Encyclopaedia of Proof Systems

http://ProofSystem.github.io/Encyclopedia/

Preface

2^{nd} Edition

In December 2014, I had the honor to submit one of the first entries to the **Encyclopedia of Proof Systems** at the request of Bruno Woltzenlogel Paleo. Less than one year later, the Encyclopedia already counted 64 entries, which were presented at a poster session during CADE-25. After this successful event, Bruno has kindly invited me to co-organize with him a workshop during the Brasilia Spring on Automated Deduction, formed by the conferences TABLEAUX, ITP (Interactive Theorem Proving) and FroCoS (Frontiers of Combining Systems), in September 2017.

The EPS workshop comprised of presentations of new entries by the authors, an open discussion about the Encyclopedia (suggestion of improvements and long-term goals), and a hands-on session for active contributions. The workshop was accompanied by a poster session where the newest entries were displayed. We would like to thank Katalin Bimbó, Serenella Cerrito, Clare Dixon, Reiner Hähnle, Rolf Hennicker, Ullrich Hustadt, Björn Lellmann, João Marcos, Renate Schmidt, and Yoni Zohar for participating in the workshop and contributing to the discussions. There was a wide variety of interesting and accessible talks about proof systems in different areas, and many suggestions of new entries and features for the Encyclopedia. We would also like to thank Cláudia Nalon for all her support with the logistics of the workshop and for organizing a great conference.

In total, 34 new entries by 32 authors were submitted to the Encyclopedia of Proof Systems. Once again, a wide range of calculi is represented, such as resolution, sequent, axiomatic, display, and natural deduction. In addition to different logics (e.g., temporal, paraconsistent, hybrid, epistemic, etc.), there are calculi for different systems as well, such as unification and structured specifications. We are particularly happy to include in this new edition Hilbert's, Bernay's and Ackermann's calculi, thanks to Richard Zach. Many people had expressed that those historically important systems deserved an entry in the Encyclopedia. Now they finally have a place here.

This second edition of the Encyclopedia of Proof Systems book extends the first edition with the 34 new entries. Additionally, with the aim of encouraging practical applications of proof systems, a new meta-data tag for implementations or formalizations of an entry is now available.

New proof systems are proposed each day, so the Encyclopedia will always be open for new contributions. With almost 100 entries on the most diverse systems, this effort of knowledge organization can only succeed as a joint effort of the community. We are grateful for the support we have received so far and hope the Encyclopedia continues to grow in the years to come.

December 2017 *Giselle Reis*

Preface

1st Edition

The **Encyclopedia of Proof Systems** aims at providing a reliable, technically accurate, historically informative, concise, uniform and convenient central repository of proof systems for various logics. The goal is to facilitate the exchange of information among logicians, in order to foster and accelerate the development of proof theory and automated deduction.

Preparatory work for the creation of the Encyclopedia, such as the implementation of the LaTeX template and the setup of the Github repository, started in October 2014, triggered by the call for workshop proposals for the 25th Conference on Automated Deduction (CADE). Christoph Benzmüller, CADE's conference chair, and Jasmin Blanchette, CADE's workshop co-chair, encouraged me to submit a workshop proposal and supported my alternative idea to organize instead a special poster session based on encyclopedia entries. I am thankful for their encouragement and support.

In December 2014, Björn Lellmann, Giselle Reis and Martin Riener kindly accepted my request to beta-test the template and the instructions I had created. They submitted the first few example entries to the encyclopedia and provided valuable feedback, for which I am grateful. Their comments were essential for improving the templates and instructions before the public announcement of the encyclopedia.

In July 2015, Julian Röder's assistance was essential for the successful organization of the poster session at CADE. Cezary Kaliszyk and Andrei Paskevitch kindly allowed me to organize a discussion session as part of the Proof Exchange for Theorem Proving (PxTP) workshop, where the participants provided useful feedback and many ideas for improvements. Discussions with Lev Beklemishev, Björn Lellmann, Tomer Libal, Roman Kuznets, Sergei Soloviev, Valeria de Paiva and Anna Zamansky also brainstormed many ideas for improving the organization and structure of the encyclopedia.

In the few months that preceded CADE, as many as 64 entries, spanning a wide range of deduction styles and logics, have been submitted by 34 contributors. Although large for a single event, these numbers are still small compared to the vast number of proof systems that have been invented and to the number of people who work on logical calculi nowadays. Therefore, this community-wide initiative is only at the beginning and the encyclopedia intends to remain open to submissions for a long time.

October 2016 *Bruno Woltzenlogel Paleo*

Contents

Part I *Proof Systems*

1 Principia Mathematica 3

2 Hilbert's Axiomatic Calculus 5

3 Bernays's Propositional Calculus 6

4 Hilbert and Ackermann's Calculus 7

5 Intuitionistic Natural Deduction NJ 8

6 Classical Sequent Calculus LK 9

7 Intuitionistic Sequent Calculus LJ 10

8 Epsilon Calculus 11

9 Kleene's Classical G3 System 12

10 Kleene's Intuitionistic G3 System 13

11 Multi-Conclusion Sequent Calculus LJ′ 14

12 Lambek Calculus 15

13 Resolution 16

14 First-Order Unification 17

15 Ordered Resolution 18

16 Paramodulation 19

17 (Unfailing) Completion 20

18 Second Order λ-Calculus (System F) 21

19 Higher-Order Pre-Unification 22
20 Resolution for Modal Logic K (RK) 23
21 Expansion Proofs 24
22 Bledsoe's Natural Deduction - Prover 25
23 Natural Knowledge Bases - Muscadet 26
24 Intuitionistic Linear Logic (ILL) 27
25 Linear Sequent Calculus LL 28
26 Proof Nets for MLL$^-$ 29
27 Entailment for Structured Specifications 31
28 Pure Type Systems 32
29 Full Intuitionistic Linear Logic (FILL) 33
30 Signed Analytic Calculi for Finite-Valued Logics 35
31 Superposition 36
32 Saturation With Redundancy 38
33 Constructive Classical Logic LC 39
34 Refinement of Structured Specifications 40
35 Resolution for Propositional Linear Time Temporal Logic (LTL) 41
36 Two-sided Linear Sequent Calculus 42
37 Constraint Superposition 43
38 Hierarchic Superposition 44
39 Classical Natural Deduction ($\lambda\mu$-calculus) 45
40 Logic of Epistemic Inconsistency 47
41 Typed LF for Type Theories 49
42 $\overline{\lambda}$-calculus 50
43 Full Intuitionistic Logic (FIL) 52
44 T[C] (LF with Coercive Subtyping) 53
45 Sequent Calculus G3c 55
46 Cancellative Superposition 56

