, and

$$\mathbb{Q}(\mathsf{K}_1) = \mathbb{Q}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-quasiequations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbf{F}_{\mathsf{K}}(\omega) \quad = \quad \mathbf{F}_{\mathbb{Q}(\mathsf{K})}(\omega).$$

Quasivarieties

A **quasivariety** is a class of $\mathcal{L}$-algebras axiomatized by quasiequations.

Quasivarieties are also the classes of $\mathcal{L}$-algebras closed under taking **isomorphic images**, **subalgebras**, **products**, and **ultraproducts**.

$\mathbb{Q}(\mathsf{K}) = \mathbb{ISPP}_U(\mathsf{K})$ is the **smallest quasivariety containing** K, and

$$\mathbb{Q}(\mathsf{K}_1) = \mathbb{Q}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-quasiequations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbf{F}_{\mathsf{K}}(\omega) \quad = \quad \mathbf{F}_{\mathbb{Q}(\mathsf{K})}(\omega).$$

A **quasivariety** is a class of $\mathcal{L}$-algebras axiomatized by quasiequations.

Quasivarieties are also the classes of $\mathcal{L}$-algebras closed under taking **isomorphic images**, **subalgebras**, **products**, and **ultraproducts**.

$\mathbb{Q}(\mathsf{K}) = \mathbb{ISPP}_U(\mathsf{K})$ is the **smallest quasivariety containing** K, and

$$\mathbb{Q}(\mathsf{K}_1) = \mathbb{Q}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-quasiequations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbf{F}_{\mathsf{K}}(\omega) \quad = \quad \mathbf{F}_{\mathbb{Q}(\mathsf{K})}(\omega).$$

Quasivarieties

A **quasivariety** is a class of $\mathcal{L}$-algebras axiomatized by quasiequations.

Quasivarieties are also the classes of $\mathcal{L}$-algebras closed under taking **isomorphic images**, **subalgebras**, **products**, and **ultraproducts**.

$\mathbb{Q}(\mathsf{K}) = \mathbb{ISPP}_U(\mathsf{K})$ is the **smallest quasivariety containing** K, and

$$\mathbb{Q}(\mathsf{K}_1) = \mathbb{Q}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-quasiequations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbf{F}_{\mathsf{K}}(\omega) \quad = \quad \mathbf{F}_{\mathbb{Q}(\mathsf{K})}(\omega).$$

A **quasivariety** is a class of $\mathcal{L}$-algebras axiomatized by quasiequations.

Quasivarieties are also the classes of $\mathcal{L}$-algebras closed under taking **isomorphic images**, **subalgebras**, **products**, and **ultraproducts**.

$\mathbb{Q}(\mathsf{K}) = \mathbb{ISPP}_U(\mathsf{K})$ is the **smallest quasivariety containing** K, and

$$\mathbb{Q}(\mathsf{K}_1) = \mathbb{Q}(\mathsf{K}_2) \quad \text{iff} \quad \text{the same } \mathcal{L}\text{-quasiequations are valid in } \mathsf{K}_1 \text{ and } \mathsf{K}_2.$$

In particular, for any class of $\mathcal{L}$-algebras K:

$$\mathbf{F}_{\mathsf{K}}(\omega) \quad = \quad \mathbf{F}_{\mathbb{Q}(\mathsf{K})}(\omega).$$

Admissibility Algebraically

Theorem

The following are equivalent:

(i) $\Gamma \rhd \varphi \approx \psi$ *is* K*-admissible*

(ii) $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \varphi \approx \psi$

(iii) $\mathbb{V}(\mathsf{K}) = \mathbb{V}(\{\mathbf{A} \in \mathbb{Q}(\mathsf{K}) \mid \Gamma \models_{\mathbf{A}} \varphi \approx \psi\})$.

Admissibility Algebraically

Theorem

The following are equivalent:

(i) $\Gamma \rhd \varphi \approx \psi$ *is* K*-admissible*

(ii) $\Gamma \models_{\mathbf{F}_{\mathsf{K}}(\omega)} \varphi \approx \psi$

(iii) $\mathbb{V}(\mathsf{K}) = \mathbb{V}(\{\mathbf{A} \in \mathbb{Q}(\mathsf{K}) \mid \Gamma \models_{\mathbf{A}} \varphi \approx \psi\})$.

Structural Completeness Algebraically

Definition

If K-admissibility and K-validity coincide for $\mathcal{L}$*-quasiequations*, that is,

$$\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_{\mathsf{K}}(\omega)),$$

then K is called **structurally complete**.

C. Bergman. Structural completeness in algebra and logic.
In *Algebraic Logic, Colloquia Mathematica Societatis János Bolyai*, 54:59–73, 1991.

Structural Completeness Algebraically

Definition

If K-admissibility and K-validity coincide for $\mathcal{L}$*-quasiequations*, that is,

$$\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_{\mathsf{K}}(\omega)),$$

then K is called **structurally complete**.

C. Bergman. Structural completeness in algebra and logic.
In *Algebraic Logic, Colloquia Mathematica Societatis János Bolyai*, 54:59–73, 1991.

Proving Structural Completeness

In our proof that **classical propositional logic** is structurally complete, we essentially gave an *embedding* of the Boolean algebra $\mathbf{2}$ into $\mathbf{F_2}(\omega)$.

More generally, if each $\mathbf{A} \in \mathsf{K}$ *embeds into* $\mathbf{F}_{\mathsf{K}}(\omega)$, then

$$\mathsf{K} \subseteq \mathbb{IS}(\mathbf{F}_{\mathsf{K}}(\omega)) \subseteq \mathbb{Q}(\mathbf{F}_{\mathsf{K}}(\omega)) \subseteq \mathbb{Q}(\mathsf{K}),$$

so $\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_{\mathsf{K}}(\omega))$ and K (also $\mathbb{Q}(\mathsf{K})$) is **structurally complete**.

J. S. Olson, J. G. Raftery, and C. J. Van Alten. Structural completeness in substructural logics. *Logic Journal of the IGPL* 16(5): 453–495, 2008.

P. Cintula and G. Metcalfe. Structural completeness in fuzzy logics. *Notre Dame Journal of Formal Logic* 50(2): 153–183, 2009.

Proving Structural Completeness

In our proof that **classical propositional logic** is structurally complete, we essentially gave an *embedding* of the Boolean algebra $\mathbf{2}$ into $\mathbf{F_2}(\omega)$.

More generally, if each $\mathbf{A} \in \mathsf{K}$ *embeds into* $\mathbf{F}_\mathsf{K}(\omega)$, then

$$\mathsf{K} \subseteq \mathbb{IS}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathsf{K}),$$

so $\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega))$ and K (also $\mathbb{Q}(\mathsf{K})$) is **structurally complete**.

J. S. Olson, J. G. Raftery, and C. J. Van Alten. Structural completeness in substructural logics. *Logic Journal of the IGPL* 16(5): 453–495, 2008.

P. Cintula and G. Metcalfe. Structural completeness in fuzzy logics. *Notre Dame Journal of Formal Logic* 50(2): 153–183, 2009.

Proving Structural Completeness

In our proof that **classical propositional logic** is structurally complete, we essentially gave an *embedding* of the Boolean algebra $\mathbf{2}$ into $\mathbf{F_2}(\omega)$.

More generally, if each $\mathbf{A} \in \mathsf{K}$ *embeds into* $\mathbf{F}_\mathsf{K}(\omega)$, then

$$\mathsf{K} \subseteq \mathbb{IS}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathsf{K}),$$

so $\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega))$ and K (also $\mathbb{Q}(\mathsf{K})$) is **structurally complete**.

J. S. Olson, J. G. Raftery, and C. J. Van Alten. Structural completeness in substructural logics. *Logic Journal of the IGPL* 16(5): 453–495, 2008.

P. Cintula and G. Metcalfe. Structural completeness in fuzzy logics. *Notre Dame Journal of Formal Logic* 50(2): 153–183, 2009.

Proving Structural Completeness

In our proof that **classical propositional logic** is structurally complete, we essentially gave an *embedding* of the Boolean algebra $\mathbf{2}$ into $\mathbf{F_2}(\omega)$.

More generally, if each $\mathbf{A} \in \mathsf{K}$ *embeds into* $\mathbf{F}_\mathsf{K}(\omega)$, then

$$\mathsf{K} \subseteq \mathbb{IS}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathsf{K}),$$

so $\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega))$ and K (also $\mathbb{Q}(\mathsf{K})$) is **structurally complete**.

J. S. Olson, J. G. Raftery, and C. J. Van Alten. Structural completeness in substructural logics. *Logic Journal of the IGPL* 16(5): 453–495, 2008.

P. Cintula and G. Metcalfe. Structural completeness in fuzzy logics. *Notre Dame Journal of Formal Logic* 50(2): 153–183, 2009.

Proving Structural Completeness

In our proof that **classical propositional logic** is structurally complete, we essentially gave an *embedding* of the Boolean algebra $\mathbf{2}$ into $\mathbf{F_2}(\omega)$.

More generally, if each $\mathbf{A} \in \mathsf{K}$ *embeds into* $\mathbf{F}_{\mathsf{K}}(\omega)$, then

$$\mathsf{K} \subseteq \mathbb{IS}(\mathbf{F}_{\mathsf{K}}(\omega)) \subseteq \mathbb{Q}(\mathbf{F}_{\mathsf{K}}(\omega)) \subseteq \mathbb{Q}(\mathsf{K}),$$

so $\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_{\mathsf{K}}(\omega))$ and K (also $\mathbb{Q}(\mathsf{K})$) is **structurally complete**.

J. S. Olson, J. G. Raftery, and C. J. Van Alten. Structural completeness in substructural logics. *Logic Journal of the IGPL* 16(5): 453–495, 2008.

P. Cintula and G. Metcalfe. Structural completeness in fuzzy logics. *Notre Dame Journal of Formal Logic* 50(2): 153–183, 2009.

Proving Structural Completeness

In our proof that **classical propositional logic** is structurally complete, we essentially gave an *embedding* of the Boolean algebra $\mathbf{2}$ into $\mathbf{F_2}(\omega)$.

More generally, if each $\mathbf{A} \in \mathsf{K}$ *embeds into* $\mathbf{F}_\mathsf{K}(\omega)$, then

$$\mathsf{K} \subseteq \mathbb{IS}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega)) \subseteq \mathbb{Q}(\mathsf{K}),$$

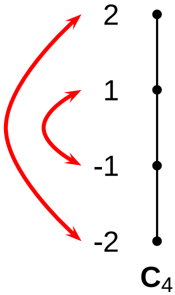
so $\mathbb{Q}(\mathsf{K}) = \mathbb{Q}(\mathbf{F}_\mathsf{K}(\omega))$ and K (also $\mathbb{Q}(\mathsf{K})$) is **structurally complete**.

J. S. Olson, J. G. Raftery, and C. J. Van Alten. Structural completeness in substructural logics. *Logic Journal of the IGPL* 16(5): 453–495, 2008.

P. Cintula and G. Metcalfe. Structural completeness in fuzzy logics. *Notre Dame Journal of Formal Logic* 50(2): 153–183, 2009.

Example: A Four Element Algebra

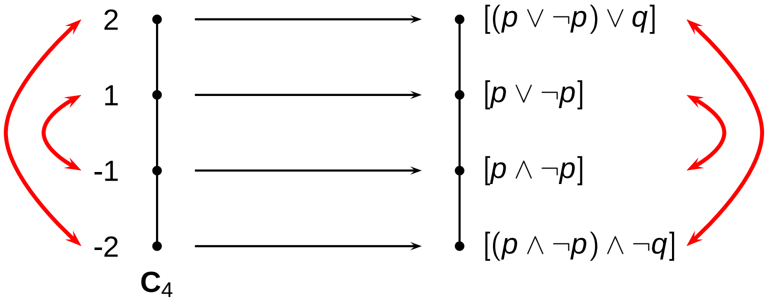
The Kleene lattice $\mathbf{C_4} = \langle\{-2, -1, 1, 2\}, \wedge, \vee, \neg\rangle$ with the usual linear order and $\neg x = -x$ can be embedded into $\mathbf{F_{C_4}}(\omega)$ as follows:



So $\mathbf{C_4}$ (equivalently, $\mathbb{Q}(\mathbf{C_4})$) is structurally complete.

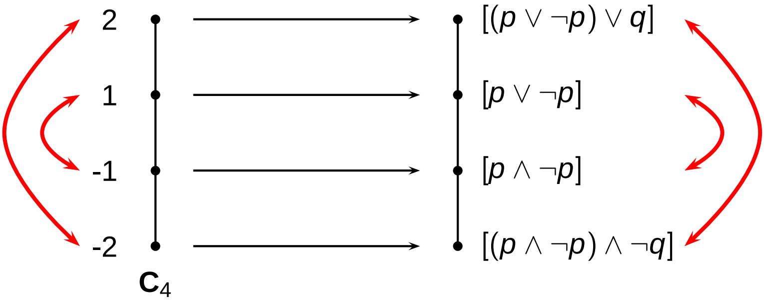
Example: A Four Element Algebra

The Kleene lattice $\mathbf{C_4} = \langle \{-2, -1, 1, 2\}, \wedge, \vee, \neg \rangle$ with the usual linear order and $\neg x = -x$ can be embedded into $\mathbf{F_{C_4}}(\omega)$ as follows:



So $\mathbf{C_4}$ (equivalently, $\mathbb{Q}(\mathbf{C_4})$) is structurally complete.

The Kleene lattice $\mathbf{C_4} = \langle \{-2, -1, 1, 2\}, \wedge, \vee, \neg \rangle$ with the usual linear order and $\neg x = -x$ can be embedded into $\mathbf{F_{C_4}}(\omega)$ as follows:



So $\mathbf{C_4}$ (equivalently, $\mathbb{Q}(\mathbf{C_4})$) is structurally complete.

Checking Admissibility in Finite Algebras

For a **finite** $\mathcal{L}$-algebra **A**:

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \qquad \text{iff} \qquad \Gamma \models_{\mathbf{F_A}(|A|)} \varphi \approx \psi.$$

Also, $\mathbf{F_A}(|A|)$ is finite, so checking **A**-admissibility is **decidable**.

But $\mathbf{F_A}(n)$ can be big even for small $|A|$ and n, e.g., $|\mathbf{F}_{\mathbf{C_3}}(2)| = 82\ldots$

However, for any (perhaps small!) *subalgebra* **B** of $\mathbf{F_A}(|A|)$ such that **A** is a *homomorphic image* of **B**:

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \qquad \text{iff} \qquad \Gamma \models_{\mathbf{B}} \varphi \approx \psi.$$

Checking Admissibility in Finite Algebras

For a **finite** $\mathcal{L}$-algebra **A**:

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \qquad \text{iff} \qquad \Gamma \models_{\mathbf{F_A}(|A|)} \varphi \approx \psi.$$

Also, $\mathbf{F_A}(|A|)$ is finite, so checking **A**-admissibility is **decidable**.

But $\mathbf{F_A}(n)$ can be big even for small $|A|$ and n, e.g., $|\mathbf{F_{C_3}}(2)| = 82\ldots$

However, for any (perhaps small!) *subalgebra* **B** of $\mathbf{F_A}(|A|)$ such that **A** is a *homomorphic image* of **B**:

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \quad \text{iff} \quad \Gamma \models_{\mathbf{B}} \varphi \approx \psi.$$

Checking Admissibility in Finite Algebras

For a **finite** $\mathcal{L}$-algebra **A**:

$$\Gamma \triangleright \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \qquad \text{iff} \qquad \Gamma \models_{\mathbf{F_A}(|A|)} \varphi \approx \psi.$$

Also, $\mathbf{F_A}(|A|)$ is finite, so checking **A**-admissibility is **decidable**.

But $\mathbf{F_A}(n)$ can be big even for small $|A|$ and n, e.g., $|\mathbf{F_{C_3}}(2)| = 82\ldots$

However, for any (perhaps small!) *subalgebra* **B** of $\mathbf{F_A}(|A|)$ such that **A** is a *homomorphic image* of **B**:

$$\Gamma \triangleright \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \qquad \text{iff} \qquad \Gamma \models_{\mathbf{B}} \varphi \approx \psi.$$

Checking Admissibility in Finite Algebras

For a **finite** $\mathcal{L}$-algebra **A**:

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \qquad \text{iff} \qquad \Gamma \models_{\mathbf{F_A}(|A|)} \varphi \approx \psi.$$

Also, $\mathbf{F_A}(|A|)$ is finite, so checking **A**-admissibility is **decidable**.

But $\mathbf{F_A}(n)$ can be big even for small $|A|$ and n, e.g., $|\mathbf{F_{C_3}}(2)| = 82\ldots$

However, for any (perhaps small!) *subalgebra* **B** of $\mathbf{F_A}(|A|)$ such that **A** is a *homomorphic image* of **B**:

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \quad \text{iff} \quad \Gamma \models_{\mathbf{B}} \varphi \approx \psi.$$

A Simple Procedure

For a finite algebra **A**:

(i) Compute the set $\mathbb{S}(\mathbf{F_A}(|A|))$ of subalgebras of $\mathbf{F_A}(|A|)$.

(ii) Construct the set $\mathrm{Adm}(\mathbf{A}) = \{\mathbf{B} \in \mathbb{S}(\mathbf{F_A}(|A|)) \mid \mathbf{A} \in \mathbb{H}(\mathbf{B})\}$.

(iii) Find a proof system to check validity in a smallest $\mathbf{B} \in \mathrm{Adm}(\mathbf{A})$.

Steps (i)-(ii) have been implemented using the Algebra Workbench; step (iii) can be implemented using, e.g., MUltlog/MUltseq or $_3T^AP$.

G. Metcalfe and C. Röthlisberger. Unifiability and admissibility in finite algebras. *Proceedings of CiE 2012*, LNCS 7318, 485–495. Springer, 2012.

A Simple Procedure

For a finite algebra $\mathbf{A}$:

(i) Compute the set $\mathbb{S}(\mathbf{F_A}(|A|))$ of subalgebras of $\mathbf{F_A}(|A|)$.

(ii) Construct the set $\mathrm{Adm}(\mathbf{A}) = \{\mathbf{B} \in \mathbb{S}(\mathbf{F_A}(|A|)) \mid \mathbf{A} \in \mathbb{H}(\mathbf{B})\}$.

(iii) Find a proof system to check validity in a smallest $\mathbf{B} \in \mathrm{Adm}(\mathbf{A})$.

Steps (i)-(ii) have been implemented using the Algebra Workbench; step (iii) can be implemented using, e.g., MUltlog/MUltseq or $_3T^AP$.

G. Metcalfe and C. Röthlisberger. Unifiability and admissibility in finite algebras. *Proceedings of CiE 2012*, LNCS 7318, 485–495. Springer, 2012.

A Simple Procedure

For a finite algebra **A**:

(i) Compute the set $\mathbb{S}(\mathbf{F_A}(|A|))$ of subalgebras of $\mathbf{F_A}(|A|)$.

(ii) Construct the set $\mathrm{Adm}(\mathbf{A}) = \{\mathbf{B} \in \mathbb{S}(\mathbf{F_A}(|A|)) \mid \mathbf{A} \in \mathbb{H}(\mathbf{B})\}$.