47 Graph-based tableaux for modal logics 58
48 Synthetic Tableaux 60
49 Polarized Linear Sequent Calculus LLP 61
50 LK$_{\mu\tilde{\mu}}$ 62
51 Constructive Modal Logic S4 (CS4) 64
52 Hybrid Logic (HL) 65
53 Sequent Calculus TC 66
54 Intuitionistic Hybrid Logic (IHL) 67
55 Resolution for Monodic First-Order Temporal Logic 68
56 Model Evolution 69
57 Resolution for Monodic First-Order Temporal Logic 70
58 Socratic Proofs for CPL 71
59 Socratic Proofs for FOL 72
60 Socratic Proofs for Modal Propositional K 73
61 Socratic Proofs for Modal Propositional Logics 74
62 LK$\mu\tilde{\mu}$ in sequent-free tree form 75
63 Conditional Labelled Sequent Calculi SeqS 77
64 Preferential Tableau Calculi $\mathcal{T}P^{\mathbf{T}}$ 78
65 HO Sequent Calculi $\mathcal{G}_{\beta}$ and $\mathcal{G}_{\beta\mathfrak{N}}$ 80
66 Extensional HO RUE-Resolution 81
67 Focused LK 83
68 Focused LJ 85
69 $\lambda\Pi$-Calculus Modulo 87
70 Ordered Fine-Grained Resolution with Selection for Monodic First-Order Temporal Logic . 89
71 Resolution for Computation Tree Logic (CTL) 90
72 Untyped λ Reduction 91
73 Sequent Calculi for Paraconsistent Logics 92
74 Counterfactual Sequent Calculi I 93

75 Counterfactual Sequent Calculi II 94
76 Conditional Nested Sequents $\mathcal{NS}$ 95
77 FILL Deep Nested Sequent Calculus 96
78 Contextual Natural Deduction 98
79 IR 99
80 Sequent Calculus for Superintuitionistic Modal Logic 100
81 Erotetic Dual Resolution for Classical Propositional Logic 102
82 Erotetic Dual Resolution for mbC 103
83 Multi-type Sequent Calculi Mt.SC 104
84 Modal Natural Deduction 106
85 Polynomial Ring Calculus with Operators 107
86 Conflict Resolution 109
87 Sequent Systems for Negative Modalities 110
88 Update Logic UL 111
89 Proper Multi-type Display Calculus for Inquisitive Logic MtD.InqL 112
90 Multi-type Sequent Calculus for Dynamic Epistemic Logic MtD.DEL 114
91 Proper Multi-type Display Calculus for Lattice Logic MtD.LatL 116
92 Proper Multi-type Display Calculus for semi De Morgan Logic MtD.SDM 118
93 Epsilon-Sound Sequent Calculus LJ* 120
94 Sequent Calculus for Logic of Partial Quasiary Predicates 121

Part II ***Indexes***

Proof Systems Grouped by Logics 125
Proof Systems Grouped by Type 127
Entry Authors 128
Proof Systems' Authors 129
Implementations of Proof Systems 130

Part I
Proof Systems

Principia Mathematica (1910)

Axioms:

(1) $(p \vee p) \supset p$ (2) $q \supset (p \vee q)$ (3) $(p \vee q) \supset (q \vee p)$ (4) $p \vee (q \vee r) \supset . q \vee (p \vee r)$

(5) $(q \supset r) \supset . (p \vee q) \supset (p \vee r)$

(6) Universal Instantiation: $\forall \nu_\tau \psi \supset \psi'$

where ψ' is like ψ except for having a term ν of r - type τ substituted for ν_τ in ψ.

(7) Comprehension: $\exists \phi \forall x_1 \ldots \forall x_m [\phi(x_1, \ldots, x_m) \equiv \psi], (\phi$ not free in $\psi)$

(8) Axiom of Reducibility: $\forall \psi \exists \phi \forall x_1 \ldots \forall x_m [\phi!(x_1, \ldots, x_m) \equiv \psi(x_1, \ldots, x_m)]$

Rules:

(1) Modus Ponens:

$$\frac{\phi \quad \phi \supset \psi}{\psi}$$

(2) Substitution for individual, functional and propositional variables of each type.

(3) Universal Generalization:

$$\frac{\phi}{\forall \nu_\tau \phi}$$

Definition of Identity:

$x = y \quad =_{df} \quad \forall \phi [\phi!(x) \equiv \phi!(y)]$, (for $\phi!$ a predicative function)

Clarifications: The *primitive connectives* are $\vee$ and $\sim$. The connective $p \supset q$ is defined as $\sim p \vee q$ and the conjunction $p \,.\, q$ as $\sim(\sim p \vee \sim q)$.

r-types: The system of symbols for *r-types* (*ramified types*) and the assignment of r-types to variables for different entities (individuals and functions) is as follows: ι is the r-type for an *individual*. Where $\tau_1 \ldots, \tau_m$ are any r-types, then $(\tau_1 \ldots, \tau_m)/n$ is the r-type of any m-ary propositional function of *level n*, which has arguments of r- types $\tau_1 \ldots, \tau_m$, respectively. The *order* of an individual is 0. The *order* of a function of r-type $\tau_1 \ldots, \tau_m/n$ is $n + N$ where N is the greatest of the order of the arguments $\tau_1 \ldots, \tau_m$.

Typical ambiguity: All statements of axioms and rules apply in each r-type.

Restriction on comprehension principle: The comprehension principle avoid paradox by imposing the following restriction on the function ϕ: ϕ is a functional variable of r-type $(\beta_1, \beta_2, \ldots, \beta_m)/n$ and $x_1, \ldots, x_m$ are distinct variables of r -types $\beta_1, \beta_2, \ldots, \beta_m$, and the bound variables of A are all of order less than the order of ϕ and the free variables of A and constants occurring in A are all of order not greater than the order of ϕ.

Predicative functions: The notation $\phi!$ indicates that the function ϕ is *predicative*, that is, the variables $x_1, \ldots, x_m$ are of r-types $\beta_1, \beta_2, \ldots, \beta_m$ respectively and ϕ is of r-type $(\beta_1, \beta_2, \ldots, \beta_m)/1$ and ψ is of r-type $(\beta_1, \beta_2, \ldots, \beta_m)/n$.

The "Multiplicative Axiom" (Axiom of Choice) and "Axiom of Infinity":. These sentences are not axioms of the formal system, but rather appear in theorems as the antecedents of conditional theorems when used to derive the consequent.

Adequacy of the Definition of Identity: The Axiom of Reducibility guarantees that if for any ψ of any r-type, $\exists \psi \sim [\psi(x) \equiv \psi(y)]$, then $x \neq y$.

Entry 1 by: Bernard Linsky

History: The system was published in [1]. This formulation, and most notably, the system of *r-types* follows that of Church [3]. Church also adds a Comprehension principle, and rule of Substitution, neither of which are explicit in [1]. See [4] for an account of Whitehead and Russell's notation and [5] for a survey of the contents of [1].

Remarks: In 1926 Paul Bernays [2] showed that the axioms can be reduced by one, as axiom 4 can be proved from 1,2,3 and 5.

---

[1] Alfred N. Whitehead and Bertrand A. Russel. *Principia Mathematica*. Vol. I in 1910, Vol. II in 1912, and Vol. III 1913. Second Edition from 1925 to 27. Cambridge University Press, 1910–1913.

[2] Alonzo Church. "Axiomatische Untersuchung des AussagenKalkuls der Principia Mathematica". In: *Mathematische Zeitschrift* 25 (1926), pp. 305–320.

[3] Alonzo Church. "Comparison of Russell's Resolution of the Semantical Antinomies with That of Tarski". In: *Journal of Symbolic Logic* 41 (1976), pp. 747–760.

[4] Bernard Linsky. "The Notation in Principia Mathematica". In: *The Stanford Encyclopedia of Philosophy* (2016). URL: `https://plato.stanford.edu/archives/fall2016/entries/pm-notation/`.

[5] Bernard Linsky and Andrew David Irvine. "Principia Mathematica". In: *The Stanford Encyclopedia of Philosophy* (2019).

Hilbert's Axiomatic Calculus (1917)

Axioms:

I.
1) $X \vee X \to X$
2) $X \to X \vee Y$
3) $X \vee Y \to Y \vee X$
4) $X \vee (Y \vee Z) \to (X \vee Y) \vee Z$
5) $(X \to Y) \to (Z \vee X \to Z \vee Y)$

II.
1) $(x)Z \to Z$
2) $(x)F(x) \to (Ex)F(x)$
3) $(x)(Z \vee F(x)) \to ((x)Z \vee (x)F(x))$
4) $(x)(F(x) \to G(x)) \to ((x)F(x) \to (x)G(x))$
5) $(x)(y)F(x,y) \to (y)(x)F(x,y)$
6) $(x)(y)F(x,y) \to (x)F(x,x)$

Rules:
1. Renaming of bound variables
2. Substitution for propositional and predicate variables
3. Universal instantiation, i.e.,
$$\frac{(x)\alpha(x)}{\alpha(a)}$$
4. Universal closure of additional argument places, i.e.,
$$\frac{\alpha(X)}{(x)\alpha(F(x))} \qquad \frac{\alpha(F(x_1,\ldots,x_n))}{(y)\alpha(F(x_1,\ldots,x_n,y))}$$
5. Modus ponens, i.e.,
$$\frac{\alpha \quad \alpha \to \beta}{\beta}$$

Clarifications: Hilbert used the symbol $\times$ for disjunction $\vee$, $+$ for conjunction $\wedge$, and overlining $\overline{\ }$ for negation. $\alpha \to \beta$ is an abbreviation for $\overline{\alpha} \vee \beta$, and $\alpha \wedge \beta$ for $\overline{\overline{\alpha} \vee \overline{\beta}}$. The usual conventions about operator precedence apply. (x) is a universal quantifier and (Ex) an existential quantifier. The system is formulated in a language that distinguishes between propositional and predicate constants and variables. In particular, the axioms are *not* understood as schemas. The substitution rule allows the replacement of propositional variables by a formula that contains no object variable free, and a predicate variable with n arguments $x_1, \ldots, x_n$ by an expression in which all and only $x_1, \ldots, x_n$ occur free.

History: The system was first presented in lecture notes by Paul Bernays to Hilbert's course "Principles of Mathematics", taught in the Winter term 1917–18; see [1]. The system is based on the axiomatic proof system of Whitehead and Russell's *Principia Mathematica*, but restricts the language to first order. It adds the explicit rules of substitution and renaming, and avoids the use of free variables in its axioms and theorems.

[1] David Hilbert. "Prinzipien der Mathematik". In: *David Hilbert's Lectures on the Foundations of Arithmetic and Logic, 1917–1933*. Ed. by William Ewald and Wilfried Sieg. Berlin and Heidelberg: Springer, 2013, pp. 59–221.

Entry 2 by: Richard Zach

Bernays's Propositional Calculus (1918)

Axioms:

1)	$\overline{X\overline{X}}X$	1)	$X \vee X \to X$
2)	$\overline{X}(XY)$	2)	$X \to X \vee X$
3)	$\overline{X\overline{Y}}(YX)$	3)	$X \vee Y \to Y \vee X$
4)	$\overline{X(Y\overline{Z})}((XY)Z)$	4)	$X \vee (Y \vee Z) \to (X \vee Y) \vee Z$
5)	$(\overline{\overline{X}Y})(\overline{Z\overline{X}}(ZY))$	5)	$(X \to Y) \to (Z \vee X \to Z \vee Y)$

Rules:
a. Substitution for propositional variables
b. Modus ponens, i.e.,

$$\frac{\alpha \quad \alpha \to \beta}{\beta}$$

Clarifications: Bernays used juxtaposition for disjunction $\vee$, + for conjunction $\wedge$, and overlining $\overline{\ }$ for negation. The axioms on the left are official, those on the right use the abbreviation $\alpha \to \beta$ for $\overline{\alpha}\beta$, i.e., $\overline{\alpha} \vee \beta$. The substitution rule allows the replacement of propositional variables by any expression.

History: The axioms are a slight variation on the propositional fragment of Whitehead and Russell's *Principia Mathematica* due to Paul Bernays [1]. The system is noteworthy since Bernays was the first to prove completeness relative to standard truth value semantics, as well as decidability; see [3]. Bernays investigated systems in which axioms are replaced by rules, e.g., $\alpha \vdash \alpha \vee \beta$, including rules that operate on parts of a formula such as $\gamma(\alpha \vee \beta) \vdash \gamma(\beta \vee \alpha)$. He showed that a system with six rules and $X \to X$ as the only axiom is complete. He also showed that axiom (4) is provable from the others and the rest are independent (published in [2]).

---

[1] Paul Bernays. "Beiträge zur axiomatischen Behandlung des Logik-Kalküls". Bernays Nachlaß, WHS, ETH Zürich Archive, Hs 973.192. Edited in [**Hilbert2013**], pp. 222–271. Habilitationsschrift. Universität Göttingen, 1918.

[2] Paul Bernays. "Axiomatische Untersuchungen des Aussagen-Kalkuls der 'Principia Mathematica'". In: *Mathematische Zeitschrift* 25 (1926), pp. 305–20.

[3] Richard Zach. "Completeness before Post: Bernays, Hilbert, and the development of propositional logic". In: *Bulletin of Symbolic Logic* 5.3 (1999), pp. 331–366.

Entry 3 by: Richard Zach

Hilbert and Ackermann's Calculus (1928)

Axioms:

a) $X \vee X \to X$

b) $X \to X \vee Y$

c) $X \vee Y \to Y \vee X$

d) $(X \to Y) \to (Z \vee X \to Z \vee Y)$

e) $(x)F(x) \to F(y)$

f) $F(y) \to (Ex)F(x)$

Rules:

α. Substitution for object, propositional, and predicate variables

β. Generalization rules, i.e.,

$$\frac{\mathfrak{A} \to \mathfrak{B}(x)}{\mathfrak{A} \to (x)\mathfrak{B}(x)} \qquad \frac{\mathfrak{B}(x) \to \mathfrak{A}}{(Ex)\mathfrak{B}(x) \to \mathfrak{A}}$$

where x must not occur in $\mathfrak{A}$.