(iii) Find a proof system to check validity in a smallest $\mathbf{B} \in \mathrm{Adm}(\mathbf{A})$.

Steps (i)-(ii) have been implemented using the Algebra Workbench; step (iii) can be implemented using, e.g., MUltlog/MUltseq or $_3T^AP$.

G. Metcalfe and C. Röthlisberger. Unifiability and admissibility in finite algebras. *Proceedings of CiE 2012*, LNCS 7318, 485–495. Springer, 2012.

A Simple Procedure

For a finite algebra $\mathbf{A}$:

(i) Compute the set $\mathbb{S}(\mathbf{F_A}(|A|))$ of subalgebras of $\mathbf{F_A}(|A|)$.

(ii) Construct the set $\mathrm{Adm}(\mathbf{A}) = \{\mathbf{B} \in \mathbb{S}(\mathbf{F_A}(|A|)) \mid \mathbf{A} \in \mathbb{H}(\mathbf{B})\}$.

(iii) Find a proof system to check validity in a smallest $\mathbf{B} \in \mathrm{Adm}(\mathbf{A})$.

Steps (i)-(ii) have been implemented using the Algebra Workbench; step (iii) can be implemented using, e.g., MUltlog/MUltseq or $_3T^AP$.

G. Metcalfe and C. Röthlisberger. Unifiability and admissibility in finite algebras. *Proceedings of CiE 2012*, LNCS 7318, 485–495. Springer, 2012.

A Simple Procedure

For a finite algebra **A**:

(i) Compute the set $\mathbb{S}(\mathbf{F_A}(|A|))$ of subalgebras of $\mathbf{F_A}(|A|)$.

(ii) Construct the set $\mathrm{Adm}(\mathbf{A}) = \{\mathbf{B} \in \mathbb{S}(\mathbf{F_A}(|A|)) \mid \mathbf{A} \in \mathbb{H}(\mathbf{B})\}$.

(iii) Find a proof system to check validity in a smallest $\mathbf{B} \in \mathrm{Adm}(\mathbf{A})$.

Steps (i)-(ii) have been implemented using the Algebra Workbench; step (iii) can be implemented using, e.g., MUltlog/MUltseq or $_3T^AP$.

G. Metcalfe and C. Röthlisberger. Unifiability and admissibility in finite algebras. *Proceedings of CiE 2012*, LNCS 7318, 485–495. Springer, 2012.

Example: A Three Element Algebra

Consider, e.g., $\mathbf{S_3^{\to}} = \langle \{-1, 0, 1\}, \to \rangle$ with operation table:

$\to$	-1	0	1
-1	1	1	1
0	-1	0	1
1	-1	-1	1

The procedure discovers a subalgebra of the 60-element free algebra $\mathbf{F}_{\mathbf{S_3^{\to}}}(2)$ isomorphic to $\mathbf{S_3^{\to}}$, and hence that $\mathbf{S_3^{\to}}$ is structurally complete.

Structural completeness has also been confirmed for the 3-element implicational Łukasiewicz algebra, Gödel algebra, and Stone algebra.

Example: A Three Element Algebra

Consider, e.g., $\mathbf{S}_3^{\rightarrow} = \langle \{-1, 0, 1\}, \rightarrow \rangle$ with operation table:

$\rightarrow$	-1	0	1
-1	1	1	1
0	-1	0	1
1	-1	-1	1

The procedure discovers a subalgebra of the 60-element free algebra $\mathbf{F}_{\mathbf{S}_3^{\rightarrow}}(2)$ isomorphic to $\mathbf{S}_3^{\rightarrow}$, and hence that $\mathbf{S}_3^{\rightarrow}$ is structurally complete.

Structural completeness has also been confirmed for the 3-element implicational Łukasiewicz algebra, Gödel algebra, and Stone algebra.

Example: A Three Element Algebra

Consider, e.g., $\mathbf{S_3^{\to}} = \langle \{-1, 0, 1\}, \to \rangle$ with operation table:

$\to$	-1	0	1
-1	1	1	1
0	-1	0	1
1	-1	-1	1

The procedure discovers a subalgebra of the 60-element free algebra $\mathbf{F_{S_3^{\to}}}(2)$ isomorphic to $\mathbf{S_3^{\to}}$, and hence that $\mathbf{S_3^{\to}}$ is structurally complete.

Structural completeness has also been confirmed for the 3-element implicational Łukasiewicz algebra, Gödel algebra, and Stone algebra.

Almost Structural Completeness

A is called **almost structurally complete** if **A**-admissibility coincides with **A**-validity for quasiequations with **A**-unifiable premises; that is

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible } \ \& \ \Gamma \text{ is } \mathbf{A}\text{-unifiable} \quad \Longrightarrow \quad \Gamma \models_{\mathbf{A}} \varphi \approx \psi.$$

Lemma

For any finite algebra **A** *and subalgebra* **B** *of* $\mathbf{F_A}(1)$*:*

$$\mathbf{A} \textit{ is almost structurally complete} \quad \textit{iff} \quad \mathbb{Q}(\mathbf{F_A}(|A|)) = \mathbb{Q}(\mathbf{A} \times \mathbf{B}).$$

A is called **almost structurally complete** if **A**-admissibility coincides with **A**-validity for quasiequations with **A**-unifiable premises; that is

$$\Gamma \rhd \varphi \approx \psi \text{ is } \mathbf{A}\text{-admissible} \ \ \& \ \ \Gamma \text{ is } \mathbf{A}\text{-unifiable} \quad \Longrightarrow \quad \Gamma \models_{\mathbf{A}} \varphi \approx \psi.$$

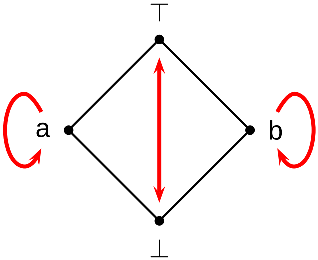
Lemma

For any finite algebra **A** *and subalgebra* **B** *of* $\mathbf{F_A}(1)$*:*

$$\mathbf{A} \text{ is almost structurally complete} \quad \textit{iff} \quad \mathbb{Q}(\mathbf{F_A}(|A|)) = \mathbb{Q}(\mathbf{A} \times \mathbf{B}).$$

Example: A Four Element Algebra

For the De Morgan lattice $\mathbf{D_4} = \langle \{\bot, a, b, \top\}, \wedge, \vee, \neg \rangle$ described by

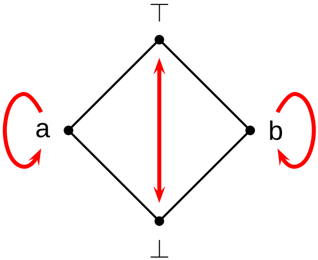


the procedure finds an algebra in $\mathrm{Adm}(\mathbf{D_4})$ isomorphic to $\mathbf{D_4} \times \mathbf{2}$ with $\mathbf{2} \in \mathbb{S}(\mathbf{F_{D_4}}(1))$, so $\mathbf{D_4}$ is almost structurally complete.

Other almost structurally complete algebras include the 3-element Łukasiewicz algebra and $\mathbf{S}_3^{\rightarrow}$ with an involutive negation.

Example: A Four Element Algebra

For the De Morgan lattice $\mathbf{D_4} = \langle\{\bot, \mathrm{a}, \mathrm{b}, \top\}, \wedge, \vee, \neg\rangle$ described by

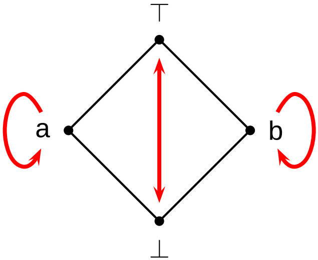


the procedure finds an algebra in $\mathrm{Adm}(\mathbf{D_4})$ isomorphic to $\mathbf{D_4} \times \mathbf{2}$ with $\mathbf{2} \in \mathbb{S}(\mathbf{F_{D_4}}(1))$, so $\mathbf{D_4}$ is almost structurally complete.

Other almost structurally complete algebras include the 3-element Łukasiewicz algebra and $\mathbf{S}_3^{\rightarrow}$ with an involutive negation.

Example: A Four Element Algebra

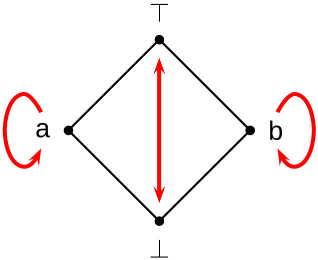
For the De Morgan lattice $\mathbf{D_4} = \langle\{\bot, \mathrm{a}, \mathrm{b}, \top\}, \wedge, \vee, \neg\rangle$ described by



the procedure finds an algebra in $\mathrm{Adm}(\mathbf{D_4})$ isomorphic to $\mathbf{D_4} \times \mathbf{2}$ with $\mathbf{2} \in \mathbb{S}(\mathbf{F_{D_4}}(1))$, so $\mathbf{D_4}$ is almost structurally complete.

Other almost structurally complete algebras include the 3-element Łukasiewicz algebra and $\mathbf{S_3^{\rightarrow}}$ with an involutive negation.

For the De Morgan lattice $\mathbf{D_4} = \langle \{\bot, \mathrm{a}, \mathrm{b}, \top\}, \wedge, \vee, \neg \rangle$ described by



the procedure finds an algebra in $\mathrm{Adm}(\mathbf{D_4})$ isomorphic to $\mathbf{D_4} \times \mathbf{2}$ with $\mathbf{2} \in \mathbb{S}(\mathbf{F_{D_4}}(1))$, so $\mathbf{D_4}$ is almost structurally complete.

Other almost structurally complete algebras include the 3-element Łukasiewicz algebra and $\mathbf{S}_\mathbf{3}^{\rightarrow}$ with an involutive negation.

Other Examples

- For the De Morgan algebra

$$\mathbf{D_4^b} = \langle \{\bot, \mathrm{a}, \mathrm{b}, \top\}, \wedge, \vee, \neg, \bot, \top \rangle$$

the procedure finds a smallest 10-element algebra in $\mathrm{Adm}(\mathbf{D_4})$.

- For the Kleene lattice and Kleene algebra

$$\mathbf{C_3} = \langle \{\top, \mathrm{a}, \bot\}, \wedge, \vee, \neg \rangle \quad \text{and} \quad \mathbf{C_3^b} = \langle \{\top, \mathrm{a}, \bot\}, \wedge, \vee, \neg, \bot, \top \rangle$$

the procedure finds smallest 4-element chains.

Other Examples

- For the De Morgan algebra

$$\mathbf{D_4^b} = \langle \{\bot, \mathrm{a}, \mathrm{b}, \top\}, \wedge, \vee, \neg, \bot, \top \rangle$$

  the procedure finds a smallest 10-element algebra in $\mathrm{Adm}(\mathbf{D_4})$.

- For the Kleene lattice and Kleene algebra

$$\mathbf{C_3} = \langle \{\top, \mathrm{a}, \bot\}, \wedge, \vee, \neg \rangle \quad \text{and} \quad \mathbf{C_3^b} = \langle \{\top, \mathrm{a}, \bot\}, \wedge, \vee, \neg, \bot, \top \rangle$$

  the procedure finds smallest 4-element chains.

Experiments in Admissibility

$\mathbf{A}$	$\lvert\mathbf{A}\rvert$	Quasivariety $\mathbb{Q}(\mathbf{A})$	Free algebra	$\lvert$Output Algebra$\rvert$
$\mathbf{Ł_3}$	3	algebras for Ł_3	$\lvert\mathbf{F_A}(1)\rvert = 12$	6
$\mathbf{Ł_3^{\to}}$	3	algebras for $\text{Ł}_3^{\to}$	$\lvert\mathbf{F_A}(2)\rvert = 40$	3
$\mathbf{B_1}$	3	Stone algebras	$\lvert\mathbf{F_A}(1)\rvert = 6$	3
$\mathbf{C_3^b}$	3	Kleene algebras	$\lvert\mathbf{F_A}(1)\rvert = 6$	4
$\mathbf{C_3}$	3	Kleene lattices	$\lvert\mathbf{F_A}(2)\rvert = 82$	4
$\mathbf{S_3^{\to\neg}}$	3	algebras for $\mathrm{RM}^{\to\neg}$	$\lvert\mathbf{F_A}(2)\rvert = 264$	6
$\mathbf{S_3^{\to}}$	3	algebras for $\mathrm{RM}^{\to}$	$\lvert\mathbf{F_A}(2)\rvert = 60$	3
$\mathbf{G_3}$	3	algebras for G_3	$\lvert\mathbf{F_A}(2)\rvert = 18$	3
$\mathbf{D_4}$	4	De Morgan lattices	$\lvert\mathbf{F_A}(2)\rvert = 166$	8
$\mathbf{D_4^b}$	4	De Morgan algebras	$\lvert\mathbf{F_A}(2)\rvert = 168$	10
$\mathbf{S_4^{\to\neg\mathrm{e}}}$	4	$\mathbb{Q}(\mathbf{S_4^{\to\neg\mathrm{e}}})$	$\lvert\mathbf{F_A}(1)\rvert = 18$	6
$\mathbf{B_2}$	5	$\mathbb{Q}(\mathbf{B_2})$	$\lvert\mathbf{F_A}(1)\rvert = 7$	5

A Last Technical Comment

Let $\mathcal{Q}$ be a $\mathcal{L}$-quasivariety.

- $\mathbf{A} \in \mathcal{Q}$ is called **exact** if it embeds into a free algebra of $\mathcal{Q}$.
- Given a finite set of $\mathcal{L}$-equations Γ over a finite set of variables X, define a congruence on $\mathbf{F}_{\mathcal{Q}}(X)$ by

$$[\varphi] \sim_\Gamma [\psi] \quad \text{iff} \quad \Gamma \models_{\mathcal{Q}} \varphi \approx \psi.$$

- If the **finitely presented algebra** $\mathbf{F}_{\mathcal{Q}}(X) \,/ \sim_\Gamma$ is exact, then

$$\Gamma \models_{\mathcal{Q}} \Delta \quad \text{iff} \quad \Gamma \mathrel{|\!\sim}_{\mathcal{Q}} \Delta.$$

- $\mathbf{A} \in \mathcal{Q}$ is **projective** if it is the *retract* of a free algebra of $\mathcal{Q}$ (i.e., embeds into and is the homomorphic image of a free algebra).

A Last Technical Comment

Let $\mathcal{Q}$ be a $\mathcal{L}$-quasivariety.

- $\mathbf{A} \in \mathcal{Q}$ is called **exact** if it embeds into a free algebra of $\mathcal{Q}$.
- Given a finite set of $\mathcal{L}$-equations Γ over a finite set of variables X, define a congruence on $\mathbf{F}_{\mathcal{Q}}(X)$ by

$$[\varphi] \sim_\Gamma [\psi] \quad \text{iff} \quad \Gamma \models_{\mathcal{Q}} \varphi \approx \psi.$$

- If the **finitely presented algebra** $\mathbf{F}_{\mathcal{Q}}(X) / \sim_\Gamma$ is exact, then

$$\Gamma \models_{\mathcal{Q}} \Delta \quad \text{iff} \quad \Gamma \mathrel{|\!\sim}_{\mathcal{Q}} \Delta.$$

- $\mathbf{A} \in \mathcal{Q}$ is **projective** if it is the *retract* of a free algebra of $\mathcal{Q}$ (i.e., embeds into and is the homomorphic image of a free algebra).

A Last Technical Comment

Let $\mathcal{Q}$ be a $\mathcal{L}$-quasivariety.

- $\mathbf{A} \in \mathcal{Q}$ is called **exact** if it embeds into a free algebra of $\mathcal{Q}$.
- Given a finite set of $\mathcal{L}$-equations Γ over a finite set of variables X, define a congruence on $\mathbf{F}_{\mathcal{Q}}(X)$ by

$$[\varphi] \sim_\Gamma [\psi] \quad \text{iff} \quad \Gamma \models_{\mathcal{Q}} \varphi \approx \psi.$$

- If the **finitely presented algebra $\mathbf{F}_{\mathcal{Q}}(X) \,/\, \sim_\Gamma$** is exact, then

$$\Gamma \models_{\mathcal{Q}} \Delta \quad \text{iff} \quad \Gamma \mathrel{|\!\sim}_{\mathcal{Q}} \Delta.$$

- $\mathbf{A} \in \mathcal{Q}$ is **projective** if it is the *retract* of a free algebra of $\mathcal{Q}$ (i.e., embeds into and is the homomorphic image of a free algebra).

A Last Technical Comment

Let $\mathcal{Q}$ be a $\mathcal{L}$-quasivariety.

- $\mathbf{A} \in \mathcal{Q}$ is called **exact** if it embeds into a free algebra of $\mathcal{Q}$.
- Given a finite set of $\mathcal{L}$-equations Γ over a finite set of variables X, define a congruence on $\mathbf{F}_{\mathcal{Q}}(X)$ by

$$[\varphi] \sim_\Gamma [\psi] \quad \text{iff} \quad \Gamma \models_{\mathcal{Q}} \varphi \approx \psi.$$

- If the **finitely presented algebra** $\mathbf{F}_{\mathcal{Q}}(X) \,/\, \sim_\Gamma$ is exact, then

$$\Gamma \models_{\mathcal{Q}} \Delta \quad \text{iff} \quad \Gamma \mathrel{|\!\sim}_{\mathcal{Q}} \Delta.$$

- $\mathbf{A} \in \mathcal{Q}$ is **projective** if it is the *retract* of a free algebra of $\mathcal{Q}$ (i.e., embeds into and is the homomorphic image of a free algebra).

A Last Technical Comment

Let $\mathcal{Q}$ be a $\mathcal{L}$-quasivariety.

- $\mathbf{A} \in \mathcal{Q}$ is called **exact** if it embeds into a free algebra of $\mathcal{Q}$.
- Given a finite set of $\mathcal{L}$-equations Γ over a finite set of variables X, define a congruence on $\mathbf{F}_{\mathcal{Q}}(X)$ by

$$[\varphi] \sim_\Gamma [\psi] \quad \text{iff} \quad \Gamma \models_{\mathcal{Q}} \varphi \approx \psi.$$

- If the **finitely presented algebra** $\mathbf{F}_{\mathcal{Q}}(X) \,/ \sim_\Gamma$ is exact, then

$$\Gamma \models_{\mathcal{Q}} \Delta \quad \text{iff} \quad \Gamma \mathrel{|\!\sim}_{\mathcal{Q}} \Delta.$$

- $\mathbf{A} \in \mathcal{Q}$ is **projective** if it is the *retract* of a free algebra of $\mathcal{Q}$ (i.e., embeds into and is the homomorphic image of a free algebra).

A Last Technical Comment

Let $\mathcal{Q}$ be a $\mathcal{L}$-quasivariety.

- $\mathbf{A} \in \mathcal{Q}$ is called **exact** if it embeds into a free algebra of $\mathcal{Q}$.
- Given a finite set of $\mathcal{L}$-equations Γ over a finite set of variables X, define a congruence on $\mathbf{F}_{\mathcal{Q}}(X)$ by

$$[\varphi] \sim_{\Gamma} [\psi] \quad \text{iff} \quad \Gamma \models_{\mathcal{Q}} \varphi \approx \psi.$$

- If the **finitely presented algebra** $\mathbf{F}_{\mathcal{Q}}(X) / \sim_{\Gamma}$ is exact, then

$$\Gamma \models_{\mathcal{Q}} \Delta \quad \text{iff} \quad \Gamma \mathrel{|\!\!\sim}_{\mathcal{Q}} \Delta.$$

- $\mathbf{A} \in \mathcal{Q}$ is **projective** if it is the *retract* of a free algebra of $\mathcal{Q}$ (i.e., embeds into and is the homomorphic image of a free algebra).

A Last Technical Comment

Let $\mathcal{Q}$ be a $\mathcal{L}$-quasivariety.

- $\mathbf{A} \in \mathcal{Q}$ is called **exact** if it embeds into a free algebra of $\mathcal{Q}$.
- Given a finite set of $\mathcal{L}$-equations Γ over a finite set of variables X, define a congruence on $\mathbf{F}_{\mathcal{Q}}(X)$ by

$$[\varphi] \sim_{\Gamma} [\psi] \quad \text{iff} \quad \Gamma \models_{\mathcal{Q}} \varphi \approx \psi.$$

- If the **finitely presented algebra** $\mathbf{F}_{\mathcal{Q}}(X) / \sim_{\Gamma}$ is exact, then

$$\Gamma \models_{\mathcal{Q}} \Delta \quad \text{iff} \quad \Gamma \mathrel{|\!\!\sim}_{\mathcal{Q}} \Delta.$$

- $\mathbf{A} \in \mathcal{Q}$ is **projective** if it is the *retract* of a free algebra of $\mathcal{Q}$ (i.e., embeds into and is the homomorphic image of a free algebra).

Part III

Unification and Admissibility

The Idea

Admissibility is a more general problem than unification, but in certain cases we can “reduce” admissibility questions to unification questions.

For convenience, we give a “logical” account of unification.

In particular, we consider a **modal** or **intermediate logic** L and can therefore treat formulas rather than finite sets of formulas.

S. Ghilardi. Unification in intuitionistic logic.
Journal of Symbolic Logic 64(2):859–880, 1999.

S. Ghilard. Best Solving Modal Equations.
Annals of Pure and Applied Logic 102(3):184–198, 2000.

The Idea

Admissibility is a more general problem than unification, but in certain cases we can "reduce" admissibility questions to unification questions.

For convenience, we give a "logical" account of unification.

In particular, we consider a **modal** or **intermediate logic** L and can therefore treat formulas rather than finite sets of formulas.

S. Ghilardi. Unification in intuitionistic logic.
Journal of Symbolic Logic 64(2):859–880, 1999.

S. Ghilard. Best Solving Modal Equations.
Annals of Pure and Applied Logic 102(3):184–198, 2000.

The Idea

Admissibility is a more general problem than unification, but in certain cases we can “reduce” admissibility questions to unification questions.

For convenience, we give a “logical” account of unification.

In particular, we consider a **modal** or **intermediate logic** L and can therefore treat formulas rather than finite sets of formulas.

S. Ghilardi. Unification in intuitionistic logic.
Journal of Symbolic Logic 64(2):859–880, 1999.

S. Ghilard. Best Solving Modal Equations.
Annals of Pure and Applied Logic 102(3):184–198, 2000.

The Idea

Admissibility is a more general problem than unification, but in certain cases we can “reduce” admissibility questions to unification questions.

For convenience, we give a “logical” account of unification.

In particular, we consider a **modal** or **intermediate logic** L and can therefore treat formulas rather than finite sets of formulas.

S. Ghilardi. Unification in intuitionistic logic.
Journal of Symbolic Logic 64(2):859–880, 1999.

S. Ghilard. Best Solving Modal Equations.
Annals of Pure and Applied Logic 102(3):184–198, 2000.

L-Unification

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \iff \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \iff$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L-**unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

L-Unification

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L-**unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \iff \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \iff$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L-**unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \iff \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \iff$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L**-unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

L-Unification

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \iff \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \iff$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L**-unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

L-Unification

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \iff \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \iff$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L-**unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

L-Unification

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L**-unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

L-Unification

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L**-unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

L-Unification

Given substitutions σ_1, σ_2:

- $\sigma_1 =_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad \vdash_{\mathrm{L}} \sigma_1 p \leftrightarrow \sigma_2 p$ for all variables p
- $\sigma_1 \leq_{\mathrm{L}} \sigma_2 \quad \Longleftrightarrow \quad$ there is a substitution σ such that $\sigma_2 =_{\mathrm{L}} \sigma\sigma_1$.

Recall that a substitution σ is an L**-unifier** of a formula φ if $\vdash_{\mathrm{L}} \sigma(\varphi)$.

A set $\mathcal{C}$ of L-unifiers of φ is called **complete** if

- for any L-unifier σ of φ, there exists $\sigma' \in \mathcal{C}$ such that $\sigma' \leq_{\mathrm{L}} \sigma$.

$\mathcal{C}$ is **minimal** if also

- for any $\sigma_1, \sigma_2 \in \mathcal{C}$, if $\sigma_1 \leq_{\mathrm{L}} \sigma_2$, then $\sigma_1 = \sigma_2$.

If $\mathcal{C} = \{\sigma\}$, then σ is a **most general unifier** of φ.

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

- with $|\mathcal{C}| = 1$, then φ has type **unitary**
- with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**
- with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

with $|\mathcal{C}| = 1$, then φ has type **unitary**

with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**

with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

- with $|\mathcal{C}| = 1$, then φ has type **unitary**
- with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**
- with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

- with $|\mathcal{C}| = 1$, then φ has type **unitary**
- with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**
- with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

with $|\mathcal{C}| = 1$, then φ has type **unitary**

with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**

with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

with $|\mathcal{C}| = 1$, then φ has type **unitary**

with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**

with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

with $|\mathcal{C}| = 1$, then φ has type **unitary**

with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**

with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

with $|\mathcal{C}| = 1$, then φ has type **unitary**

with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**

with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

with $|\mathcal{C}| = 1$, then φ has type **unitary**

with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**

with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

with $|\mathcal{C}| = 1$, then φ has type **unitary**

with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**

with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Unification Type

If there exists a minimal complete set of L-unifiers $\mathcal{C}$ for φ

- with $|\mathcal{C}| = 1$, then φ has type **unitary**
- with $|\mathcal{C}|$ finite and $|\mathcal{C}| \neq 1$, then φ has type **finitary**
- with $|\mathcal{C}|$ infinite, then φ has type **infinitary**.

Otherwise, φ has type **nullary**.

The **unification type** of L is the maximal type of φ as ranked by

$$\text{unitary} < \text{finitary} < \text{infinitary} < \text{nullary}.$$

E.g., CPC unitary; IPC finitary (Rozière 1995); $\mathrm{K4}$ finitary (Ghilardi 2000); K nullary (Jeřábek 2011); Ł nullary (Marra & Spada 2011).

Recall that a rule $\{\varphi\} \rhd \Delta$ is L**-admissible**, if every L-unifier of φ is an L-unifier of some formula in Δ.

On the one hand:

$$\varphi \text{ is L-unifiable} \quad \Longleftrightarrow \quad \{\varphi\} \rhd \emptyset \ \text{ is not L-admissible.}$$

But also, if $\mathcal{C}$ is a **complete set of** L**-unifiers** for φ, then

$$\{\varphi\} \rhd \Delta \ \text{ is L-admissible} \quad \Longleftrightarrow \quad \begin{array}{l}\text{each } \sigma \in \mathcal{C} \text{ is an L-unifier} \\ \text{of some formula in } \Delta.\end{array}$$

So if L is **decidable** and **finitary** and we can effectively find finite complete sets of L-unifiers, then L-admissibility is decidable.

Recall that a rule $\{\varphi\} \triangleright \Delta$ is L**-admissible**, if every L-unifier of φ is an L-unifier of some formula in Δ.

On the one hand:

$$\varphi \text{ is L-unifiable} \quad \Longleftrightarrow \quad \{\varphi\} \triangleright \emptyset \text{ is not L-admissible.}$$

But also, if $\mathcal{C}$ is a **complete set of** L**-unifiers** for φ, then

$$\{\varphi\} \triangleright \Delta \text{ is L-admissible} \quad \Longleftrightarrow \quad \begin{array}{l}\text{each } \sigma \in \mathcal{C} \text{ is an L-unifier}\\ \text{of some formula in } \Delta.\end{array}$$

So if L is **decidable** and **finitary** and we can effectively find finite complete sets of L-unifiers, then L-admissibility is decidable.

Recall that a rule $\{\varphi\} \rhd \Delta$ is L-**admissible**, if every L-unifier of φ is an L-unifier of some formula in Δ.

On the one hand:

$$\varphi \text{ is L-unifiable} \quad \Longleftrightarrow \quad \{\varphi\} \rhd \emptyset \ \text{ is not L-admissible.}$$

But also, if $\mathcal{C}$ is a **complete set of** L**-unifiers** for φ, then

$$\{\varphi\} \rhd \Delta \ \text{ is L-admissible} \quad \Longleftrightarrow \quad \begin{array}{l}\text{each } \sigma \in \mathcal{C} \text{ is an L-unifier} \\ \text{of some formula in } \Delta.\end{array}$$

So if L is **decidable** and **finitary** and we can effectively find finite complete sets of L-unifiers, then L-admissibility is decidable.

Unification and Admissibility

Recall that a rule $\{\varphi\} \rhd \Delta$ is L-**admissible**, if every L-unifier of φ is an L-unifier of some formula in Δ.

On the one hand:

$$\varphi \text{ is L-unifiable} \quad \iff \quad \{\varphi\} \rhd \emptyset \text{ is not L-admissible.}$$

But also, if $\mathcal{C}$ is a **complete set of** L**-unifiers** for φ, then

$$\{\varphi\} \rhd \Delta \text{ is L-admissible} \quad \iff \quad \begin{array}{l}\text{each } \sigma \in \mathcal{C} \text{ is an L-unifier} \\ \text{of some formula in } \Delta.\end{array}$$

So if L is **decidable** and **finitary** and we can effectively find finite complete sets of L-unifiers, then L-admissibility is decidable.

Recall that a rule $\{\varphi\} \rhd \Delta$ is L-**admissible**, if every L-unifier of φ is an L-unifier of some formula in Δ.

On the one hand:

$$\varphi \text{ is L-unifiable} \quad \Longleftrightarrow \quad \{\varphi\} \rhd \emptyset \ \text{ is not L-admissible.}$$

But also, if $\mathcal{C}$ is a **complete set of** L**-unifiers** for φ, then

$$\{\varphi\} \rhd \Delta \ \text{ is L-admissible} \quad \Longleftrightarrow \quad \begin{array}{l}\text{each } \sigma \in \mathcal{C} \text{ is an L-unifier} \\ \text{of some formula in } \Delta.\end{array}$$

So if L is **decidable** and **finitary** and we can effectively find finite complete sets of L-unifiers, then L-admissibility is decidable.

Projective Formulas

Definition

An L-unifier σ of an $\mathcal{L}$-formula φ is called **projective** if:

$$\{\varphi\} \vdash_{\mathrm{L}} \sigma\psi \leftrightarrow \psi \qquad \text{for all } \psi \in \mathrm{Fm}_{\mathcal{L}}.$$

In this case, φ is said to be L-**projective**.

Lemma

If φ is L-projective, then $\{\varphi\} \mathrel{\vert\!\sim}_{\mathrm{L}} \Delta$ iff $\{\varphi\} \vdash_{\mathrm{L}} \Delta$.

Note that a projective L-unifier σ of φ is a **most general unifier** of φ; i.e., if φ is L-projective, then it has **unitary** unification type.

Projective Formulas

Definition

An L-unifier σ of an $\mathcal{L}$-formula φ is called **projective** if:

$$\{\varphi\} \vdash_{\mathrm{L}} \sigma\psi \leftrightarrow \psi \qquad \text{for all } \psi \in \mathrm{Fm}_{\mathcal{L}}.$$

In this case, φ is said to be L**-projective**.

Lemma

If φ is L-projective, then $\{\varphi\} \mid\!\sim_{\mathrm{L}} \Delta$ iff $\{\varphi\} \vdash_{\mathrm{L}} \Delta$.

Note that a projective L-unifier σ of φ is a **most general unifier** of φ; i.e., if φ is L-projective, then it has **unitary** unification type.

Projective Formulas

Definition

An L-unifier σ of an $\mathcal{L}$-formula φ is called **projective** if:

$$\{\varphi\} \vdash_{\mathrm{L}} \sigma\psi \leftrightarrow \psi \qquad \text{for all } \psi \in \mathrm{Fm}_{\mathcal{L}}.$$

In this case, φ is said to be L**-projective**.

Lemma

If φ is L-projective, then $\{\varphi\} \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta$ iff $\{\varphi\} \vdash_{\mathrm{L}} \Delta$.

Note that a projective L-unifier σ of φ is a **most general unifier** of φ; i.e., if φ is L-projective, then it has **unitary** unification type.

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \;\text{ for all }\; \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{aligned}
\sigma \in \mathcal{C} \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi)\\
&\Longrightarrow \quad \sigma \text{ is an L-unifier of } \varphi\\
\sigma \text{ is an L-unifier of } \varphi \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi)\\
&\Longrightarrow \quad \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C}
\end{aligned}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \;\text{ for all }\; \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \ \text{ for all } \ \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{array}{rcl}
\sigma \in \mathcal{C} & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
 & \Longrightarrow & \sigma \text{ is an L-unifier of } \varphi \\
\sigma \text{ is an L-unifier of } \varphi & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
 & \Longrightarrow & \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C}
\end{array}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \ \text{ for all } \ \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \text{ for all } \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{aligned} \sigma \in \mathcal{C} \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ &\Longrightarrow \quad \sigma \text{ is an L-unifier of } \varphi \\ \sigma \text{ is an L-unifier of } \varphi \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ &\Longrightarrow \quad \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C} \end{aligned}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \text{ for all } \psi \in \Pi(\varphi).$$

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \;\text{ for all }\; \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{aligned}
\sigma \in \mathcal{C} \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
&\Longrightarrow \quad \sigma \text{ is an L-unifier of } \varphi \\
\sigma \text{ is an L-unifier of } \varphi \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
&\Longrightarrow \quad \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C}
\end{aligned}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \;\text{ for all }\; \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \ \text{ for all } \ \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{array}{rcl} \sigma \in \mathcal{C} & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ & \Longrightarrow & \sigma \text{ is an L-unifier of } \varphi \\ \sigma \text{ is an L-unifier of } \varphi & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ & \Longrightarrow & \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C} \end{array}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \ \text{ for all } \ \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \ \text{ for all } \ \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{array}{rcl} \sigma \in \mathcal{C} & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ & \Longrightarrow & \sigma \text{ is an L-unifier of } \varphi \end{array}$$

$$\begin{array}{rcl} \sigma \text{ is an L-unifier of } \varphi & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ & \Longrightarrow & \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C} \end{array}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \ \text{ for all } \ \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \ \text{ for all } \ \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{array}{rcl}
\sigma \in \mathcal{C} & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
 & \Longrightarrow & \sigma \text{ is an L-unifier of } \varphi \\
\sigma \text{ is an L-unifier of } \varphi & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
 & \Longrightarrow & \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C}
\end{array}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \ \text{ for all } \ \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \ \text{ for all } \ \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{array}{rcl} \sigma \in \mathcal{C} & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ & \Longrightarrow & \sigma \text{ is an L-unifier of } \varphi \\ \sigma \text{ is an L-unifier of } \varphi & \Longrightarrow & \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\ & \Longrightarrow & \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C} \end{array}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \ \text{ for all } \ \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \text{ for all } \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{aligned}
\sigma \in \mathcal{C} \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
&\Longrightarrow \quad \sigma \text{ is an L-unifier of } \varphi \\
\sigma \text{ is an L-unifier of } \varphi \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
&\Longrightarrow \quad \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C}
\end{aligned}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \text{ for all } \psi \in \Pi(\varphi).$$

Projective Axiomatizations

A **projective approximation** of an $\mathcal{L}$-formula φ is a finite set $\Pi(\varphi)$ of L-projective formulas satisfying

$$\{\psi\} \vdash_{\mathrm{L}} \varphi \text{ for all } \psi \in \Pi(\varphi) \qquad \text{and} \qquad \{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Pi(\varphi).$$

For any set $\mathcal{C}$ of L-projective unifiers of members of $\Pi(\varphi)$:

$$\begin{aligned}
\sigma \in \mathcal{C} \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
&\Longrightarrow \quad \sigma \text{ is an L-unifier of } \varphi \\
\sigma \text{ is an L-unifier of } \varphi \quad &\Longrightarrow \quad \sigma \text{ is an L-unifier of some } \psi \in \Pi(\varphi) \\
&\Longrightarrow \quad \sigma' \leq \sigma \text{ for some } \sigma' \in \mathcal{C}
\end{aligned}$$

I.e., $\mathcal{C}$ is a complete set of L-unifiers for φ. Moreover:

$$\{\varphi\} \mathrel{|\!\sim}_{\mathrm{L}} \Delta \qquad \text{iff} \qquad \{\psi\} \vdash_{\mathrm{L}} \Delta \text{ for all } \psi \in \Pi(\varphi).$$

Projective Formulas in IPC

A class of Kripke models $\mathcal{K}$ has the **extension property** if given $K_1, \ldots, K_n \in \mathcal{K}$, there is a Kripke model in $\mathcal{K}$ obtained by attaching one new node below all nodes in $K_1, \ldots, K_n$.

Theorem (Ghilardi 1999)

A formula is IPC*-projective iff its class of finite Kripke models has the extension property.*

Theorem (Ghilardi 1999)

An IPC*-projective approximation can be found effectively for any formula.*

Corollary (Ghilardi 1999)

IPC *has finitary unification type.*

Projective Formulas in IPC

A class of Kripke models $\mathcal{K}$ has the **extension property** if given $K_1, \ldots, K_n \in \mathcal{K}$, there is a Kripke model in $\mathcal{K}$ obtained by attaching one new node below all nodes in $K_1, \ldots, K_n$.

Theorem (Ghilardi 1999)

A formula is IPC*-projective iff its class of finite Kripke models has the extension property.*

Theorem (Ghilardi 1999)

An IPC*-projective approximation can be found effectively for any formula.*

Corollary (Ghilardi 1999)

IPC *has finitary unification type.*

Projective Formulas in IPC

A class of Kripke models $\mathcal{K}$ has the **extension property** if given $K_1, \ldots, K_n \in \mathcal{K}$, there is a Kripke model in $\mathcal{K}$ obtained by attaching one new node below all nodes in $K_1, \ldots, K_n$.

Theorem (Ghilardi 1999)

A formula is IPC*-projective iff its class of finite Kripke models has the extension property.*

Theorem (Ghilardi 1999)

An IPC*-projective approximation can be found effectively for any formula.*

Corollary (Ghilardi 1999)

IPC *has finitary unification type.*

Projective Formulas in IPC

A class of Kripke models $\mathcal{K}$ has the **extension property** if given $K_1, \ldots, K_n \in \mathcal{K}$, there is a Kripke model in $\mathcal{K}$ obtained by attaching one new node below all nodes in $K_1, \ldots, K_n$.