γ. Modus ponens, i.e.,

$$\frac{\mathfrak{A} \quad \mathfrak{A} \to \mathfrak{B}}{\mathfrak{A}}$$

Clarifications: As in {2}, the system is formulated in a language that distinguishes between propositional and predicate constants and variables, but theorems with free variables are now allowed. Fraktur letters are used as schematic metavariables. The substitution rule allows the replacement of individual variables by variables or constants, propositional variables by any formula, and predicate variables with n arguments $x_1, \ldots, x_n$ by a formula in which $x_1, \ldots, x_n$ occur free.

History: The system was published in [1]. It was the systems for which the problem of providing a completeness proof was first raised.

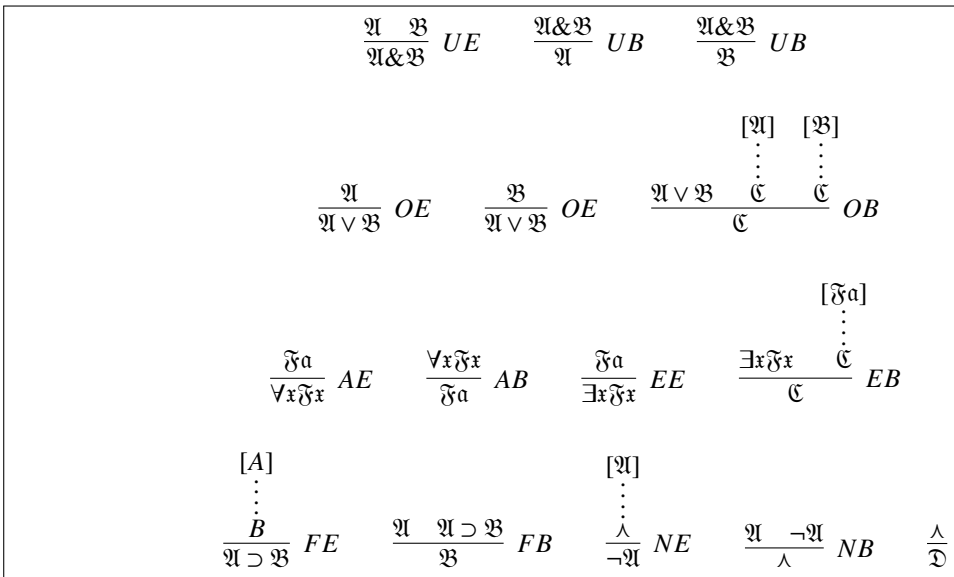
Remarks: This system was proved complete in [2].

[1] David Hilbert and Wilhelm Ackermann. *Grundzüge der theoretischen Logik*. Berlin: Springer, 1928.

[2] Kurt Gödel. "Die Vollständigkeit der Axiome des logischen Funktionenkalküls". In: *Monatshefte für Mathematik und Physik* 37 (1930), pp. 349–360.

Entry 4 by: Richard Zach

Intuitionistic Natural Deduction NJ (1935)



The eigenvariable $\mathfrak{a}$ of an *AE* must not occur in the formula designated in the schema by $\forall \mathfrak{x}\mathfrak{F}\mathfrak{x}$; nor in any assumption formula upon which that formula depends. The eigenvariable $\mathfrak{a}$ of an *EB* must not occur in the formula designated in the schema by $\exists \mathfrak{x}\mathfrak{F}\mathfrak{x}$; nor in any assumption formula upon which that formula depends, with the exception of the assumption formulae designated by $\mathfrak{F}\mathfrak{a}$.

Clarifications: The names of the rules are those originally given by Gentzen [1]:
U = und (and), *O* = oder (or), *A* = all, *E* = es-gibt (exists), *F* = folgt (follows),
N = nicht (not), *E* = Einführung (introduction), *B* = Beseitigung (elimination).

History: The main novelty introduced by Gentzen in this proof system is its *assumption* handling mechanism, which allows formal proofs to reflect more naturally the logical reasoning involved in mathematical proofs.

Remarks: In [1], completeness of **NJ** is proven by showing how to translate proofs in the Hilbert-style calculus **LHJ** to **NJ**-proofs, and soundness is proven by showing how to translate **NJ**-proofs to **LJ**-proofs {7}.

---

[1] Gerhard Gentzen. "Untersuchungen über das logische Schließen I". In: *Mathematische Zeitschrift* 39.1 (Dec. 1935), pp. 176–210.

Entry 5 by: Bruno Woltzenlogel Paleo

Classical Sequent Calculus LK (1935)

$$\frac{}{A \vdash A} \qquad \frac{\Gamma \vdash \Lambda, A \quad A, \Delta \vdash \Theta}{\Gamma, \Delta \vdash \Lambda, \Theta}\ cut$$

$$\frac{\Gamma \vdash \Theta}{A, \Gamma \vdash \Theta}\ w_l \qquad \frac{\Gamma \vdash \Theta}{\Gamma \vdash \Theta, A}\ w_r$$

$$\frac{\Gamma, B, A, \Delta \vdash \Theta}{\Gamma, A, B, \Delta \vdash \Theta}\ e_l \qquad \frac{A, A, \Gamma \vdash \Theta}{A, \Gamma \vdash \Theta}\ c_l \qquad \frac{\Gamma \vdash \Theta, B, A, \Delta}{\Gamma \vdash \Theta, A, B, \Delta}\ e_r \qquad \frac{\Gamma \vdash \Theta, A, A}{\Gamma \vdash \Theta, A}\ c_r$$

$$\frac{\Gamma \vdash \Theta, A}{\neg A, \Gamma \vdash \Theta}\ \neg_l \qquad \frac{A, \Gamma \vdash \Theta}{\Gamma \vdash \Theta, \neg A}\ \neg_r$$

$$\frac{A_i, \Gamma \vdash \Theta}{A_1 \wedge A_2, \Gamma \vdash \Theta}\ \wedge_l \qquad \frac{\Gamma \vdash \Theta, A \quad \Gamma \vdash \Theta, B}{\Gamma \vdash \Theta, A \wedge B}\ \wedge_r$$

$$\frac{A, \Gamma \vdash \Theta \quad B, \Gamma \vdash \Theta}{A \vee B, \Gamma \vdash \Theta}\ \vee_l \qquad \frac{\Gamma \vdash \Theta, A_i}{\Gamma \vdash \Theta, A_1 \vee A_2}\ \vee_r$$

$$\frac{\Gamma \vdash \Lambda, A \quad B, \Delta \vdash \Theta}{A \to B, \Gamma, \Delta \vdash \Lambda, \Theta}\ \to_l \qquad \frac{A, \Gamma \vdash \Theta, B}{\Gamma \vdash \Theta, A \to B}\ \to_r$$

$$\frac{A[\alpha], \Gamma \vdash \Theta}{\exists x.A[x], \Gamma \vdash \Theta}\ \exists_l \qquad \frac{A[t], \Gamma \vdash \Theta}{\forall x.A[x], \Gamma \vdash \Theta}\ \forall_l \qquad \frac{\Gamma \vdash \Theta, A[\alpha]}{\Gamma \vdash \Theta, \forall x.A[x]}\ \forall_r \qquad \frac{\Gamma \vdash \Theta, A[t]}{\Gamma \vdash \Theta, \exists x.A[x]}\ \exists_r$$

The eigenvariable α should not occur in Γ, Θ or $A[x]$.
The term t should not contain variables bound in $A[t]$.