Theorem (Ghilardi 1999)

A formula is IPC*-projective iff its class of finite Kripke models has the extension property.*

Theorem (Ghilardi 1999)

An IPC*-projective approximation can be found effectively for any formula.*

Corollary (Ghilardi 1999)

IPC *has finitary unification type.*

Axiomatizing Admissibility

For a logic L, we are interested in finding a set of rules that "axiomatizes" (over L) the admissible rules of L.

Definition

A **basis** for $\mathrel{|\!\sim}_{\mathrm{L}}$ over L is a set B of rules such that $\mathrel{|\!\sim}_{\mathrm{L}}$ is the smallest m-logic extending $B \cup \mathrm{L}$.

We can also consider bases for "single-conclusion" $\mathrel{|\!\sim}_{\mathrm{L}}$.

Axiomatizing Admissibility

For a logic L, we are interested in finding a set of rules that "axiomatizes" (over L) the admissible rules of L.

Definition

A **basis** for $\mid\!\sim_{\mathrm{L}}$ over L is a set B of rules such that $\mid\!\sim_{\mathrm{L}}$ is the smallest m-logic extending $B \cup \mathrm{L}$.

We can also consider bases for "single-conclusion" $\mid\!\sim_{\mathrm{L}}$.

For a logic L, we are interested in finding a set of rules that "axiomatizes" (over L) the admissible rules of L.

Definition

A **basis** for $\mathrel{|\!\!\sim}_{\mathrm{L}}$ over L is a set B of rules such that $\mathrel{|\!\!\sim}_{\mathrm{L}}$ is the smallest m-logic extending $B \cup \mathrm{L}$.

We can also consider bases for "single-conclusion" $\mathrel{|\!\!\sim}_{\mathrm{L}}$.

Bases

Iemhoff and Rozière proved independently that the "Visser rules"

$$\{\bigwedge_{i=1}^{n}(p_i \to q_i) \to (p_{n+1} \vee p_{n+2})\} \rhd \bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(p_i \to q_i) \to p_j) \quad n = 2, 3, \ldots$$

plus the disjunction property provide a basis for admissibility in IPC.

P. Rozière. Regles admissibles en calcul propositionnel intuitionniste. Ph.D. thesis, Université Paris VII, 1992.

R. Iemhoff. On the admissible rules of intuitionistic propositional logic. *Journal of Symbolic Logic* 66(1):281–294, 2001.

Iemhoff has also shown that the Visser rules provide a basis for certain **intermediate logics**, and Jeřàbek has given bases for a wide range of **transitive modal logics** and **Łukasiewicz logics**.

Bases

Iemhoff and Rozière proved independently that the "Visser rules"

$$\{\bigwedge_{i=1}^{n}(p_i \to q_i) \to (p_{n+1} \lor p_{n+2})\} \rhd \bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(p_i \to q_i) \to p_j) \quad n = 2, 3, \ldots$$

plus the disjunction property provide a basis for admissibility in IPC.

P. Rozière. Regles admissibles en calcul propositionnel intuitionniste. Ph.D. thesis, Université Paris VII, 1992.

R. Iemhoff. On the admissible rules of intuitionistic propositional logic. *Journal of Symbolic Logic* 66(1):281–294, 2001.

Iemhoff has also shown that the Visser rules provide a basis for certain **intermediate logics**, and Jeřàbek has given bases for a wide range of **transitive modal logics** and **Łukasiewicz logics**.

A Last Example

The following "Wroński rules" ($n \in \mathbb{N}$):

$$(\mathrm{W}_n) \quad \{p_1 \to \ldots \to p_{n-1} \to \neg p_n\} \rhd \{\neg\neg p_1 \to p_1, \ldots, \neg\neg p_n \to p_n\}$$

provide a basis for the admissible rules of the implication-negation fragment of any intermediate logic.

P. Cintula and G. Metcalfe. Admissible rules in the implication-negation fragment of intuitionistic logic. *Annals of Pure and Applied Logic* 162(2): 162–171 (2010).

Part IV

Proof Theory for Admissible Rules

The Idea

Recall that a rule $\Gamma \rhd \Delta$ is L**-admissible** if every L-unifier of Γ is an L-unifier of some formula in Δ.

Since the sets of admissible rules of IPC and extensible modal logics are **recursively enumerable**, they are also **decidable**.

However, we would also like to have analytic "Gentzen-style" calculi for deciding the admissibility of rules in these logics.

Instead of treating sequents as the proof objects of a calculus, we deal with **sequent rules**.

R. Iemhoff and G. Metcalfe. Proof theory of admissible rules.
Annals of Pure and Applied Logic 159(1–2):171–186, 2009.

The Idea

Recall that a rule $\Gamma \rhd \Delta$ is L**-admissible** if every L-unifier of Γ is an L-unifier of some formula in Δ.

Since the sets of admissible rules of IPC and extensible modal logics are **recursively enumerable**, they are also **decidable**.

However, we would also like to have analytic "Gentzen-style" calculi for deciding the admissibility of rules in these logics.

Instead of treating sequents as the proof objects of a calculus, we deal with **sequent rules**.

R. Iemhoff and G. Metcalfe. Proof theory of admissible rules.
Annals of Pure and Applied Logic 159(1–2):171–186, 2009.

The Idea

Recall that a rule $\Gamma \rhd \Delta$ is L**-admissible** if every L-unifier of Γ is an L-unifier of some formula in Δ.

Since the sets of admissible rules of IPC and extensible modal logics are **recursively enumerable**, they are also **decidable**.

However, we would also like to have analytic "Gentzen-style" calculi for deciding the admissibility of rules in these logics.

Instead of treating sequents as the proof objects of a calculus, we deal with **sequent rules**.

R. Iemhoff and G. Metcalfe. Proof theory of admissible rules. *Annals of Pure and Applied Logic* 159(1–2):171–186, 2009.

The Idea

Recall that a rule $\Gamma \rhd \Delta$ is L**-admissible** if every L-unifier of Γ is an L-unifier of some formula in Δ.

Since the sets of admissible rules of IPC and extensible modal logics are **recursively enumerable**, they are also **decidable**.

However, we would also like to have analytic "Gentzen-style" calculi for deciding the admissibility of rules in these logics.

Instead of treating sequents as the proof objects of a calculus, we deal with **sequent rules**.

R. Iemhoff and G. Metcalfe. Proof theory of admissible rules. *Annals of Pure and Applied Logic* 159(1–2):171–186, 2009.

A **sequent** S is an ordered pair of finite sets of formulas, written

$$\Gamma \Rightarrow \Delta,$$

and interpreted by (with $\bigwedge \emptyset = \top$ and $\bigvee \emptyset = \bot$)

$$\mathcal{I}(S) = \bigwedge \Gamma \to \bigvee \Delta.$$

We will use $\mathcal{G}$ and $\mathcal{H}$ to stand for **sets of sequents**.

Sequents for IPC

A **sequent** S is an ordered pair of finite sets of formulas, written

$$\Gamma \Rightarrow \Delta,$$

and interpreted by (with $\bigwedge \emptyset = \top$ and $\bigvee \emptyset = \bot$)

$$\mathcal{I}(S) = \bigwedge \Gamma \to \bigvee \Delta.$$

We will use $\mathcal{G}$ and $\mathcal{H}$ to stand for **sets of sequents**.

Sequents for IPC

A **sequent** S is an ordered pair of finite sets of formulas, written

$$\Gamma \Rightarrow \Delta,$$

and interpreted by (with $\bigwedge \emptyset = \top$ and $\bigvee \emptyset = \bot$)

$$\mathcal{I}(S) = \bigwedge \Gamma \to \bigvee \Delta.$$

We will use $\mathcal{G}$ and $\mathcal{H}$ to stand for **sets of sequents**.

Sequent Rules

A **sequent rule** is an ordered pair of finite sets of sequents, written

$$\mathcal{G} \rhd \mathcal{H},$$

and called

- IPC-**derivable** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \vdash_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$
- IPC-**admissible** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \mathrel{|\!\sim}_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$.

For example, the sequent rule

$$\{(\neg p \Rightarrow q, r)\} \rhd \{(\neg p \Rightarrow q), (\neg p \Rightarrow r)\}$$

is IPC-admissible, since

$$\{\neg p \to (q \lor r)\} \mathrel{|\!\sim}_{\mathrm{IPC}} \{\neg p \to q, \neg p \to r\}.$$

Sequent Rules

A **sequent rule** is an ordered pair of finite sets of sequents, written

$$\mathcal{G} \rhd \mathcal{H},$$

and called

- IPC-**derivable** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \vdash_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$
- IPC-**admissible** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \mid\!\sim_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$.

For example, the sequent rule

$$\{(\neg p \Rightarrow q, r)\} \rhd \{(\neg p \Rightarrow q), (\neg p \Rightarrow r)\}$$

is IPC-admissible, since

$$\{\neg p \to (q \lor r)\} \mid\!\sim_{\mathrm{IPC}} \{\neg p \to q, \neg p \to r\}.$$

A **sequent rule** is an ordered pair of finite sets of sequents, written

$$\mathcal{G} \rhd \mathcal{H},$$

and called

- IPC-**derivable** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \vdash_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$
- IPC-**admissible** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \mathrel{|\!\!\sim}_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$.

For example, the sequent rule

$$\{(\neg p \Rightarrow q, r)\} \rhd \{(\neg p \Rightarrow q), (\neg p \Rightarrow r)\}$$

is IPC-admissible, since

$$\{\neg p \to (q \vee r)\} \mathrel{|\!\!\sim}_{\mathrm{IPC}} \{\neg p \to q, \neg p \to r\}.$$

Sequent Rules

A **sequent rule** is an ordered pair of finite sets of sequents, written

$$\mathcal{G} \rhd \mathcal{H},$$

and called

- IPC**-derivable** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \vdash_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$
- IPC**-admissible** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \mathrel{|\!\!\sim}_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$.

For example, the sequent rule

$$\{(\neg p \Rightarrow q, r)\} \rhd \{(\neg p \Rightarrow q), (\neg p \Rightarrow r)\}$$

is IPC-admissible, since

$$\{\neg p \to (q \lor r)\} \mathrel{|\!\!\sim}_{\mathrm{IPC}} \{\neg p \to q, \neg p \to r\}.$$

Sequent Rules

A **sequent rule** is an ordered pair of finite sets of sequents, written

$$\mathcal{G} \rhd \mathcal{H},$$

and called

- IPC-**derivable** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \vdash_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$
- IPC-**admissible** if $\{\mathcal{I}(S) \mid S \in \mathcal{G}\} \mathrel{|\!\!\sim}_{\mathrm{IPC}} \{\mathcal{I}(S) \mid S \in \mathcal{H}\}$.

For example, the sequent rule

$$\{(\neg p \Rightarrow q, r)\} \rhd \{(\neg p \Rightarrow q), (\neg p \Rightarrow r)\}$$

is IPC-admissible, since

$$\{\neg p \to (q \vee r)\} \mathrel{|\!\!\sim}_{\mathrm{IPC}} \{\neg p \to q, \neg p \to r\}.$$

Our **proof system** GAMI for admissibility in IPC treats *sequent rules* $\mathcal{G} \triangleright \mathcal{H}$ as proof objects, and consists of

- right rules
- left rules
- structural rules
- a Visser rule.

Right Rules

$$\frac{}{\mathcal{G} \rhd (\Gamma, \varphi \Rightarrow \varphi, \Delta), \mathcal{H}} \text{ (ID)}$$

$$\frac{}{\mathcal{G} \rhd (\Gamma, \bot \Rightarrow \Delta), \mathcal{H}} \ \rhd(\bot\Rightarrow) \qquad \frac{\mathcal{G} \rhd (\Gamma \Rightarrow \Delta), \mathcal{H}}{\mathcal{G} \rhd (\Gamma \Rightarrow \bot, \Delta), \mathcal{H}} \ \rhd(\Rightarrow\bot)$$

$$\frac{\mathcal{G} \rhd (\Gamma \Rightarrow \varphi, \Delta), \mathcal{H} \quad \mathcal{G} \rhd (\Gamma \Rightarrow \psi, \Delta), \mathcal{H}}{\mathcal{G} \rhd (\Gamma \Rightarrow \varphi \wedge \psi, \Delta), \mathcal{H}} \ \rhd(\Rightarrow\wedge) \qquad \frac{\mathcal{G} \rhd (\Gamma, \varphi, \psi \Rightarrow \Delta), \mathcal{H}}{\mathcal{G} \rhd (\Gamma, \varphi \wedge \psi \Rightarrow \Delta), \mathcal{H}} \ \rhd(\wedge\Rightarrow)$$

$$\frac{\mathcal{G} \rhd (\Gamma, \varphi \Rightarrow \Delta), \mathcal{H} \quad \mathcal{G} \rhd (\Gamma, \psi \Rightarrow \Delta), \mathcal{H}}{\mathcal{G} \rhd (\Gamma, \varphi \vee \psi \Rightarrow \Delta), \mathcal{H}} \ \rhd(\vee\Rightarrow) \qquad \frac{\mathcal{G} \rhd (\Gamma \Rightarrow \varphi, \psi, \Delta), \mathcal{H}}{\mathcal{G} \rhd (\Gamma \Rightarrow \varphi \vee \psi, \Delta), \mathcal{H}} \ \rhd(\Rightarrow\vee)$$

$$\frac{\mathcal{G} \rhd (\Gamma \Rightarrow \varphi, \Delta), \mathcal{H} \quad \mathcal{G} \rhd (\Gamma, \psi \Rightarrow \Delta), \mathcal{H}}{\mathcal{G} \rhd (\Gamma, \varphi \to \psi \Rightarrow \Delta), \mathcal{H}} \ \rhd(\to\Rightarrow) \qquad \frac{\mathcal{G} \rhd (\Gamma, \varphi \Rightarrow \psi), \mathcal{H}}{\mathcal{G} \rhd (\Gamma \Rightarrow \varphi \to \psi, \Delta), \mathcal{H}} \ \rhd(\Rightarrow\to)$$

Left Rules

$$\frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \bot \Rightarrow \Delta) \rhd \mathcal{H}} \ (\bot\Rightarrow)_\rhd \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \bot, \Delta) \rhd \mathcal{H}} \ (\Rightarrow\bot)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma, \varphi, \psi \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \wedge \psi \Rightarrow \Delta) \rhd \mathcal{H}} \ (\wedge\Rightarrow)_\rhd \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \Delta), (\Gamma \Rightarrow \psi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \wedge \psi, \Delta) \rhd \mathcal{H}} \ (\Rightarrow\wedge)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \psi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \vee \psi, \Delta) \rhd \mathcal{H}} \ (\Rightarrow\vee)_\rhd \qquad \frac{\mathcal{G}, (\Gamma, \varphi \Rightarrow \Delta), (\Gamma, \psi \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \vee \psi \Rightarrow \Delta) \rhd \mathcal{H}} \ (\vee\Rightarrow)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma, \psi \Rightarrow \Delta), (\Gamma, \varphi \to \psi \Rightarrow \varphi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \rhd \mathcal{H}} \ (\to)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow p, \Delta), (p, \varphi \Rightarrow \psi) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \to \psi, \Delta) \rhd \mathcal{H}} \ (\Rightarrow\to)_\rhd \qquad p \text{ new}$$

$$\frac{\mathcal{G}, (\Gamma, p \to q \Rightarrow \Delta), (p \Rightarrow \varphi), (\psi \Rightarrow q) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \rhd \mathcal{H}} \ (\to\Rightarrow)_\rhd \qquad p, q \text{ new}$$

Left Rules

$$\frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \bot \Rightarrow \Delta) \rhd \mathcal{H}} \;(\bot\Rightarrow)_\rhd \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \bot, \Delta) \rhd \mathcal{H}} \;(\Rightarrow\bot)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma, \varphi, \psi \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \wedge \psi \Rightarrow \Delta) \rhd \mathcal{H}} \;(\wedge\Rightarrow)_\rhd \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \Delta), (\Gamma \Rightarrow \psi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \wedge \psi, \Delta) \rhd \mathcal{H}} \;(\Rightarrow\wedge)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \psi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \vee \psi, \Delta) \rhd \mathcal{H}} \;(\Rightarrow\vee)_\rhd \qquad \frac{\mathcal{G}, (\Gamma, \varphi \Rightarrow \Delta), (\Gamma, \psi \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \vee \psi \Rightarrow \Delta) \rhd \mathcal{H}} \;(\vee\Rightarrow)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma, \psi \Rightarrow \Delta), (\Gamma, \varphi \to \psi \Rightarrow \varphi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \rhd \mathcal{H}} \;(\to)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow p, \Delta), (p, \varphi \Rightarrow \psi) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \to \psi, \Delta) \rhd \mathcal{H}} \;(\Rightarrow\to)_\rhd \qquad p \text{ new}$$

$$\frac{\mathcal{G}, (\Gamma, p \to q \Rightarrow \Delta), (p \Rightarrow \varphi), (\psi \Rightarrow q) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \rhd \mathcal{H}} \;(\to\Rightarrow)_\rhd \qquad p, q \text{ new}$$

Left Rules

$$\frac{\mathcal{G} \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma, \bot \Rightarrow \Delta) \triangleright \mathcal{H}} \ (\bot\Rightarrow)_\triangleright \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \Delta) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \bot, \Delta) \triangleright \mathcal{H}} \ (\Rightarrow\bot)_\triangleright$$

$$\frac{\mathcal{G}, (\Gamma, \varphi, \psi \Rightarrow \Delta) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \land \psi \Rightarrow \Delta) \triangleright \mathcal{H}} \ (\land\Rightarrow)_\triangleright \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \Delta), (\Gamma \Rightarrow \psi, \Delta) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \land \psi, \Delta) \triangleright \mathcal{H}} \ (\Rightarrow\land)_\triangleright$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \psi, \Delta) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \lor \psi, \Delta) \triangleright \mathcal{H}} \ (\Rightarrow\lor)_\triangleright \qquad \frac{\mathcal{G}, (\Gamma, \varphi \Rightarrow \Delta), (\Gamma, \psi \Rightarrow \Delta) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \lor \psi \Rightarrow \Delta) \triangleright \mathcal{H}} \ (\lor\Rightarrow)_\triangleright$$

$$\frac{\mathcal{G}, (\Gamma, \psi \Rightarrow \Delta), (\Gamma, \varphi \to \psi \Rightarrow \varphi, \Delta) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \triangleright \mathcal{H}} \ (\to)_\triangleright$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow p, \Delta), (p, \varphi \Rightarrow \psi) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \to \psi, \Delta) \triangleright \mathcal{H}} \ (\Rightarrow\to)_\triangleright \qquad p \text{ new}$$

$$\frac{\mathcal{G}, (\Gamma, p \to q \Rightarrow \Delta), (p \Rightarrow \varphi), (\psi \Rightarrow q) \triangleright \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \triangleright \mathcal{H}} \ (\to\Rightarrow)_\triangleright \qquad p, q \text{ new}$$