History: This is a modern presentation of Gentzen's original **LK** calculus [1], using modern notations and rule names.

Remarks: **LK** is complete relative to **NK** (i.e. **NJ** {5} with the axiom of excluded middle) and sound relative to a Hilbert-style calculus **LHK** [2]. Cut is eliminable (*Hauptsatz* [1]), and hence classical predicate logic is consistent. Any *prenex* cut-free proof may be further transformed into a shape with only propositional inferences above and only quantifier and structural inferences below a *midsequent* [2].

---

[1] Gerhard Gentzen. "Untersuchungen über das logische Schließen I". In: *Mathematische Zeitschrift* 39.1 (Dec. 1935), pp. 176–210.

[2] Gerhard Gentzen. "Untersuchungen über das logische Schließen II". In: *Mathematische Zeitschrift* 39.1 (Dec. 1935), pp. 405–431.

Entry 6 by: Martin Riener

Intuitionistic Sequent Calculus LJ (1935)

$$\frac{}{A \vdash A} \qquad \frac{\Gamma \vdash A \quad A, \Delta \vdash \Theta}{\Gamma, \Delta \vdash \Theta}\ cut$$

$$\frac{\Gamma \vdash \Theta}{A, \Gamma \vdash \Theta}\ w_l \qquad \frac{\Gamma \vdash}{\Gamma \vdash A}\ w_r$$

$$\frac{\Gamma, B, A, \Delta \vdash \Theta}{\Gamma, A, B, \Delta \vdash \Theta}\ e_l \qquad \frac{A, A, \Gamma \vdash \Theta}{A, \Gamma \vdash \Theta}\ c_l$$

$$\frac{\Gamma \vdash A}{\neg A, \Gamma \vdash}\ \neg_l \qquad \frac{A, \Gamma \vdash}{\Gamma \vdash \neg A}\ \neg_r$$

$$\frac{A_i, \Gamma \vdash \Theta}{A_1 \wedge A_2, \Gamma \vdash \Theta}\ \wedge_l \qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge_r$$

$$\frac{A, \Gamma \vdash \Theta \quad B, \Gamma \vdash \Theta}{A \vee B, \Gamma \vdash \Theta}\ \vee_l \qquad \frac{\Gamma \vdash A_i}{\Gamma \vdash A_1 \vee A_2}\ \vee_r$$

$$\frac{\Gamma \vdash A \quad B, \Delta \vdash \Theta}{A \to B, \Gamma, \Delta \vdash \Theta}\ \to_l \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \to B}\ \to_r$$

$$\frac{A[\alpha], \Gamma \vdash \Theta}{\exists x.A[x], \Gamma \vdash \Theta}\ \exists_l \qquad \frac{\Gamma \vdash A[t]}{\Gamma \vdash \exists x.A[x]}\ \exists_r$$

$$\frac{A[t], \Gamma \vdash \Theta}{\forall x.A[x], \Gamma \vdash \Theta}\ \forall_l \qquad \frac{\Gamma \vdash A[\alpha]}{\Gamma \vdash \forall x.A[x]}\ \forall_r$$

The eigenvariable α should not occur in Γ, Θ or $A[x]$.
The term t should not contain variables bound in $A[t]$.

Clarifications: Gentzen introduced the sequent calculi **LK** {6} and **LJ** for classical and intuitionistic logics respectively. The rules in both systems have the same shape, but in **LJ** they may have at most one formula in the succedent (right side of $\vdash$). This restriction is equivalent to forbidding the axiom of excluded middle in natural deduction.

Remarks: The cut rule is eliminable (*Hauptsatz* [1]), and hence intuitionistic predicate logic is consistent and its propositional fragment is decidable [2]. **LJ** is complete relative to **NJ** {5} and sound relative to the Hilbert-style calculus **LHJ** [2].

[1] Gerhard Gentzen. "Untersuchungen über das logische Schließen I". In: 39.1 (Dec. 1935), pp. 176–210.

[2] Gerhard Gentzen. "Untersuchungen über das logische Schließen II". In: *Mathematische Zeitschrift* 39.1 (Dec. 1935), pp. 405–431.

Entry 7 by: Giselle Reis

Epsilon Calculus (1923, 1939)

Epsilon calculus is first-order predicate calculus extended by the epsilon-operator and the critical axiom. Terms t and Formulas A, B of epsilon calculus are defined as follows.

$$t ::= a \mid x \mid f(t_0, \ldots, t_{n-1}) \mid \varepsilon_x A, \qquad A, B ::= P(t_0, \ldots, t_{n-1}) \mid \neg A \mid A \wedge B \mid A \vee B \mid A \to B \mid \exists x.A \mid \forall x.A,$$

where a, x, f, and P range over free variables, bound variables, function symbols, and predicate symbols, respectively. Each f, P has an arbitrary arity n. The critical axiom is, for any formula $A(x)$, given as follows.

$$A(t) \to A(\varepsilon_x A(x)).$$

Clarifications: *Epsilon calculus* is an extension of classical first-order predicate calculus [3, 4, 5]. The symbol ε is called the *epsilon-operator*, which constructs a term by quantifying a bound variable in a formula. A formula A with occurrences of a variable x which is not quantified is written as $A(x)$, and $A(t)$ denotes a formula obtained by replacing the corresponding x by a term t in A. The existential and universal quantifiers are definable due to the epsilon-operator.

$$\exists x.A(x) := A(\varepsilon_x A(x)), \qquad \forall x.A(x) := A(\varepsilon_x \neg A(x)).$$

Pure epsilon calculus is elementary calculus extended by the epsilon-operator and the critical axiom.

History: Epsilon calculus is due to Hilbert. He formulated the prototype of epsilon calculus [1] by means of the *tau-operator* and the axiom $A(\tau_x A(x)) \to A(t)$ instead of the epsilon-operator and the critical axiom. The formulation based on the epsilon-operator first appeared in Ackermann's dissertation [2] under the supervision of Hilbert. Hilbert and Bernays gave a comprehensive account of epsilon calculus and its applications [3].

Remarks: *First epsilon theorem* states that if there is a proof in epsilon calculus of an $\exists, \forall, \varepsilon$-free formula, this formula is provable in elementary calculus. *Second epsilon theorem* states that if there is a proof in epsilon calculus of an ε-free formula, this formula is provable in predicate calculus. By means of epsilon calculus Hilbert and Bernays gave the first correct proof of *Herbrand's theorem* [3, 4].

[1] David Hilbert. "Die logischen Grundlagen der Mathematik". In: *Mathematische Annalen* 88 (1923), pp. 151–165.

[2] Wilhelm Ackermann. "Begründung des "tertium non datur" mittels der Hilbertschen Theorie der Widerspruchsfreiheit". In: *Mathematische Annalen* 93 (1924), pp. 1–36.

[3] David Hilbert and Paul Bernays. *Grundlagen der Mathematik*. vol. 2. Springer, Berlin, 1939.

[4] Georg Moser and Richard Zach. "The Epsilon Calculus and Herbrand Complexity". In: *Studia Logica* 82.1 (2006), pp. 133–155. DOI: 10.1007/s11225-006-6610-7.

[5] Jeremy Avigad and Richard Zach. "The Epsilon Calculus". In: *Stanford Encyclopedia of Philosophy* (Nov. 2013). URL: https://plato.stanford.edu/entries/epsilon-calculus/.

Entry 8 by: Kenji Miyamoto, Georg Moser

Kleene's Classical G3 System (1952)

$$\frac{}{A, \Gamma \vdash \Theta, A}$$

$$\frac{A \to B, \Gamma \vdash \Theta, A \quad B, A \to B, \Gamma \vdash \Theta}{A \to B, \Gamma \vdash \Theta} \to\vdash \qquad \frac{A, \Gamma \vdash \Theta, A \to B, B}{\Gamma \vdash \Theta, A \to B} \vdash\to$$

$$\frac{A, A \vee B, \Gamma \vdash \Theta \quad B, A \vee B, \Gamma \vdash \Theta}{A \vee B, \Gamma \vdash \Theta} \vee\vdash \qquad \frac{\Gamma \vdash \Theta, A \vee B, A}{\Gamma \vdash \Theta, A \vee B} \vdash \vee_1 \quad \frac{\Gamma \vdash \Theta, A \vee B, B}{\Gamma \vdash \Theta, A \vee B} \vdash \vee_2$$

$$\frac{A, A \wedge B, \Gamma \vdash \Theta}{A \wedge B, \Gamma \vdash \Theta} \wedge\vdash_1 \quad \frac{B, A \wedge B, \Gamma \vdash \Theta}{A \wedge B, \Gamma \vdash \Theta} \wedge\vdash_2 \qquad \frac{\Gamma \vdash \Theta, A \wedge B, A \quad \Gamma \vdash \Theta, A \wedge B, B}{\Gamma \vdash \Theta, A \wedge B} \vdash \wedge$$

$$\frac{\neg A, \Gamma \vdash \Theta, A}{\neg A, \Gamma \vdash \Theta} \neg\vdash \qquad \frac{A, \Gamma \vdash \Theta, \neg A}{\Gamma \vdash \Theta, \neg A} \vdash\neg$$

$$\frac{A(t), \forall x A(x), \Gamma \vdash \Theta}{\forall x A(x), \Gamma \vdash \Theta} \forall\vdash \qquad \frac{\Gamma \vdash \Theta, \forall x A(x), A(b)}{\Gamma \vdash \Theta, \forall x A(x)} \vdash\forall$$

$$\frac{A(b), \exists x A(x), \Gamma \vdash \Theta}{\exists x A(x), \Gamma \vdash \Theta} \exists\vdash \qquad \frac{\Gamma \vdash \Theta, \exists x A(x), A(t)}{\Gamma \vdash \Theta, \exists x A(x)} \vdash\exists$$

The term t is free for x in $A(x)$.
The variable b is free for x in $A(x)$ and (unless b is x) does not occur in $\Gamma, \Theta, A(x)$.

Clarifications: A, B are formulae; Γ, Θ are finite (possibly empty) sequences of formulae; x is a variable; $A(x)$ is a formula. In applications of the rules every sequent $\Gamma \vdash \Theta$ can be replaced with a *cognate* one, i.e., a sequent $\Gamma' \vdash \Theta'$ such that the sets of formulae occurring in Γ and Γ' resp. Θ and Θ' are the same.

History: Kleene's systems, introduced in his 1952 monograph, were the staple of generations of logicians, who learned about sequent calculus from his textbooks [1] and [2].

Remarks: Based on Gentzen's sequent calculus **LK** {6} (called classical **G1** in [1]). Seems to be the first system (with {10}) in which admissibility of contraction is obtained by copying the principal formulae into the premisses (accordingly, this is sometimes called *Kleene's Method*). Used together with its single-conclusion version for intuitionistic logic {10} to uniformly obtain decidability of propositional classical and intuitionistic logics via backwards proof search in [1].

[1] Stephen Cole Kleene. *Introduction to Metamathematics*. Ishi Press reprint (2009). Amsterdam: North-Holland, 1952.

[2] Stephen Cole Kleene. *Mathematical Logic*. Dover reprint (2001). New York: John Wiley, 1967.

Entry 9 by: Björn Lellmann, Valeria de Paiva

Kleene's Intuitionistic G3 System (1952)

$$\frac{}{A,\Gamma \vdash A}$$

$$\frac{A \to B,\Gamma \vdash A \quad B,A \to B,\Gamma \vdash \Theta}{A \to B,\Gamma \vdash \Theta} \to\vdash \qquad \frac{A,\Gamma \vdash B}{\Gamma \vdash A \to B} \vdash\to$$

$$\frac{A,A \vee B,\Gamma \vdash \Theta \quad B,A \vee B,\Gamma \vdash \Theta}{A \vee B,\Gamma \vdash \Theta} \vee\vdash \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vdash \vee_1 \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \vdash \vee_2$$

$$\frac{A,A \wedge B,\Gamma \vdash \Theta}{A \wedge B,\Gamma \vdash \Theta} \wedge\vdash_1 \quad \frac{B,A \wedge B,\Gamma \vdash \Theta}{A \wedge B,\Gamma \vdash \Theta} \wedge\vdash_2 \qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \vdash\wedge$$

$$\frac{\neg A,\Gamma \vdash A}{\neg A,\Gamma \vdash \Theta} \neg\vdash \qquad \frac{A,\Gamma \vdash \neg A}{\Gamma \vdash \neg A} \vdash\neg$$

$$\frac{A(t),\forall x A(x),\Gamma \vdash \Theta}{\forall x A(x),\Gamma \vdash \Theta} \forall\vdash \qquad \frac{\Gamma \vdash A(b)}{\Gamma \vdash \forall x A(x)} \vdash\forall$$

$$\frac{A(b),\exists x A(x),\Gamma \vdash \Theta}{\exists x A(x),\Gamma \vdash \Theta} \exists\vdash \qquad \frac{\Gamma \vdash A(t)}{\Gamma \vdash \exists x A(x)} \vdash\exists$$

The term t is free for x in $A(x)$.
The variable b is free for x in $A(x)$ and (unless b is x) does not occur in $\Gamma,\Theta,A(x)$.