Left Rules

$$\frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \bot \Rightarrow \Delta) \rhd \mathcal{H}} \; (\bot\Rightarrow)_\rhd \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \bot, \Delta) \rhd \mathcal{H}} \; (\Rightarrow\bot)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma, \varphi, \psi \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \wedge \psi \Rightarrow \Delta) \rhd \mathcal{H}} \; (\wedge\Rightarrow)_\rhd \qquad \frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \Delta), (\Gamma \Rightarrow \psi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \wedge \psi, \Delta) \rhd \mathcal{H}} \; (\Rightarrow\wedge)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow \varphi, \psi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \vee \psi, \Delta) \rhd \mathcal{H}} \; (\Rightarrow\vee)_\rhd \qquad \frac{\mathcal{G}, (\Gamma, \varphi \Rightarrow \Delta), (\Gamma, \psi \Rightarrow \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \vee \psi \Rightarrow \Delta) \rhd \mathcal{H}} \; (\vee\Rightarrow)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma, \psi \Rightarrow \Delta), (\Gamma, \varphi \to \psi \Rightarrow \varphi, \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \rhd \mathcal{H}} \; (\to)_\rhd$$

$$\frac{\mathcal{G}, (\Gamma \Rightarrow p, \Delta), (p, \varphi \Rightarrow \psi) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma \Rightarrow \varphi \to \psi, \Delta) \rhd \mathcal{H}} \; (\Rightarrow\to)_\rhd \qquad p \text{ new}$$

$$\frac{\mathcal{G}, (\Gamma, p \to q \Rightarrow \Delta), (p \Rightarrow \varphi), (\psi \Rightarrow q) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \to \psi \Rightarrow \Delta) \rhd \mathcal{H}} \; (\to\Rightarrow)_\rhd \qquad p, q \text{ new}$$

Structural Rules

Weakening Rules

$$\frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G}, S \rhd \mathcal{H}} \ (\mathrm{W})\rhd \qquad \frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G} \rhd S, \mathcal{H}} \ \rhd(\mathrm{W})$$

Anti-Cut Rule

$$\frac{\mathcal{G}, (\Gamma, \Gamma' \Rightarrow \Delta', \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \Rightarrow \Delta), (\Gamma' \Rightarrow \varphi, \Delta') \rhd \mathcal{H}} \ (\mathrm{AC})$$

Projection Rule

$$\frac{\mathcal{G} \rhd (\Gamma, \mathcal{I}(S) \Rightarrow \Delta), \mathcal{H}}{\mathcal{G}, S \rhd \mathcal{H}} \ (\mathrm{PJ})$$

where $(\Gamma \Rightarrow \Delta) \in \mathcal{H} \cup \{\Rightarrow\}$

Weakening Rules

$$\frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G}, S \rhd \mathcal{H}} \;(\text{W})\rhd \qquad \frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G} \rhd S, \mathcal{H}} \;\rhd(\text{W})$$

Anti-Cut Rule

$$\frac{\mathcal{G}, (\Gamma, \Gamma' \Rightarrow \Delta', \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \Rightarrow \Delta), (\Gamma' \Rightarrow \varphi, \Delta') \rhd \mathcal{H}} \;(\text{AC})$$

Projection Rule

$$\frac{\mathcal{G} \rhd (\Gamma, \mathcal{I}(S) \Rightarrow \Delta), \mathcal{H}}{\mathcal{G}, S \rhd \mathcal{H}} \;(\text{PJ})$$

where $(\Gamma \Rightarrow \Delta) \in \mathcal{H} \cup \{\Rightarrow\}$

Structural Rules

Weakening Rules

$$\frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G}, S \rhd \mathcal{H}}\ (\text{W})\rhd \qquad \frac{\mathcal{G} \rhd \mathcal{H}}{\mathcal{G} \rhd S, \mathcal{H}}\ \rhd(\text{W})$$

Anti-Cut Rule

$$\frac{\mathcal{G}, (\Gamma, \Gamma' \Rightarrow \Delta', \Delta) \rhd \mathcal{H}}{\mathcal{G}, (\Gamma, \varphi \Rightarrow \Delta), (\Gamma' \Rightarrow \varphi, \Delta') \rhd \mathcal{H}}\ (\text{AC})$$

Projection Rule

$$\frac{\mathcal{G} \rhd (\Gamma, \mathcal{I}(S) \Rightarrow \Delta), \mathcal{H}}{\mathcal{G}, S \rhd \mathcal{H}}\ (\text{PJ})$$

where $(\Gamma \Rightarrow \Delta) \in \mathcal{H} \cup \{\Rightarrow\}$

Examples

The following "identity" sequent rules are derivable using (PJ):

$$\frac{}{\mathcal{G}, (\Gamma \Rightarrow \Delta) \;\triangleright\; (\Gamma, \Gamma' \Rightarrow \Delta', \Delta), \mathcal{H}} \;(\text{SID})$$

We can also derive sequent rules corresponding to the usual cut rule:

$$\frac{\dfrac{}{(\Gamma, \Gamma' \Rightarrow \Delta', \Delta) \;\triangleright\; (\Gamma, \Gamma' \Rightarrow \Delta', \Delta)} \;(\text{SID})}{(\Gamma, \varphi \Rightarrow \Delta), (\Gamma' \Rightarrow \varphi, \Delta') \;\triangleright\; (\Gamma, \Gamma' \Rightarrow \Delta', \Delta)} \;(\text{AC})$$

Examples

The following “identity” sequent rules are derivable using (PJ):

$$\frac{}{\mathcal{G}, (\Gamma \Rightarrow \Delta) \rhd (\Gamma, \Gamma' \Rightarrow \Delta', \Delta), \mathcal{H}} \ (\text{SID})$$

We can also derive sequent rules corresponding to the usual cut rule:

$$\frac{\overline{(\Gamma, \Gamma' \Rightarrow \Delta', \Delta) \rhd (\Gamma, \Gamma' \Rightarrow \Delta', \Delta)} \ (\text{SID})}{(\Gamma, \varphi \Rightarrow \Delta), (\Gamma' \Rightarrow \varphi, \Delta') \rhd (\Gamma, \Gamma' \Rightarrow \Delta', \Delta)} \ (\text{AC})$$

A Visser Rule

$$\frac{\{\mathcal{G}, (\Gamma \Rightarrow \varphi) \rhd \mathcal{H}\}_{\varphi \in \Delta} \quad \{\mathcal{G} \rhd (\Gamma^{\Pi}, \Pi \Rightarrow \Delta), \mathcal{H}\}_{\emptyset \neq \Pi \subseteq \Gamma_{\Delta}}}{\mathcal{G}, (\Gamma \Rightarrow \Delta) \rhd \mathcal{H}} \ \text{(V)}$$

where Γ contains only implications, and

1. $\Gamma^{\Pi} = \{\varphi \to \psi \in \Gamma \mid \varphi \notin \Pi\}$
2. $\Gamma_{\Delta} = \{\varphi \notin \Delta \mid \exists \psi\, (\varphi \to \psi) \in \Gamma\}$.

A Visser Rule

$$\frac{\{\mathcal{G},(\Gamma \Rightarrow \varphi) \rhd \mathcal{H}\}_{\varphi \in \Delta} \quad \{\mathcal{G} \rhd (\Gamma^{\Pi}, \Pi \Rightarrow \Delta), \mathcal{H}\}_{\emptyset \neq \Pi \subseteq \Gamma_{\Delta}}}{\mathcal{G},(\Gamma \Rightarrow \Delta) \rhd \mathcal{H}} \text{ (V)}$$

where Γ contains only implications, and

1. $\Gamma^{\Pi} = \{\varphi \to \psi \in \Gamma \mid \varphi \notin \Pi\}$
2. $\Gamma_{\Delta} = \{\varphi \notin \Delta \mid \exists \psi\, (\varphi \to \psi) \in \Gamma\}$.

A Visser Rule

$$\frac{\{\mathcal{G}, (\Gamma \Rightarrow \varphi) \rhd \mathcal{H}\}_{\varphi \in \Delta} \quad \{\mathcal{G} \rhd (\Gamma^{\Pi}, \Pi \Rightarrow \Delta), \mathcal{H}\}_{\emptyset \neq \Pi \subseteq \Gamma_{\Delta}}}{\mathcal{G}, (\Gamma \Rightarrow \Delta) \rhd \mathcal{H}} \ (\mathrm{V})$$

where Γ contains only implications, and

1. $\Gamma^{\Pi} = \{\varphi \to \psi \in \Gamma \mid \varphi \notin \Pi\}$
2. $\Gamma_{\Delta} = \{\varphi \notin \Delta \mid \exists \psi \, (\varphi \to \psi) \in \Gamma\}$.

Example

Let (V)′ or (PJ)′ denote (V) or (PJ) with applications of (W)$\triangleright$ and $\triangleright$(W):

The two rightmost leaves in this proof tree are instances of (SID), while the derivability of the other leaf follows from the right rules.

Example

Let $(\text{V})'$ or $(\text{PJ})'$ denote (V) or (PJ) with applications of $(\text{W})\triangleright$ and $\triangleright(\text{W})$:

$$\dfrac{}{(\neg p \Rightarrow q, r) \;\triangleright\; (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\;(\rightarrow)\triangleright$$

The two rightmost leaves in this proof tree are instances of (SID), while the derivability of the other leaf follows from the right rules.

Example

Let $(\textsc{v})'$ or $(\textsc{pj})'$ denote $(\textsc{v})$ or $(\textsc{pj})$ with applications of $(\textsc{w})\triangleright$ and $\triangleright(\textsc{w})$:

$$\dfrac{\dfrac{}{(\neg p \Rightarrow p, q, r), (\bot \Rightarrow q, r) \ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\ (\textsc{v})'}{(\neg p \Rightarrow q, r) \ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\ (\to)\triangleright$$

The two rightmost leaves in this proof tree are instances of $(\textsc{sid})$, while the derivability of the other leaf follows from the right rules.

Example

Let $(\textsc{v})'$ or $(\textsc{pj})'$ denote $(\textsc{v})$ or $(\textsc{pj})$ with applications of $(\textsc{w})\triangleright$ and $\triangleright(\textsc{w})$:

$$\dfrac{\dfrac{}{(\neg p \Rightarrow p) \;\triangleright\; (\neg p \Rightarrow q)}\;(\textsc{pj})' \qquad \dfrac{(\neg p \Rightarrow p, q, r), (\bot \Rightarrow q, r) \;\triangleright\; (\neg p \Rightarrow q), (\neg p \Rightarrow r)}{(\neg p \Rightarrow q, r) \;\triangleright\; (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\;(\to)\triangleright}{}\;(\textsc{v})'$$

The two rightmost leaves in this proof tree are instances of $(\textsc{sid})$, while the derivability of the other leaf follows from the right rules.

Example

Let $(\textsc{v})'$ or $(\textsc{pj})'$ denote $(\textsc{v})$ or $(\textsc{pj})$ with applications of $(\textsc{w})\triangleright$ and $\triangleright(\textsc{w})$:

$$\dfrac{\dfrac{\triangleright\ (\neg p, \neg p \to p \Rightarrow q)}{(\neg p \Rightarrow p)\ \triangleright\ (\neg p \Rightarrow q)}\ (\textsc{pj})' \qquad \dfrac{(\neg p \Rightarrow p, q, r), (\bot \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}{(\neg p \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\ (\to)\triangleright}{}\ (\textsc{v})'$$

The two rightmost leaves in this proof tree are instances of $(\textsc{sid})$, while the derivability of the other leaf follows from the right rules.

Example

Let $(\textsc{v})'$ or $(\textsc{pj})'$ denote $(\textsc{v})$ or $(\textsc{pj})$ with applications of $(\textsc{w})\triangleright$ and $\triangleright(\textsc{w})$:

$$
\dfrac{\dfrac{\triangleright\ (\neg p, \neg p \to p \Rightarrow q)}{(\neg p \Rightarrow p)\ \triangleright\ (\neg p \Rightarrow q)}\ (\textsc{pj})' \qquad (\neg p \Rightarrow q)\ \triangleright\ (\neg p \Rightarrow q)}{\dfrac{(\neg p \Rightarrow p, q, r), (\bot \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}{(\neg p \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\ (\to)\triangleright}\ (\textsc{v})'
$$

The two rightmost leaves in this proof tree are instances of $(\textsc{sid})$, while the derivability of the other leaf follows from the right rules.

Example

Let $(\textsc{v})'$ or $(\textsc{pj})'$ denote $(\textsc{v})$ or $(\textsc{pj})$ with applications of $(\textsc{w})\triangleright$ and $\triangleright(\textsc{w})$:

$$\dfrac{\dfrac{\dfrac{\triangleright\ (\neg p, \neg p \to p \Rightarrow q)}{(\neg p \Rightarrow p)\ \triangleright\ (\neg p \Rightarrow q)}\ (\textsc{pj})' \qquad (\neg p \Rightarrow q)\ \triangleright\ (\neg p \Rightarrow q) \qquad (\neg p \Rightarrow r)\ \triangleright\ (\neg p \Rightarrow r)}{(\neg p \Rightarrow p, q, r), (\bot \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\ (\textsc{v})'}{(\neg p \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\ (\to)\triangleright$$

The two rightmost leaves in this proof tree are instances of $(\textsc{sid})$, while the derivability of the other leaf follows from the right rules.

Example

Let $(\mathrm{V})'$ or $(\mathrm{PJ})'$ denote (V) or (PJ) with applications of $(\mathrm{W})\triangleright$ and $\triangleright(\mathrm{W})$:

$$\dfrac{\dfrac{\triangleright\ (\neg p, \neg p \to p \Rightarrow q)}{(\neg p \Rightarrow p)\ \triangleright\ (\neg p \Rightarrow q)}\ (\mathrm{PJ})' \quad (\neg p \Rightarrow q)\ \triangleright\ (\neg p \Rightarrow q) \quad (\neg p \Rightarrow r)\ \triangleright\ (\neg p \Rightarrow r)}{\dfrac{(\neg p \Rightarrow p, q, r), (\bot \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}{(\neg p \Rightarrow q, r)\ \triangleright\ (\neg p \Rightarrow q), (\neg p \Rightarrow r)}\ (\to)\triangleright}\ (\mathrm{V})'$$

The two rightmost leaves in this proof tree are instances of (SID), while the derivability of the other leaf follows from the right rules.

We also have systems for various classes of transitive modal logics;
e.g., for the Gödel-Löb logic GL, we make use of the rules:

$$\frac{\mathcal{G} \triangleright (\Box\Gamma, \Gamma, \Box\varphi \Rightarrow \varphi), \mathcal{H}}{\mathcal{G} \triangleright (\Box\Gamma, \Gamma' \Rightarrow \Box\varphi, \Delta), \mathcal{H}}$$

$$\frac{\{\mathcal{G}, (\Box\Gamma, \Gamma \Rightarrow \varphi) \triangleright \mathcal{H}\}_{\varphi \in \Delta}}{\mathcal{G}, (\Box\Gamma \Rightarrow \Box\Delta) \triangleright \mathcal{H}}$$

We also have systems for various classes of transitive modal logics; e.g., for the Gödel-Löb logic GL, we make use of the rules:

$$\frac{\mathcal{G} \,\triangleright\, (\Box\Gamma, \Gamma, \Box\varphi \Rightarrow \varphi), \mathcal{H}}{\mathcal{G} \,\triangleright\, (\Box\Gamma, \Gamma' \Rightarrow \Box\varphi, \Delta), \mathcal{H}}$$

$$\frac{\{\mathcal{G}, (\Box\Gamma, \Gamma \Rightarrow \varphi) \,\triangleright\, \mathcal{H}\}_{\varphi \in \Delta}}{\mathcal{G}, (\Box\Gamma \Rightarrow \Box\Delta) \,\triangleright\, \mathcal{H}}$$

Modal Logics

We also have systems for various classes of transitive modal logics; e.g., for the Gödel-Löb logic GL, we make use of the rules:

$$\frac{\mathcal{G} \ \triangleright\ (\Box\Gamma, \Gamma, \Box\varphi \Rightarrow \varphi), \mathcal{H}}{\mathcal{G} \ \triangleright\ (\Box\Gamma, \Gamma' \Rightarrow \Box\varphi, \Delta), \mathcal{H}}$$

$$\frac{\{\mathcal{G}, (\Box\Gamma, \Gamma \Rightarrow \varphi) \ \triangleright\ \mathcal{H}\}_{\varphi \in \Delta}}{\mathcal{G}, (\Box\Gamma \Rightarrow \Box\Delta) \ \triangleright\ \mathcal{H}}$$

We also have systems for various classes of transitive modal logics; e.g., for the Gödel-Löb logic GL, we make use of the rules:

$$\frac{\mathcal{G} \;\triangleright\; (\Box\Gamma, \Gamma, \Box\varphi \Rightarrow \varphi), \mathcal{H}}{\mathcal{G} \;\triangleright\; (\Box\Gamma, \Gamma' \Rightarrow \Box\varphi, \Delta), \mathcal{H}}$$

$$\frac{\{\mathcal{G}, (\Box\Gamma, \Gamma \Rightarrow \varphi) \;\triangleright\; \mathcal{H}\}_{\varphi \in \Delta}}{\mathcal{G}, (\Box\Gamma \Rightarrow \Box\Delta) \;\triangleright\; \mathcal{H}}$$

Related Work

Tableaux methods for checking admissibility in IPC and modal logics have also been developed.

S. Ghilardi. A resolution/tableaux algorithm for projective approximations in IPC. *Logic journal of the IGPL* 10(3):227–241, 2002.

S. Babenyshev, V. Rybakov, R. A. Schmidt, and D. Tishkovsky. A tableau method for checking rule admissibility in S4. *Proceedings of UNIF 2009*, ENTCS 262:17–32, 2010.

Recall also that proof systems for checking admissibility in finite-valued logics can be automatically generated:

G. Metcalfe and C. Röthlisberger. Unifiability and admissibility in finite algebras. *Proceedings of CiE 2012*, LNCS 7318: 485–495. Springer, 2012.

Complexity Results

Jeřábek has characterized the computational complexity of admissibility in various families of intermediate and modal logics.

In particular, deciding admissibility is coNEXP-complete for IPC, KC, $\mathrm{K4}$, $\mathrm{S4}$, GL, etc.

E. Jeřábek. Complexity of admissible rules.
Archive for Mathematical Logic 46(2):73–92, 2007.

A Question

Can admissible rules be *useful* for proof theory? E.g., for shortening proofs or speeding up proof search?

This is the case for the **cut rule** in sequent calculi. . . .

Note, however, that for IPC and extensible modal logics, systems with admissible rules are polynomially simulated by the original systems.

G. Mints and A. Kojevnikov. Intuitionistic Frege systems are polynomially equivalent. *Zapisky Nauchnych Seminarov POMI* 316:129–146, 2004.

E. Jeřábek. Frege systems for extensible modal logics. *Annals of Pure and Applied Logic* 142: 366–379, 2006.

A Question

Can admissible rules be *useful* for proof theory? E.g., for shortening proofs or speeding up proof search?

This is the case for the **cut rule** in sequent calculi. . . .

Note, however, that for IPC and extensible modal logics, systems with admissible rules are polynomially simulated by the original systems.

G. Mints and A. Kojevnikov. Intuitionistic Frege systems are polynomially equivalent. *Zapisky Nauchnych Seminarov POMI* 316:129–146, 2004.

E. Jeřábek. Frege systems for extensible modal logics. *Annals of Pure and Applied Logic* 142: 366–379, 2006.

A Question

Can admissible rules be *useful* for proof theory? E.g., for shortening proofs or speeding up proof search?