Clarifications: A, B are formulae; Γ and Θ are a finite (possibly empty) sequences of formulae with Θ containing at most one formula; x is a variable; $A(x)$ is a formula. In applications of the rules every sequent $\Gamma \vdash \Theta$ can be replaced with a *cognate* one, i.e., a sequent $\Gamma' \vdash \Theta'$ such that the sets of formulae occurring in Γ and Γ' resp. Θ and Θ' are the same (respecting the restriction to at most one formula on the right hand side).

History: Kleene's systems, introduced in his 1952 monograph, were the staple of generations of logicians, who learned about sequent calculus from his textbooks [1] and [2].

Remarks: Based on Gentzen's sequent calculus **LJ** {7} (corresponding to intuitionistic **G1** in [1]). Seems to be the first system (with {9}) in which admissibility of contraction is obtained by copying the principal formulae into the premisses (accordingly, this is sometimes called *Kleene's Method*). Used together with its multi-conclusion version for classical logic {9} to uniformly obtain decidability of propositional classical and intuitionistic logics via backwards proof search in [1].

---

[1] Stephen Cole Kleene. *Introduction to Metamathematics*. Ishi Press reprint (2009). North-Holland, 1952.

[2] Stephen Cole Kleene. *Mathematical Logic*. Dover reprint (2001). John Wiley, 1967.

Entry 10 by: Björn Lellmann, Valeria de Paiva

Multi-Conclusion Sequent Calculus LJ′ (1954)

$$\frac{}{A \vdash A} \qquad \frac{\Gamma \vdash \Theta, A \quad A, \Delta \vdash \Lambda}{\Gamma, \Delta \vdash \Theta, \Lambda}\ cut$$

$$\frac{A_i, \Gamma \vdash \Theta}{A_1 \wedge A_2, \Gamma \vdash \Theta}\ \wedge_l \qquad \frac{\Gamma \vdash \Theta, A \quad \Gamma \vdash \Theta, B}{\Gamma \vdash \Theta, A \wedge B}\ \wedge_r$$

$$\frac{A, \Gamma \vdash \Theta \quad B, \Gamma \vdash \Theta}{A \vee B, \Gamma \vdash \Theta}\ \vee_l \qquad \frac{\Gamma \vdash \Theta, A_i}{\Gamma \vdash \Theta, A_1 \vee A_2}\ \vee_r$$

$$\frac{\Gamma \vdash \Theta, A \quad B, \Delta \vdash \Lambda}{A \to B, \Gamma, \Delta \vdash \Theta, \Lambda}\ \to_l \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \to B}\ \to_r$$

$$\frac{A\alpha, \Gamma \vdash \Theta}{\exists x.Ax, \Gamma \vdash \Theta}\ \exists_l \quad \frac{\Gamma \vdash \Theta, At}{\Gamma \vdash \Theta, \exists x.Ax}\ \exists_r \quad \frac{At, \Gamma \vdash \Theta}{\forall x.Ax, \Gamma \vdash \Theta}\ \forall_l \quad \frac{\Gamma \vdash A\alpha}{\Gamma \vdash \forall x.Ax}\ \forall_r$$

$$\frac{\Gamma \vdash \Theta, A}{\neg A, \Gamma \vdash \Theta}\ \neg_l \qquad \frac{A, \Gamma \vdash}{\Gamma \vdash \neg A}\ \neg_r \qquad \frac{\Gamma, B, A, \Delta \vdash \Theta}{\Gamma, A, B, \Delta \vdash \Theta}\ e_l \quad \frac{\Gamma \vdash \Theta, B, A, \Lambda}{\Gamma \vdash \Theta, A, B, \Lambda}\ e_r$$

$$\frac{A, A, \Gamma \vdash \Theta}{A, \Gamma \vdash \Theta}\ c_l \qquad \frac{\Gamma \vdash \Theta, A, A}{\Gamma \vdash \Theta, A}\ c_r \qquad \frac{\Gamma \vdash \Theta}{A, \Gamma \vdash \Theta}\ w_l \qquad \frac{\Gamma \vdash}{\Gamma \vdash A}\ w_r$$

The eigenvariable α should not occur in Γ, Θ or $A[x]$.
The term t should not contain variables bound in $A[t]$.

Clarifications: While **LJ** {7} is defined by restricting **LK** {6} to single conclusion, in **LJ′** only the rules $\neg_r$, $\to_r$ and $\forall_r$ have this restriction.

History: **LJ′** was proposed in [1] and used to prove the completeness of **LJ** {7} in [3]. It also appears in [4] (as GHPC) and [2] (as L').

Remarks: **LJ′** is equivalent to **LJ**, and this is established by translating sequents of the form $\Gamma \vdash A_1, ..., A_n$ into sequents of the form $\Gamma \vdash A_1 \vee ... \vee A_n$. Cut can be eliminated by using a combination of the rewriting rules for cut-elimination in **LJ** and **LK** and permutation of inferences, as shown by Schellinx [5] and Reis [6].

---

[1] Shôji Maehara. "Eine Darstellung der intuitionistischen Logik in der klassischen". In: *Nagoya Math. J.* 7 (1954), pp. 45–64.

[2] Michael Dummett. *Elements of Intuitionism*. Oxford: Clarendon Press, 1977.

[3] Gaisi Takeuti. *Proof Theory*. 2nd Edition. North Holland, 1987.

[4] A. G. Dragalin. *Mathematical Intuitionism: Introduction to Proof Theory*. American Mathematical Society, 1988.

[5] Harold Schellinx. "Some Syntactical Observations on Linear Logic". In: *J. Log. Comput.* 1.4 (1991), pp. 537–559.

[6] Giselle Reis. "Cut-elimination by resolution in intuitionistic logic". PhD thesis. Vienna University of Technology, July 2014.

Entry 11 by: Giselle Reis, Valeria de Paiva

Lambek Calculus (1958)

$$\frac{}{A \vdash A}\ ax \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2, A, \Gamma_3 \vdash C}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash C}\ cut$$

$$\frac{}{\cdot \vdash I}\ I_r \qquad \frac{\Gamma_1, \Gamma_2 \vdash A}{\Gamma_1, I, \Gamma_2 \vdash A}\ I_l \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2 \vdash B}{\Gamma_1, \Gamma_2 \vdash A \otimes B}\ \otimes_r \qquad \frac{\Gamma_1, A, B, \Gamma_2 \vdash C}{\Gamma_1, A \otimes B, \Gamma_2 \vdash C}\ \otimes_l$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightharpoonup B}\ \rightharpoonup_r \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2, B, \Gamma_3 \vdash C}{\Gamma_1, A \rightharpoonup B, \Gamma_2, \Gamma_3 \vdash C}\ \rightharpoonup_l \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash B \leftharpoonup A}\ \leftharpoonup_r \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2, B, \Gamma_3 \vdash C}{\Gamma_1, B \leftharpoonup A, \Gamma_2, \Gamma_3 \vdash C}\ \leftharpoonup_l$$

Clarifications: The Lambek Calculus described here was introduced by Joachim Lambek to study sentence structure in 1958 [1]. Actually the calculus Lambek first introduced, despite being motivated by algebraic considerations as we are told in [3], had no constant corresponding to the unit of the tensor product I. The Lambek calculus can be seen as the logic one obtains from Gentzen's Intuitionistic Propositional Logic (LJ) {7} if we remove the structural rules of contraction, weakening and commutation. Lambek also introduced another calculus [2] where even the associativity of the tensor is not valid.