This is the case for the **cut rule** in sequent calculi. . . .

Note, however, that for IPC and extensible modal logics, systems with admissible rules are polynomially simulated by the original systems.

G. Mints and A. Kojevnikov. Intuitionistic Frege systems are polynomially equivalent. *Zapisky Nauchnych Seminarov POMI* 316:129–146, 2004.

E. Jeřábek. Frege systems for extensible modal logics. *Annals of Pure and Applied Logic* 142: 366–379, 2006.

Part V

A First-Order Framework

Two Notions of Admissibility

Informally, for a system S and rules consisting of a finite sets of premises and conclusions, there are two notions of admissibility:

(A) *"A rule is* **admissible** *in S if the set of theorems of S does not change when the rule is added to the existing rules of S."*

(B) *"A rule is* **admissible** *in S if any substitution mapping all of its premises to theorems of S, also maps one of its conclusions to a theorem of S."*

We have seen that these notions coincide for the single-conclusion rules of a logic, but not always in other cases...

Two Notions of Admissibility

Informally, for a system S and rules consisting of a finite sets of premises and conclusions, there are two notions of admissibility:

(A) *"A rule is* **admissible** *in S if the set of theorems of S does not change when the rule is added to the existing rules of S."*

(B) *"A rule is* **admissible** *in S if any substitution mapping all of its premises to theorems of S, also maps one of its conclusions to a theorem of S."*

We have seen that these notions coincide for the single-conclusion rules of a logic, but not always in other cases...

Two Notions of Admissibility

Informally, for a system S and rules consisting of a finite sets of premises and conclusions, there are two notions of admissibility:

(A) *"A rule is* **admissible** *in S if the set of theorems of S does not change when the rule is added to the existing rules of S."*

(B) *"A rule is* **admissible** *in S if any substitution mapping all of its premises to theorems of S, also maps one of its conclusions to a theorem of S."*

We have seen that these notions coincide for the single-conclusion rules of a logic, but not always in other cases. . .

Two Notions of Admissibility

Informally, for a system S and rules consisting of a finite sets of premises and conclusions, there are two notions of admissibility:

(A) *"A rule is* **admissible** *in S if the set of theorems of S does not change when the rule is added to the existing rules of S."*

(B) *"A rule is* **admissible** *in S if any substitution mapping all of its premises to theorems of S, also maps one of its conclusions to a theorem of S."*

We have seen that these notions coincide for the single-conclusion rules of a logic, but not always in other cases...

The Disjunction Property

The **disjunction property**

$$\{p \lor q\} \ \rhd \ \{p, \ q\}$$

is admissible in IPC according to both (A) and (B).

The Linearity Property

However, the **linearity property**

$$\rhd \{p \to q,\ q \to p\}$$

is admissible in **Gödel logic** (i.e., $\mathrm{IPC} + (\varphi \to \psi) \vee (\psi \to \varphi)$) according to (A), but not (B).

The Density Rule

Moreover, the **density rule**

$$\{((\varphi \to p) \lor (p \to \psi)) \lor \chi\} \rhd \{(\varphi \to \psi) \lor \chi\}$$
$$\text{where } p \text{ does not occur in } \varphi, \psi, \text{ or } \chi$$

is admissible in Gödel logic according to (A), but admissibility according to (B) does not really make much sense…

G. Takeuti and T. Titani. Intuitionistic fuzzy logic and intuitionistic fuzzy set theory. *Journal of Symbolic Logic*, 49(3):851–866, 1984.

The Density Rule

Moreover, the **density rule**

$$\{((\varphi \to p) \lor (p \to \psi)) \lor \chi\} \rhd \{(\varphi \to \psi) \lor \chi\}$$
$$\text{where } p \text{ does not occur in } \varphi, \psi, \text{ or } \chi$$

is admissible in Gödel logic according to (A), but admissibility according to (B) does not really make much sense...

G. Takeuti and T. Titani. Intuitionistic fuzzy logic and intuitionistic fuzzy set theory. *Journal of Symbolic Logic*, 49(3):851–866, 1984.

More Generally...

What does it mean for a **first-order sentence** such as

$$(\exists x)(\forall y)(x \leq y) \qquad \text{or} \qquad (\forall x)(\exists y)\neg(x \leq y)$$

to be admissible in a logic / class of algebras?

First-Order Logic

We assume the usual terminology of **first-order logic with equality**, making use of the symbols $\forall$, $\exists$, $\sqcap$, $\sqcup$, $\Rightarrow$, $\sim$, 0, 1, and $\approx$.

In particular, for a first-order language $\mathcal{L}$, Sen($\mathcal{L}$) is the set of sentences of $\mathcal{L}$ with respect to a countably infinite set of variables.

We will denote $\mathcal{L}$**-terms** by s, t, u, **(first-order)** $\mathcal{L}$**-formulas** by φ, ψ, and **sets of** $\mathcal{L}$**-formulas** by Σ, Θ.

We assume the usual terminology of **first-order logic with equality**, making use of the symbols $\forall$, $\exists$, $\sqcap$, $\sqcup$, $\Rightarrow$, $\sim$, 0, 1, and $\approx$.

In particular, for a first-order language $\mathcal{L}$, Sen($\mathcal{L}$) is the set of sentences of $\mathcal{L}$ with respect to a countably infinite set of variables.

We will denote $\mathcal{L}$**-terms** by s, t, u, **(first-order)** $\mathcal{L}$**-formulas** by φ, ψ, and **sets of** $\mathcal{L}$**-formulas** by Σ, Θ.

First-Order Logic

We assume the usual terminology of **first-order logic with equality**, making use of the symbols $\forall$, $\exists$, $\sqcap$, $\sqcup$, $\Rightarrow$, $\sim$, 0, 1, and $\approx$.

In particular, for a first-order language $\mathcal{L}$, $\mathrm{Sen}(\mathcal{L})$ is the set of sentences of $\mathcal{L}$ with respect to a countably infinite set of variables.

We will denote $\mathcal{L}$**-terms** by s, t, u, **(first-order)** $\mathcal{L}$**-formulas** by φ, ψ, and **sets of** $\mathcal{L}$**-formulas** by Σ, Θ.

Preserving First-Order Sentences

Definition

For a class of $\mathcal{L}$-structures K and $\Sigma \subseteq \mathrm{Sen}(\mathcal{L})$, we set

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \{\psi \in \Sigma \mid \mathsf{K} \models \psi\}$$

and say that $\varphi \in \mathrm{Sen}(\mathcal{L})$ **preserves** Σ **in** K if

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \mathrm{Th}_\Sigma(\{\mathbf{A} \in \mathsf{K} \mid \mathbf{A} \models \varphi\}).$$

If K is axiomatized by $\Theta \subseteq \mathrm{Sen}(\mathcal{L})$, then φ *preserves* Σ *in* K when:

$$\text{For all } \psi \in \Sigma: \quad \Theta \models \psi \quad \text{iff} \quad \Theta \cup \{\varphi\} \models \psi.$$

Preserving First-Order Sentences

Definition

For a class of $\mathcal{L}$-structures K and $\Sigma \subseteq \text{Sen}(\mathcal{L})$, we set

$$\text{Th}_\Sigma(\mathsf{K}) = \{\psi \in \Sigma \mid \mathsf{K} \models \psi\}$$

and say that $\varphi \in \text{Sen}(\mathcal{L})$ **preserves** Σ **in** K if

$$\text{Th}_\Sigma(\mathsf{K}) = \text{Th}_\Sigma(\{\mathbf{A} \in \mathsf{K} \mid \mathbf{A} \models \varphi\}).$$

If K is axiomatized by $\Theta \subseteq \text{Sen}(\mathcal{L})$, then φ *preserves* Σ *in* K when:

$$\text{For all } \psi \in \Sigma: \quad \Theta \models \psi \quad \text{iff} \quad \Theta \cup \{\varphi\} \models \psi.$$

Preserving First-Order Sentences

Definition

For a class of $\mathcal{L}$-structures K and $\Sigma \subseteq \text{Sen}(\mathcal{L})$, we set

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \{\psi \in \Sigma \mid \mathsf{K} \models \psi\}$$

and say that $\varphi \in \text{Sen}(\mathcal{L})$ **preserves** Σ **in** K if

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \mathrm{Th}_\Sigma(\{\mathbf{A} \in \mathsf{K} \mid \mathbf{A} \models \varphi\}).$$

If K is axiomatized by $\Theta \subseteq \text{Sen}(\mathcal{L})$, then φ *preserves* Σ *in* K when:

$$\text{For all } \psi \in \Sigma\text{:} \quad \Theta \models \psi \quad \text{iff} \quad \Theta \cup \{\varphi\} \models \psi.$$

Preserving First-Order Sentences

Definition

For a class of $\mathcal{L}$-structures K and $\Sigma \subseteq$ Sen($\mathcal{L}$), we set

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \{\psi \in \Sigma \mid \mathsf{K} \models \psi\}$$

and say that $\varphi \in$ Sen($\mathcal{L}$) **preserves** Σ **in** K if

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \mathrm{Th}_\Sigma(\{\mathbf{A} \in \mathsf{K} \mid \mathbf{A} \models \varphi\}).$$

If K is axiomatized by $\Theta \subseteq$ Sen($\mathcal{L}$), then φ *preserves* Σ *in* K when:

$$\text{For all } \psi \in \Sigma\text{:} \quad \Theta \models \psi \quad \text{iff} \quad \Theta \cup \{\varphi\} \models \psi.$$

Definition

For a class of $\mathcal{L}$-structures K and $\Sigma \subseteq \mathrm{Sen}(\mathcal{L})$, we set

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \{\psi \in \Sigma \mid \mathsf{K} \models \psi\}$$

and say that $\varphi \in \mathrm{Sen}(\mathcal{L})$ **preserves** Σ **in** K if

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \mathrm{Th}_\Sigma(\{\mathbf{A} \in \mathsf{K} \mid \mathbf{A} \models \varphi\}).$$

If K is axiomatized by $\Theta \subseteq \mathrm{Sen}(\mathcal{L})$, then φ *preserves* Σ *in* K when:

$$\text{For all } \psi \in \Sigma: \quad \Theta \models \psi \quad \text{iff} \quad \Theta \cup \{\varphi\} \models \psi.$$

Preserving First-Order Sentences

Definition

For a class of $\mathcal{L}$-structures K and $\Sigma \subseteq$ Sen($\mathcal{L}$), we set

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \{\psi \in \Sigma \mid \mathsf{K} \models \psi\}$$

and say that $\varphi \in$ Sen($\mathcal{L}$) **preserves** Σ **in** K if

$$\mathrm{Th}_\Sigma(\mathsf{K}) = \mathrm{Th}_\Sigma(\{\mathbf{A} \in \mathsf{K} \mid \mathbf{A} \models \varphi\}).$$

If K is axiomatized by $\Theta \subseteq$ Sen($\mathcal{L}$), then φ *preserves* Σ *in* K when:

$$\text{For all } \psi \in \Sigma\text{:} \quad \Theta \models \psi \quad \text{iff} \quad \Theta \cup \{\varphi\} \models \psi.$$

Consider the variety BA of **Boolean algebras** in a language $\mathcal{L}_{\text{Bool}}$ and

$$\varphi = (\forall x)((x \approx \bot) \sqcup (x \approx \top)).$$

Then φ preserves the set of $\mathcal{L}_{\text{Bool}}$-equations in BA, but $\mathbf{F}_{\text{BA}}(\omega) \not\models \varphi$.

Note that $\neg\varphi$ also preserves the set of $\mathcal{L}_{\text{Bool}}$-equations in BA.

Consider the variety BA of **Boolean algebras** in a language $\mathcal{L}_{\mathrm{Bool}}$ and

$$\varphi = (\forall x)((x \approx \bot) \sqcup (x \approx \top)).$$

Then φ preserves the set of $\mathcal{L}_{\mathrm{Bool}}$-equations in BA, but $\mathbf{F}_{\mathsf{BA}}(\omega) \not\models \varphi$.

Note that $\neg\varphi$ also preserves the set of $\mathcal{L}_{\mathrm{Bool}}$-equations in BA.

Example: Boolean Algebras

Consider the variety BA of **Boolean algebras** in a language $\mathcal{L}_{\mathrm{Bool}}$ and

$$\varphi = (\forall x)((x \approx \bot) \sqcup (x \approx \top)).$$

Then φ preserves the set of $\mathcal{L}_{\mathrm{Bool}}$-equations in BA, but $\mathbf{F}_{\mathsf{BA}}(\omega) \not\models \varphi$.

Note that $\neg\varphi$ also preserves the set of $\mathcal{L}_{\mathrm{Bool}}$-equations in BA.

Consider the variety BA of **Boolean algebras** in a language $\mathcal{L}_{\mathrm{Bool}}$ and

$$\varphi = (\forall x)((x \approx \bot) \sqcup (x \approx \top)).$$

Then φ preserves the set of $\mathcal{L}_{\mathrm{Bool}}$-equations in BA, but $\mathbf{F}_{\mathsf{BA}}(\omega) \not\models \varphi$.

Note that $\neg\varphi$ also preserves the set of $\mathcal{L}_{\mathrm{Bool}}$-equations in BA.

Theorem

The following are equivalent for an $\mathcal{L}$**-quasivariety** $\mathcal{Q}$ *and* $\mathcal{L}$**-quasiequation** φ*:*

(i) φ *is* $\mathcal{Q}$*-admissible*

(ii) $\mathbf{F}_{\mathcal{Q}}(\omega) \models \varphi$

(iii) $\mathbb{V}(\mathcal{Q}) = \mathbb{V}(\{\mathbf{A} \in \mathcal{Q} \mid \mathbf{A} \models \varphi\})$

(iv) φ *preserves the set of* $\mathcal{L}$**-equations** *in* $\mathcal{Q}$.

Quasivarieties and Quasiequations

Theorem

*The following are equivalent for an $\mathcal{L}$-**quasivariety** $\mathcal{Q}$ and $\mathcal{L}$-**quasiequation** φ:*

(i) φ *is* $\mathcal{Q}$*-admissible*

(ii) $\mathbf{F}_{\mathcal{Q}}(\omega) \models \varphi$

(iii) $\mathbb{V}(\mathcal{Q}) = \mathbb{V}(\{\mathbf{A} \in \mathcal{Q} \mid \mathbf{A} \models \varphi\})$

(iv) φ *preserves the set of* $\mathcal{L}$-**equations** *in* $\mathcal{Q}$*.*

Theorem

The following are equivalent for an $\mathcal{L}$-**quasivariety** $\mathcal{Q}$ *and* $\mathcal{L}$-**quasiequation** φ:

(i) φ *is* $\mathcal{Q}$*-admissible*

(ii) $\mathbf{F}_{\mathcal{Q}}(\omega) \models \varphi$

(iii) $\mathbb{V}(\mathcal{Q}) = \mathbb{V}(\{\mathbf{A} \in \mathcal{Q} \mid \mathbf{A} \models \varphi\})$

(iv) φ *preserves the set of* $\mathcal{L}$-**equations** *in* $\mathcal{Q}$.

Theorem

*The following are equivalent for an $\mathcal{L}$-**quasivariety** $\mathcal{Q}$ and $\mathcal{L}$-**clause** φ:*

(i) φ *is* $\mathcal{Q}$*-admissible*

(ii) $\mathbf{F}_{\mathcal{Q}}(\omega) \models \varphi$

(iii) $\mathbb{U}^{+}(\mathcal{Q}) = \mathbb{U}^{+}(\{\mathbf{A} \in \mathcal{Q} \mid \mathbf{A} \models \varphi\})$

(iv) φ *preserves the set of **positive** $\mathcal{L}$**-clauses** in* $\mathcal{Q}$.

Quasivarieties and Clauses

Theorem

*The following are equivalent for an $\mathcal{L}$-**quasivariety** $\mathcal{Q}$ and $\mathcal{L}$-**clause** φ:*

(i) φ *is* $\mathcal{Q}$*-admissible*

(ii) $\mathbf{F}_{\mathcal{Q}}(\omega) \models \varphi$

(iii) $\mathbb{U}^+(\mathcal{Q}) = \mathbb{U}^+(\{\mathbf{A} \in \mathcal{Q} \mid \mathbf{A} \models \varphi\})$

(iv) φ *preserves the set of* **positive** $\mathcal{L}$**-clauses** *in* $\mathcal{Q}$.

Quasivarieties and Clauses

Theorem

*The following are equivalent for an $\mathcal{L}$-**quasivariety** $\mathcal{Q}$ and $\mathcal{L}$-**clause** φ:*

(i) φ *is $\mathcal{Q}$-admissible*

(ii) $\mathbf{F}_{\mathcal{Q}}(\omega) \models \varphi$

(iii) $\mathbb{U}^+(\mathcal{Q}) = \mathbb{U}^+(\{\mathbf{A} \in \mathcal{Q} \mid \mathbf{A} \models \varphi\})$

(iv) φ *preserves the set of **positive** $\mathcal{L}$**-clauses** in $\mathcal{Q}$.*

Positive Clauses Preserving Equations

Theorem

If $\mathcal{V}$ is a **congruence distributive $\mathcal{L}$-variety**, *then the following are equivalent for any positive $\mathcal{L}$-clause φ:*

(i) $\mathbf{A} \models \varphi$ *for all* **subdirectly irreducible** *algebras* $\mathbf{A} \in \mathcal{V}$

(ii) φ *preserves the set of $\mathcal{L}$-quasiequations in $\mathcal{V}$*

(iii) φ *preserves the set of $\mathcal{L}$-equations in $\mathcal{V}$.*

For example, the positive clause

$$\rhd \{x \leq y, \; y \leq x\}.$$

is valid in all subdirectly irreducible Gödel Algebras.

Positive Clauses Preserving Equations

Theorem

If $\mathcal{V}$ is a **congruence distributive $\mathcal{L}$-variety**, *then the following are equivalent for any positive $\mathcal{L}$-clause φ:*

(i) $\mathbf{A} \models \varphi$ *for all* **subdirectly irreducible** *algebras* $\mathbf{A} \in \mathcal{V}$

(ii) φ *preserves the set of $\mathcal{L}$-quasiequations in $\mathcal{V}$*

(iii) φ *preserves the set of $\mathcal{L}$-equations in $\mathcal{V}$.*

For example, the positive clause

$$\rhd \{x \le y, \ y \le x\}.$$

is valid in all subdirectly irreducible Gödel Algebras.

Positive Clauses Preserving Equations

Theorem

If $\mathcal{V}$ is a **congruence distributive $\mathcal{L}$-variety**, *then the following are equivalent for any positive $\mathcal{L}$-clause φ:*

(i) $\mathbf{A} \models \varphi$ *for all* **subdirectly irreducible** *algebras* $\mathbf{A} \in \mathcal{V}$

(ii) φ *preserves the set of $\mathcal{L}$-quasiequations in $\mathcal{V}$*

(iii) φ *preserves the set of $\mathcal{L}$-equations in $\mathcal{V}$.*

For example, the positive clause

$$\rhd \{x \leq y, \; y \leq x\}.$$

is valid in all subdirectly irreducible Gödel Algebras.

Skolemization

Consider an algebraic language $\mathcal{L}$ and a **prenex formula** $\varphi \in \mathrm{Sen}(\mathcal{L})$.

The **Skolem form** $\mathrm{sk}(\varphi) \in \mathrm{Sen}(\mathcal{L}')$ of φ is obtained by repeating

$$(\forall \bar{x})(\exists y)\varphi(\bar{x}, y) \qquad \Longrightarrow \qquad (\forall \bar{x})\varphi(\bar{x}, f(\bar{x})) \qquad f \text{ new}.$$

Then for any $\Theta \cup \{\psi\} \subseteq \mathrm{Sen}(\mathcal{L})$:

$$\Theta \cup \{\varphi\} \models \psi \quad \text{iff} \quad \Theta \cup \{\mathrm{sk}(\varphi)\} \models \psi.$$

Skolemization

Consider an algebraic language $\mathcal{L}$ and a **prenex formula** $\varphi \in \text{Sen}(\mathcal{L})$.

The **Skolem form** $\text{sk}(\varphi) \in \text{Sen}(\mathcal{L}')$ of φ is obtained by repeating

$$(\forall \bar{x})(\exists y)\varphi(\bar{x}, y) \qquad \Longrightarrow \qquad (\forall \bar{x})\varphi(\bar{x}, f(\bar{x})) \qquad f \text{ new}.$$

Then for any $\Theta \cup \{\psi\} \subseteq \text{Sen}(\mathcal{L})$:

$$\Theta \cup \{\varphi\} \models \psi \quad \text{iff} \quad \Theta \cup \{\text{sk}(\varphi)\} \models \psi.$$

Skolemization

Consider an algebraic language $\mathcal{L}$ and a **prenex formula** $\varphi \in \mathrm{Sen}(\mathcal{L})$.

The **Skolem form** $\mathrm{sk}(\varphi) \in \mathrm{Sen}(\mathcal{L}')$ of φ is obtained by repeating

$$(\forall \bar{x})(\exists y)\varphi(\bar{x}, y) \qquad \Longrightarrow \qquad (\forall \bar{x})\varphi(\bar{x}, f(\bar{x})) \qquad f \text{ new}.$$

Then for any $\Theta \cup \{\psi\} \subseteq \mathrm{Sen}(\mathcal{L})$:

$$\Theta \cup \{\varphi\} \models \psi \quad \text{iff} \quad \Theta \cup \{\mathrm{sk}(\varphi)\} \models \psi.$$

Skolemization

Let K be an elementary class of $\mathcal{L}$-structures, $\mathcal{L}'$ an extension of $\mathcal{L}$, and K$'$ the class of $\mathcal{L}'$-structures whose $\mathcal{L}$-reducts are in K.

Lemma

The following are equivalent for any $\Sigma \cup \{\varphi\} \subseteq \text{Sen}(\mathcal{L})$*:*

(1) φ *preserves* Σ *in* K

(2) $\text{sk}(\varphi) \in \text{Sen}(\mathcal{L}')$ *preserves* Σ *in* K$'$.

Skolemization

Let K be an elementary class of $\mathcal{L}$-structures, $\mathcal{L}'$ an extension of $\mathcal{L}$, and K$'$ the class of $\mathcal{L}'$-structures whose $\mathcal{L}$-reducts are in K.

Lemma

The following are equivalent for any $\Sigma \cup \{\varphi\} \subseteq \mathrm{Sen}(\mathcal{L})$*:*

(1) φ *preserves* Σ *in* K

(2) $\mathrm{sk}(\varphi) \in \mathrm{Sen}(\mathcal{L}')$ *preserves* Σ *in* K$'$.

Skolemization

Let K be an elementary class of $\mathcal{L}$-structures, $\mathcal{L}'$ an extension of $\mathcal{L}$, and K$'$ the class of $\mathcal{L}'$-structures whose $\mathcal{L}$-reducts are in K.

Lemma

The following are equivalent for any $\Sigma \cup \{\varphi\} \subseteq \mathrm{Sen}(\mathcal{L})$*:*

(1) φ *preserves* Σ *in* K

(2) $\mathrm{sk}(\varphi) \in \mathrm{Sen}(\mathcal{L}')$ *preserves* Σ *in* K$'$.

Skolemization

Let K be an elementary class of $\mathcal{L}$-structures, $\mathcal{L}'$ an extension of $\mathcal{L}$, and K$'$ the class of $\mathcal{L}'$-structures whose $\mathcal{L}$-reducts are in K.

Lemma

The following are equivalent for any $\Sigma \cup \{\varphi\} \subseteq \mathrm{Sen}(\mathcal{L})$*:*

(1) φ *preserves* Σ *in* K

(2) $\mathrm{sk}(\varphi) \in \mathrm{Sen}(\mathcal{L}')$ *preserves* Σ *in* K$'$.

Skolemization

Let K be an elementary class of $\mathcal{L}$-structures, $\mathcal{L}'$ an extension of $\mathcal{L}$, and K$'$ the class of $\mathcal{L}'$-structures whose $\mathcal{L}$-reducts are in K.

Lemma

The following are equivalent for any $\Sigma \cup \{\varphi\} \subseteq \mathrm{Sen}(\mathcal{L})$*:*

(1) φ *preserves* Σ *in* K

(2) $\mathrm{sk}(\varphi) \in \mathrm{Sen}(\mathcal{L}')$ *preserves* Σ *in* K$'$*.*

Part VI

Eliminations and Applications

The Idea

Question. How can we prove that φ **preserves** Σ **in** K?

An Answer.

(a) Give a **proof system** that checks for a given $\psi \in \Sigma$ whether

$$\mathrm{Th}(\mathrm{K}) \cup \{\varphi\} \models \psi.$$

(b) Show that "applications of φ" can be **eliminated** from proofs.

Let us begin with some simple observations for **lattices**.

S. Negri and J. Von Plato. Proof systems for lattice theory.
Mathematical Structures in Computer Science, 14(4):507–526, 2004.

The Idea

Question. How can we prove that φ **preserves** Σ **in** K?

An Answer.

(a) Give a **proof system** that checks for a given $\psi \in \Sigma$ whether

$$\mathrm{Th}(\mathrm{K}) \cup \{\varphi\} \models \psi.$$

(b) Show that "applications of φ" can be **eliminated** from proofs.

Let us begin with some simple observations for **lattices**.

S. Negri and J. Von Plato. Proof systems for lattice theory.
Mathematical Structures in Computer Science, 14(4):507–526, 2004.

The Idea

Question. How can we prove that φ **preserves** Σ **in** K?

An Answer.

(a) Give a **proof system** that checks for a given $\psi \in \Sigma$ whether

$$\mathrm{Th}(\mathrm{K}) \cup \{\varphi\} \models \psi.$$

(b) Show that "applications of φ" can be **eliminated** from proofs.

Let us begin with some simple observations for **lattices**.

S. Negri and J. Von Plato. Proof systems for lattice theory.
Mathematical Structures in Computer Science, 14(4):507–526, 2004.

The Idea

Question. How can we prove that φ **preserves** Σ **in** K?

An Answer.

(a) Give a **proof system** that checks for a given $\psi \in \Sigma$ whether

$$\mathrm{Th}(\mathsf{K}) \cup \{\varphi\} \models \psi.$$

(b) Show that "applications of φ" can be **eliminated** from proofs.

Let us begin with some simple observations for **lattices**.

S. Negri and J. Von Plato. Proof systems for lattice theory.
Mathematical Structures in Computer Science, 14(4):507–526, 2004.

Question. How can we prove that φ **preserves** Σ **in** K?

An Answer.

(a) Give a **proof system** that checks for a given $\psi \in \Sigma$ whether

$$\mathrm{Th}(\mathsf{K}) \cup \{\varphi\} \models \psi.$$

(b) Show that "applications of φ" can be **eliminated** from proofs.

Let us begin with some simple observations for **lattices**.

S. Negri and J. Von Plato. Proof systems for lattice theory.
Mathematical Structures in Computer Science, 14(4):507–526, 2004.

A Proof System GLat for Lattices

Axioms

$$\frac{}{s \leq s}\ (\text{ID})$$

Cut rule

$$\frac{s \leq u \quad u \leq t}{s \leq t}\ (\text{CUT})$$

Left rules

$$\frac{s_1 \leq t}{s_1 \wedge s_2 \leq t}\ (\wedge\Rightarrow)_1$$

$$\frac{s_2 \leq t}{s_1 \wedge s_2 \leq t}\ (\wedge\Rightarrow)_2$$

$$\frac{s_1 \leq t \quad s_2 \leq t}{s_1 \vee s_2 \leq t}\ (\vee\Rightarrow)$$

Right rules

$$\frac{t \leq s_1}{t \leq s_1 \vee s_2}\ (\Rightarrow\vee)_1$$

$$\frac{t \leq s_2}{t \leq s_1 \vee s_2}\ (\Rightarrow\vee)_2$$

$$\frac{t \leq s_1 \quad t \leq s_2}{t \leq s_1 \wedge s_2}\ (\Rightarrow\wedge)$$

Theorem

(a) $\vdash_{\mathrm{GLat}} s \leq t$ *iff* $\models_{\mathrm{Lat}} s \leq t$.

(b) GLat *admits cut-elimination.*

A Proof System GLat for Lattices

Axioms

$$\frac{}{s \leq s}\ (\textsc{id})$$

Cut rule

$$\frac{s \leq u \quad u \leq t}{s \leq t}\ (\textsc{cut})$$

Left rules

$$\frac{s_1 \leq t}{s_1 \wedge s_2 \leq t}\ (\wedge\Rightarrow)_1$$

$$\frac{s_2 \leq t}{s_1 \wedge s_2 \leq t}\ (\wedge\Rightarrow)_2$$

$$\frac{s_1 \leq t \quad s_2 \leq t}{s_1 \vee s_2 \leq t}\ (\vee\Rightarrow)$$

Right rules

$$\frac{t \leq s_1}{t \leq s_1 \vee s_2}\ (\Rightarrow\vee)_1$$

$$\frac{t \leq s_2}{t \leq s_1 \vee s_2}\ (\Rightarrow\vee)_2$$

$$\frac{t \leq s_1 \quad t \leq s_2}{t \leq s_1 \wedge s_2}\ (\Rightarrow\wedge)$$

Theorem

(a) $\vdash_{\mathrm{GLat}} s \leq t$ *iff* $\models_{\mathrm{Lat}} s \leq t$.

(b) GLat *admits cut-elimination.*

A Proof System GLat for Lattices

Axioms

$$\frac{}{s \leq s}\ (\text{ID})$$

Cut rule

$$\frac{s \leq u \quad u \leq t}{s \leq t}\ (\text{CUT})$$

Left rules

$$\frac{s_1 \leq t}{s_1 \wedge s_2 \leq t}\ (\wedge\Rightarrow)_1$$

$$\frac{s_2 \leq t}{s_1 \wedge s_2 \leq t}\ (\wedge\Rightarrow)_2$$

$$\frac{s_1 \leq t \quad s_2 \leq t}{s_1 \vee s_2 \leq t}\ (\vee\Rightarrow)$$

Right rules

$$\frac{t \leq s_1}{t \leq s_1 \vee s_2}\ (\Rightarrow\vee)_1$$

$$\frac{t \leq s_2}{t \leq s_1 \vee s_2}\ (\Rightarrow\vee)_2$$

$$\frac{t \leq s_1 \quad t \leq s_2}{t \leq s_1 \wedge s_2}\ (\Rightarrow\wedge)$$

Theorem

(a) $\vdash_{\mathrm{GLat}} s \leq t$ *iff* $\models_{\mathrm{Lat}} s \leq t$.

(b) GLat *admits cut-elimination.*

Example: Boundedness in Lattices

Consider the following $\mathcal{L}_{\text{Lat}}$-sentence for expressing **boundedness**:

$$\varphi_{bd} = (\exists x)(\exists y)(\forall z)((x \leq z) \sqcap (z \leq y)).$$

Skolemizing this sentence gives

$$\text{sk}(\varphi_{bd}) = (\forall z)((\bot \leq z) \sqcap (z \leq \top))$$

in a language $\mathcal{L}^{b}_{\text{Lat}}$ containing extra constants $\bot$ and $\top$.

We consider GLat extended with the rules:

$$\frac{}{\bot \leq t}\,(\bot\Rightarrow) \qquad \text{and} \qquad \frac{}{s \leq \top}\,(\Rightarrow\top).$$

Theorem

(a) φ_{bd} *preserves the set of* $\mathcal{L}_{\text{Lat}}$*-equations in* Lat.

(b) $\mathsf{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathsf{Lat} \mid \mathbf{A} \textit{ is bounded}\})$.

Example: Boundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **boundedness**:

$$\varphi_{bd} = (\exists x)(\exists y)(\forall z)((x \leq z) \sqcap (z \leq y)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{bd}) = (\forall z)((\bot \leq z) \sqcap (z \leq \top))$$

in a language $\mathcal{L}^{b}_{\mathrm{Lat}}$ containing extra constants $\bot$ and $\top$.

We consider GLat extended with the rules:

$$\frac{}{\bot \leq t}\ (\bot\Rightarrow) \quad \text{and} \quad \frac{}{s \leq \top}\ (\Rightarrow\top).$$

Theorem

(a) φ_{bd} *preserves the set of* $\mathcal{L}_{\mathrm{Lat}}$*-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A} \textit{ is bounded}\})$.

Example: Boundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **boundedness**:

$$\varphi_{bd} = (\exists x)(\exists y)(\forall z)((x \leq z) \sqcap (z \leq y)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{bd}) = (\forall z)((\bot \leq z) \sqcap (z \leq \top))$$

in a language $\mathcal{L}^{b}_{\mathrm{Lat}}$ containing extra constants $\bot$ and $\top$.

We consider GLat extended with the rules:

$$\frac{}{\bot \leq t}\ (\bot\Rightarrow) \qquad \text{and} \qquad \frac{}{s \leq \top}\ (\Rightarrow\top).$$

Theorem

(a) *φ_{bd} preserves the set of $\mathcal{L}_{\mathrm{Lat}}$-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A} \textit{ is bounded}\})$.

Example: Boundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **boundedness**:

$$\varphi_{bd} = (\exists x)(\exists y)(\forall z)((x \leq z) \sqcap (z \leq y)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{bd}) = (\forall z)((\bot \leq z) \sqcap (z \leq \top))$$

in a language $\mathcal{L}^{b}_{\mathrm{Lat}}$ containing extra constants $\bot$ and $\top$.

We consider GLat extended with the rules:

$$\frac{}{\bot \leq t}\ (\bot\Rightarrow) \qquad \text{and} \qquad \frac{}{s \leq \top}\ (\Rightarrow\top).$$

Theorem

(a) φ_{bd} *preserves the set of* $\mathcal{L}_{\mathrm{Lat}}$*-equations in* Lat.

(b) $\mathsf{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathsf{Lat} \mid \mathbf{A} \textit{ is bounded}\})$.

Example: Boundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **boundedness**:

$$\varphi_{bd} = (\exists x)(\exists y)(\forall z)((x \leq z) \sqcap (z \leq y)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{bd}) = (\forall z)((\bot \leq z) \sqcap (z \leq \top))$$

in a language $\mathcal{L}^{b}_{\mathrm{Lat}}$ containing extra constants $\bot$ and $\top$.

We consider GLat extended with the rules:

$$\frac{}{\bot \leq t}\ (\bot\Rightarrow) \qquad \text{and} \qquad \frac{}{s \leq \top}\ (\Rightarrow\top).$$

Theorem

(a) φ_{bd} *preserves the set of* $\mathcal{L}_{\mathrm{Lat}}$*-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A} \textit{ is bounded}\})$.

Example: Unboundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **unboundedness**:

$$\varphi_{unbd} = (\forall x)(\exists y)(\exists z)(\neg(x \le y) \sqcap \neg(z \le x)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{unbd}) = (\forall x)(\neg(x \le \downarrow x) \sqcap \neg(\uparrow x \le x))$$

in a language $\mathcal{L}^{u}_{\mathrm{Lat}}$ with extra unary function symbols $\downarrow$ and $\uparrow$.

We consider GLat extended with the rules:

$$\frac{u \le \downarrow u}{s \le t}\ (\le\downarrow) \qquad \text{and} \qquad \frac{\uparrow u \le u}{s \le t}\ (\uparrow\le).$$

Theorem

(a) *φ_{unbd} preserves the set of $\mathcal{L}_{\mathrm{Lat}}$-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A}$ *is unbounded*$\})$.

Example: Unboundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **unboundedness**:

$$\varphi_{unbd} = (\forall x)(\exists y)(\exists z)(\neg(x \le y) \sqcap \neg(z \le x)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{unbd}) = (\forall x)(\neg(x \le \downarrow x) \sqcap \neg(\uparrow x \le x))$$

in a language $\mathcal{L}^u_{\mathrm{Lat}}$ with extra unary function symbols $\downarrow$ and $\uparrow$.

We consider GLat extended with the rules:

$$\frac{u \le \downarrow u}{s \le t}\ (\le\downarrow) \qquad \text{and} \qquad \frac{\uparrow u \le u}{s \le t}\ (\uparrow\le).$$

Theorem

(a) φ_{unbd} *preserves the set of* $\mathcal{L}_{\mathrm{Lat}}$*-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A} \textit{ is unbounded}\})$.

Example: Unboundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **unboundedness**:

$$\varphi_{unbd} = (\forall x)(\exists y)(\exists z)(\neg(x \le y) \sqcap \neg(z \le x)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{unbd}) = (\forall x)(\neg(x \le {\downarrow} x) \sqcap \neg({\uparrow} x \le x))$$

in a language $\mathcal{L}^u_{\mathrm{Lat}}$ with extra unary function symbols $\downarrow$ and $\uparrow$.

We consider GLat extended with the rules:

$$\frac{u \le {\downarrow} u}{s \le t}\ (\le\downarrow) \qquad \text{and} \qquad \frac{{\uparrow} u \le u}{s \le t}\ (\uparrow\le).$$

Theorem

(a) φ_{unbd} *preserves the set of* $\mathcal{L}_{\mathrm{Lat}}$*-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A} \textit{ is unbounded}\})$.

Example: Unboundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **unboundedness**:

$$\varphi_{unbd} = (\forall x)(\exists y)(\exists z)(\neg(x \leq y) \sqcap \neg(z \leq x)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{unbd}) = (\forall x)(\neg(x \leq \downarrow x) \sqcap \neg(\uparrow x \leq x))$$

in a language $\mathcal{L}^{u}_{\mathrm{Lat}}$ with extra unary function symbols $\downarrow$ and $\uparrow$.

We consider GLat extended with the rules:

$$\frac{u \leq \downarrow u}{s \leq t}\ (\leq\downarrow) \qquad \text{and} \qquad \frac{\uparrow u \leq u}{s \leq t}\ (\uparrow\leq).$$

Theorem

(a) *φ_{unbd} preserves the set of $\mathcal{L}_{\mathrm{Lat}}$-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A}$ *is unbounded*$\})$.

Example: Unboundedness in Lattices

Consider the following $\mathcal{L}_{\mathrm{Lat}}$-sentence for expressing **unboundedness**:

$$\varphi_{unbd} = (\forall x)(\exists y)(\exists z)(\neg(x \leq y) \sqcap \neg(z \leq x)).$$

Skolemizing this sentence gives

$$\mathrm{sk}(\varphi_{unbd}) = (\forall x)(\neg(x \leq \downarrow x) \sqcap \neg(\uparrow x \leq x))$$

in a language $\mathcal{L}^{u}_{\mathrm{Lat}}$ with extra unary function symbols $\downarrow$ and $\uparrow$.

We consider GLat extended with the rules:

$$\frac{u \leq \downarrow u}{s \leq t}\ (\leq\downarrow) \qquad \text{and} \qquad \frac{\uparrow u \leq u}{s \leq t}\ (\uparrow\leq).$$

Theorem

(a) φ_{unbd} *preserves the set of* $\mathcal{L}_{\mathrm{Lat}}$*-equations in* Lat.

(b) $\mathrm{Lat} = \mathbb{V}(\{\mathbf{A} \in \mathrm{Lat} \mid \mathbf{A} \textit{ is unbounded}\})$.

Density

Consider the variety G of **Gödel algebras** and the following $\mathcal{L}$-sentence φ expressing *linearity* and *density*:

$$(\forall x)(\forall y)(\exists z)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq z) \sqcup (z \leq y)) \Rightarrow (x \leq y))).$$

Skolemizing, we obtain the sentence

$$(\forall x)(\forall y)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq d(x,y)) \sqcup (d(x,y) \leq y)) \Rightarrow (x \leq y))).$$

in a language $\mathcal{L}^d$ containing an extra binary function symbol d.

Theorem

(a) *φ preserves the set of $\mathcal{L}$-equations in* G.

(b) $\mathrm{G} = \mathbb{V}(\{\mathbf{A} \in \mathrm{G} \mid \mathbf{A}$ *is linearly and densely ordered*$\})$.

Density

Consider the variety G of **Gödel algebras** and the following $\mathcal{L}$-sentence φ expressing *linearity* and *density*:

$$(\forall x)(\forall y)(\exists z)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq z) \sqcup (z \leq y)) \Rightarrow (x \leq y))).$$

Skolemizing, we obtain the sentence

$$(\forall x)(\forall y)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq d(x,y)) \sqcup (d(x,y) \leq y)) \Rightarrow (x \leq y))).$$

in a language $\mathcal{L}^d$ containing an extra binary function symbol d.

Theorem

(a) *φ preserves the set of $\mathcal{L}$-equations in* G.

(b) $\mathsf{G} = \mathbb{V}(\{\mathbf{A} \in \mathsf{G} \mid \mathbf{A} \textit{ is linearly and densely ordered}\})$.

Density

Consider the variety G of **Gödel algebras** and the following $\mathcal{L}$-sentence φ expressing *linearity* and *density*:

$$(\forall x)(\forall y)(\exists z)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq z) \sqcup (z \leq y)) \Rightarrow (x \leq y))).$$

Skolemizing, we obtain the sentence

$$(\forall x)(\forall y)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq d(x,y)) \sqcup (d(x,y) \leq y)) \Rightarrow (x \leq y))).$$

in a language $\mathcal{L}^d$ containing an extra binary function symbol d.

Theorem

(a) φ *preserves the set of $\mathcal{L}$-equations in* G.

(b) $\mathsf{G} = \mathbb{V}(\{\mathbf{A} \in \mathsf{G} \mid \mathbf{A}$ *is linearly and densely ordered*$\})$.

Density

Consider the variety G of **Gödel algebras** and the following $\mathcal{L}$-sentence φ expressing *linearity* and *density*:

$$(\forall x)(\forall y)(\exists z)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq z) \sqcup (z \leq y)) \Rightarrow (x \leq y))).$$

Skolemizing, we obtain the sentence

$$(\forall x)(\forall y)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq d(x,y)) \sqcup (d(x,y) \leq y)) \Rightarrow (x \leq y))).$$

in a language $\mathcal{L}^d$ containing an extra binary function symbol d.

Theorem

(a) *φ preserves the set of $\mathcal{L}$-equations in* G.

(b) $\mathsf{G} = \mathbb{V}(\{\mathbf{A} \in \mathsf{G} \mid \mathbf{A}$ *is linearly and densely ordered*$\})$.

Density

Consider the variety G of **Gödel algebras** and the following $\mathcal{L}$-sentence φ expressing *linearity* and *density*:

$$(\forall x)(\forall y)(\exists z)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq z) \sqcup (z \leq y)) \Rightarrow (x \leq y))).$$

Skolemizing, we obtain the sentence

$$(\forall x)(\forall y)(((x \leq y) \sqcup (y \leq x)) \sqcap (((x \leq d(x,y)) \sqcup (d(x,y) \leq y)) \Rightarrow (x \leq y))).$$

in a language $\mathcal{L}^d$ containing an extra binary function symbol d.

Theorem

(a) φ *preserves the set of* $\mathcal{L}$*-equations in* G.

(b) $\mathrm{G} = \mathbb{V}(\{\mathbf{A} \in \mathrm{G} \mid \mathbf{A} \textit{ is linearly and densely ordered}\})$.

A Proof System for Lattices

Axioms

$$\frac{}{t \Rightarrow t}\ (\text{ID})$$

Cut rule

$$\frac{s \Rightarrow u \quad u \Rightarrow t}{s \Rightarrow t}\ (\text{CUT})$$

Left rules

$$\frac{t_i \Rightarrow s}{t_1 \wedge t_2 \Rightarrow s}\ (\wedge\Rightarrow)_i \quad i = 1, 2$$

$$\frac{t_1 \Rightarrow s \quad t_2 \Rightarrow s}{t_1 \vee t_2 \Rightarrow s}\ (\vee\Rightarrow)$$

Right rules

$$\frac{s \Rightarrow t_1 \quad s \Rightarrow t_2}{s \Rightarrow t_1 \wedge t_2}\ (\Rightarrow\wedge)$$

$$\frac{s \Rightarrow t_i}{s \Rightarrow t_1 \vee t_2}\ (\Rightarrow\vee)_i \quad (\text{i} = 1,2)$$

A Sequent Calculus for Distributive Lattices

Axioms

$$\frac{}{\Gamma, t \Rightarrow t}\ (\text{ID}) \qquad \frac{}{\Gamma, \bot \Rightarrow t}\ (\bot\Rightarrow)$$

Cut rule

$$\frac{\Gamma_1 \Rightarrow u \quad \Gamma_2, u \Rightarrow t}{\Gamma_1, \Gamma_2 \Rightarrow t}\ (\text{CUT})$$

Left rules

$$\frac{\Gamma, t_i \Rightarrow u}{\Gamma, t_1 \wedge t_2 \Rightarrow u}\ (\wedge\Rightarrow)_i \quad i = 1, 2$$

$$\frac{\Gamma, t_1 \Rightarrow u \quad \Gamma, t_2 \Rightarrow u}{\Gamma, t_1 \vee t_2 \Rightarrow u}\ (\vee\Rightarrow)$$

Right rules

$$\frac{\Gamma \Rightarrow t_1 \quad \Gamma \Rightarrow t_2}{\Gamma \Rightarrow t_1 \wedge t_2}\ (\Rightarrow\wedge)$$

$$\frac{\Gamma \Rightarrow t_i}{\Gamma \Rightarrow t_1 \vee t_2}\ (\Rightarrow\vee)_i \quad (\text{i =1,2})$$

A Sequent Calculus for Heyting Algebras

Axioms

$$\frac{}{\Gamma, t \Rightarrow t}\ (\textsc{id}) \qquad \frac{}{\Gamma, \bot \Rightarrow t}\ (\bot\Rightarrow)$$

Cut rule

$$\frac{\Gamma_1 \Rightarrow u \quad \Gamma_2, u \Rightarrow t}{\Gamma_1, \Gamma_2 \Rightarrow t}\ (\textsc{cut})$$

Left rules

$$\frac{\Gamma, t_i \Rightarrow u}{\Gamma, t_1 \wedge t_2 \Rightarrow u}\ (\wedge\Rightarrow)_i \quad i = 1, 2$$

$$\frac{\Gamma, t_1 \Rightarrow u \quad \Gamma, t_2 \Rightarrow u}{\Gamma, t_1 \vee t_2 \Rightarrow u}\ (\vee\Rightarrow)$$

$$\frac{\Gamma \Rightarrow t \quad \Gamma, s \Rightarrow u}{\Gamma, t \to s \Rightarrow u}\ (\to\Rightarrow)$$

Right rules

$$\frac{\Gamma \Rightarrow t_1 \quad \Gamma \Rightarrow t_2}{\Gamma \Rightarrow t_1 \wedge t_2}\ (\Rightarrow\wedge)$$

$$\frac{\Gamma \Rightarrow t_i}{\Gamma \Rightarrow t_1 \vee t_2}\ (\Rightarrow\vee)_i \quad \text{(i =1,2)}$$

$$\frac{\Gamma, t \Rightarrow s}{\Gamma \Rightarrow t \to s}\ (\Rightarrow\to)$$

A Hypersequent Calculus for Heyting Algebras

Axioms

$$\frac{}{\mathcal{G} \mid \Gamma, t \Rightarrow t} \text{(ID)} \qquad \frac{}{\mathcal{G} \mid \Gamma, \bot \Rightarrow t} (\bot\Rightarrow)$$

Cut rule

$$\frac{\mathcal{G} \mid \Gamma_1 \Rightarrow u \quad \mathcal{G} \mid \Gamma_2, u \Rightarrow t}{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t} \text{(CUT)}$$

Left rules

$$\frac{\mathcal{G} \mid \Gamma, t_i \Rightarrow u}{\mathcal{G} \mid \Gamma, t_1 \wedge t_2 \Rightarrow u} (\wedge\Rightarrow)_i \quad i = 1, 2$$

$$\frac{\mathcal{G} \mid \Gamma, t_1 \Rightarrow u \quad \mathcal{G} \mid \Gamma, t_2 \Rightarrow u}{\mathcal{G} \mid \Gamma, t_1 \vee t_2 \Rightarrow u} (\vee\Rightarrow)$$

$$\frac{\mathcal{G} \mid \Gamma \Rightarrow t \quad \mathcal{G} \mid \Gamma, s \Rightarrow u}{\mathcal{G} \mid \Gamma, t \to s \Rightarrow u} (\to\Rightarrow)$$

Right rules

$$\frac{\mathcal{G} \mid \Gamma \Rightarrow t_1 \quad \mathcal{G} \mid \Gamma \Rightarrow t_2}{\mathcal{G} \mid \Gamma \Rightarrow t_1 \wedge t_2} (\Rightarrow\wedge)$$

$$\frac{\mathcal{G} \mid \Gamma \Rightarrow t_i}{\mathcal{G} \mid \Gamma \Rightarrow t_1 \vee t_2} (\Rightarrow\vee)_i \quad \text{(i =1,2)}$$

$$\frac{\mathcal{G} \mid \Gamma, t \Rightarrow s}{\mathcal{G} \mid \Gamma \Rightarrow t \to s} (\Rightarrow\to)$$

We obtain a hypersequent calculus GG for **Gödel algebras** by adding the **communication** rule:

$$\frac{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow s \quad \mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t}{\mathcal{G} \mid \Gamma_1 \Rightarrow s \mid \Gamma_2 \Rightarrow t} \; (\textsc{com})$$

A. Avron. Hypersequents, logical consequence and intermediate logics for concurrency. *Annals of Mathematics and Artificial Intelligence*, 4(3–4):225–248, 1991.

A Hypersequent Calculus for Gödel Algebras

We obtain a hypersequent calculus GG for **Gödel algebras** by adding the **communication** rule:

$$\frac{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow s \quad \mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t}{\mathcal{G} \mid \Gamma_1 \Rightarrow s \mid \Gamma_2 \Rightarrow t} \text{ (COM)}$$

A. Avron. Hypersequents, logical consequence and intermediate logics for concurrency. *Annals of Mathematics and Artificial Intelligence*, 4(3–4):225–248, 1991.

A Hypersequent Calculus for Gödel Algebras

We obtain a hypersequent calculus GG for **Gödel algebras** by adding the **communication** rule:

$$\frac{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow s \quad \mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t}{\mathcal{G} \mid \Gamma_1 \Rightarrow s \mid \Gamma_2 \Rightarrow t} \ (\textsc{com})$$

A. Avron. Hypersequents, logical consequence and intermediate logics for concurrency. *Annals of Mathematics and Artificial Intelligence*, 4(3–4):225–248, 1991.

The Density Rule

Let GG^{D} be GG extended with:

$$\frac{\mathcal{G} \mid \Gamma_1 \Rightarrow x \mid \Gamma_2, x \Rightarrow t}{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t} \ (\textsc{density})$$

where x does not occur in the conclusion.

Theorem

(a) $\vdash_{\mathrm{GG}^{\mathrm{D}}} \varphi \Rightarrow \psi$ *iff* $\varphi \leq \psi$ *in all dense linearly ordered Gödel algebras.*

(b) GG^{D} *admits density elimination.*

M. Baaz and R. Zach. Hypersequents and the proof theory of intuitionistic fuzzy logic. *Proceedings of CSL 2000.* LNCS 1862:187–201, 2000.

G. Metcalfe and F. Montagna. Substructural fuzzy logics. *Journal of Symbolic Logic* 72(3):834–864, 2007.

A. Ciabattoni and G. Metcalfe. Density elimination. *Theoretical Computer Science* 403:328–346, 2008.

Let GG^{D} be GG extended with:

$$\frac{\mathcal{G} \mid \Gamma_1 \Rightarrow x \mid \Gamma_2, x \Rightarrow t}{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t} \ (\textsc{density})$$

where x does not occur in the conclusion.

Theorem

(a) $\vdash_{\mathrm{GG}^{\mathrm{D}}} \varphi \Rightarrow \psi$ *iff* $\varphi \leq \psi$ *in all dense linearly ordered Gödel algebras.*

(b) GG^{D} *admits density elimination.*

M. Baaz and R. Zach. Hypersequents and the proof theory of intuitionistic fuzzy logic. *Proceedings of CSL 2000.* LNCS 1862:187–201, 2000.

G. Metcalfe and F. Montagna. Substructural fuzzy logics. *Journal of Symbolic Logic* 72(3):834–864, 2007.

A. Ciabattoni and G. Metcalfe. Density elimination. *Theoretical Computer Science* 403:328–346, 2008.

The Density Rule

Let GG^{D} be GG extended with:

$$\frac{\mathcal{G} \mid \Gamma_1 \Rightarrow x \mid \Gamma_2, x \Rightarrow t}{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t} \ (\textsc{density})$$

where x does not occur in the conclusion.

Theorem

(a) $\vdash_{\mathrm{GG}^{\mathrm{D}}} \varphi \Rightarrow \psi$ *iff* $\varphi \leq \psi$ *in all dense linearly ordered Gödel algebras.*

(b) GG^{D} *admits density elimination.*

M. Baaz and R. Zach. Hypersequents and the proof theory of intuitionistic fuzzy logic. *Proceedings of CSL 2000.* LNCS 1862:187–201, 2000.

G. Metcalfe and F. Montagna. Substructural fuzzy logics. *Journal of Symbolic Logic* 72(3):834–864, 2007.

A. Ciabattoni and G. Metcalfe. Density elimination. *Theoretical Computer Science* 403:328–346, 2008.

The Density Rule

Let GG^{D} be GG extended with:

$$\frac{\mathcal{G} \mid \Gamma_1 \Rightarrow x \mid \Gamma_2, x \Rightarrow t}{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t} \ (\textsc{density})$$

where x does not occur in the conclusion.

Theorem

(a) $\vdash_{\mathrm{GG}^{\mathrm{D}}} \varphi \Rightarrow \psi$ *iff* $\varphi \leq \psi$ *in all dense linearly ordered Gödel algebras.*

(b) GG^{D} *admits density elimination.*

M. Baaz and R. Zach. Hypersequents and the proof theory of intuitionistic fuzzy logic. *Proceedings of CSL 2000*. LNCS 1862:187–201, 2000.

G. Metcalfe and F. Montagna. Substructural fuzzy logics. *Journal of Symbolic Logic* 72(3):834–864, 2007.

A. Ciabattoni and G. Metcalfe. Density elimination. *Theoretical Computer Science* 403:328–346, 2008.

What Can Go Wrong With Adding Density?

A calculus GCL for **classical logic** is obtained by extending GG with

$$\frac{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t}{\mathcal{G} \mid \Gamma_1 \Rightarrow s \mid \Gamma_2 \Rightarrow t} \ (\textsc{split})$$

But then for *any* term t, we have a derivation in $\mathrm{GCL}^{\mathrm{D}}$:

$$\dfrac{\dfrac{\overline{x \Rightarrow x} \ (\textsc{id})}{\Rightarrow x \mid x \Rightarrow t} \ (\textsc{split})}{\Rightarrow t} \ (\textsc{density})$$

I.e., $\mathrm{GCL}^{\mathrm{D}}$ is *trivial* – as it should be.

What Can Go Wrong With Adding Density?

A calculus GCL for **classical logic** is obtained by extending GG with

$$\frac{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t}{\mathcal{G} \mid \Gamma_1 \Rightarrow s \mid \Gamma_2 \Rightarrow t} \; (\textsc{split})$$

But then for *any* term t, we have a derivation in $\mathrm{GCL}^{\mathrm{D}}$:

$$\cfrac{\cfrac{\cfrac{}{x \Rightarrow x}\;(\textsc{id})}{\Rightarrow x \mid x \Rightarrow t}\;(\textsc{split})}{\Rightarrow t}\;(\textsc{density})$$

I.e., $\mathrm{GCL}^{\mathrm{D}}$ is *trivial* – as it should be.

What Can Go Wrong With Adding Density?

A calculus GCL for **classical logic** is obtained by extending GG with

$$\frac{\mathcal{G} \mid \Gamma_1, \Gamma_2 \Rightarrow t}{\mathcal{G} \mid \Gamma_1 \Rightarrow s \mid \Gamma_2 \Rightarrow t} \ (\text{SPLIT})$$

But then for *any* term t, we have a derivation in $\mathrm{GCL}^{\mathrm{D}}$:

$$\dfrac{\dfrac{\overline{x \Rightarrow x}\ (\text{ID})}{\Rightarrow x \mid x \Rightarrow t}\ (\text{SPLIT})}{\Rightarrow t}\ (\text{DENSITY})$$

I.e., $\mathrm{GCL}^{\mathrm{D}}$ is *trivial* – as it should be.

Concluding Remarks

- Admissible rules play a subtle but crucial role in logic and algebra.
- Algebraically, admissibility corresponds to validity in free algebras.
- However, there are interesting examples that fit better into a first-order framework.
- Establishing the admissibility of a rule can be useful.

Concluding Remarks

- Admissible rules play a subtle but crucial role in logic and algebra.
- Algebraically, admissibility corresponds to validity in free algebras.
- However, there are interesting examples that fit better into a first-order framework.
- Establishing the admissibility of a rule can be useful.

Concluding Remarks

- Admissible rules play a subtle but crucial role in logic and algebra.
- Algebraically, admissibility corresponds to validity in free algebras.
- However, there are interesting examples that fit better into a first-order framework.
- Establishing the admissibility of a rule can be useful.

Concluding Remarks

- Admissible rules play a subtle but crucial role in logic and algebra.
- Algebraically, admissibility corresponds to validity in free algebras.
- However, there are interesting examples that fit better into a first-order framework.
- Establishing the admissibility of a rule can be useful.