History: The system now known as the basic Lambek Calculus was introduced in 1958 by Joachim Lambek as the "Syntactic Calculus" [1]. Lambek's motivation was to "to obtain an effective rule (or algorithm) for distinguishing sentences from non-sentences, which works not only for the formal languages of interest to the mathematical logician, but also for natural languages [...]", as explained by Moortgat in [4]. After a long period of ostracism, around the middle 1980s the Syntactic Calculus, now called the Lambek Calculus was taken up by logicians interested in Computational Linguistics, especially van Benthem, Buszkowski and Moortgat. They realized that a computational semantics for categorical derivations along the lines of the Curry-Howard proofs-as-programs interpretation would provide us with a "parsing-as-deduction" paradigm and a powerful tool to study "logical" derivational semantics. Around the same time, the introduction of Linear Logic {25}, by Jean-Yves Girard also gave a new impulse to the work in Categorical Grammars. This was because of Linear Logic's insight that even if you had a very weak proof system, you could introduce structural rules in a controlled fashion and hence obtain more expressive systems, by the use of the so called modalities. Since no expressivity is lost in this process, this opened the way for various types of experiments, trying to make sure that the logical system could cope with more phenomena from the language, see discussion of examples in [4].

[1] Joachim Lambek. "The mathematics of sentence structure". In: *American Mathematical Monthly* (1958), pp. 154–170.

[2] J. Lambek. "Structure of Language and its Mathematical Aspects, Proceedings of the Symposia in Applied Mathematics (Volume XII)". In: ed. by R. Jacobson. American Mathematical Society, 1961. Chap. On the Calculus of Syntactic Types, pp. 166–178.

[3] J. Lambek. "Categorial Grammars and Natural Language Structures". In: ed. by Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler. Springer Netherlands, 1988. Chap. Categorial and Categorical Grammars, pp. 297–317. ISBN: 978-94-015-6878-4. DOI: 10.1007/978-94-015-6878-4_11.

[4] Michael Moortgat. "Typelogical Grammar". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2014. 2014.

Entry 12 by: Harley Eades III, Valeria de Paiva

Resolution (1965)

$$\frac{D \vee B_1 \vee \cdots \vee B_m \quad C \vee \neg A_1 \vee \cdots \vee \neg A_n}{(D \vee C)\sigma}\ \textit{Resolution}$$

C, D are (possibly empty) clauses, A_i, B_j are atoms.
$A_1, \ldots, A_n, B_1, \ldots, B_m$ are unifiable with most general unifier σ.

Clarifications: Resolution is a refutational saturation calculus for first-order clauses (disjunctions of possibly negated atoms). It works on a set N of clauses that is saturated by successively computing *Resolution* inferences with premises in N and adding the conclusion of the inference to N, until the empty clause (i. e., false) is derived.

History: The ground version of the *Resolution* rule appeared already as "Rule for Eliminating Atomic Formulas" in [1]. To refute a set of non-ground clauses, the rule was combined with a naïve enumeration of ground instances. Robinson's fundamental achievement [3] was to extend the inference rule to non-ground clauses in such a way that the computation of useful instances became a by-product of the rule application. It was later detected that resolution can also be described as the dual form of a special case of Maslov's *inverse method* [2, 4].

Many refinements of resolution were developed in the sequel, aiming on the one hand at reducing the number of possible inferences (e. g., using atom orderings {15}, selection functions, set-of-support strategies) and on the other hand at integrating particular axioms into the calculus (e. g., the equality axioms, yielding paramodulation {16}). Note that the factoring step (i. e., unification of literals within the same clause) that is built into Robinson's original *Resolution* rule is usually given as a separate inference rule in later publications, e. g., {16}.

Remarks: The resolution calculus is refutationally complete for sets of first-order clauses.

---

[1] Martin Davis and Hilary Putnam. "A Computing Procedure for Quantification Theory". In: *Journal of the ACM* 7 (1960), pp. 201–215.

[2] S. Ju. Maslov. "An inverse method of establishing deducibility in classical predicate calculus". In: *Dokl. Akad. Nauk. SSSR* 159 (1964), pp. 17–20.

[3] John Alan Robinson. "A Machine-Oriented Logic Based on the Resolution Principle". In: *Journal of the ACM* 12.1 (1965), pp. 23–41.

[4] S. Ju. Maslov. "Proof-search Strategies for Methods of the Resolution Type". In: *Machine Intelligence 6*. Ed. by Bernard Meltzer and Donald Michie. Edinburgh University Press, 1971. Chap. 6, pp. 77–90.

Entry 13 by: Uwe Waldmann

First-Order Unification (1965)

$$\frac{\{\langle u, u\rangle\} \cup S}{S}\ delete \qquad \frac{\{\langle f(v_1,\ldots,v_n), f(u_1,\ldots,u_n)\rangle\} \cup S}{\{\langle v_1,u_1\rangle,\ldots,\langle v_n,u_n\rangle\} \cup S}\ decomp \qquad \frac{\{\langle x, v\rangle\} \cup S}{\{\langle\langle x, v\rangle\rangle\} \cup \sigma(S)}\ varelim$$

Where x does not occur in v and $\sigma = [v/x]$.

Clarifications: x is a variable. $v_1,\ldots,v_n,u_1,\ldots,u_n$ are terms. $\langle v,u\rangle$ and $\langle\langle v,u\rangle\rangle$ are unsolved and solved, respectively, pairs of first-order terms. S is a set of such pairs and σ is a substitution. The set S is considered solved if it contains only solved pairs.

History: The unification principle was first described by Herbrand in his thesis [1] but was overlooked until rediscovered independently by Prawitz [2] and Guard [3] (where it is called matching, not to be confused with the modern notion of matching - the unification of a term with a ground term). These findings helped pave the way for Robinson's seminal work on Resolution [4] (see {13}). The above set of rules is taken from Snyder and Gallier [5].

Remarks: The application of the above rules always terminates on a given set of pairs of terms and if, in addition, the set is unifiable, then it terminates in a set S' containing only solved pairs. The set S' contains the substitution components [4] of a most general unifier of S. The choice of which rule to apply is a "don't-care" non-determinism, which means that the resulting substitutions, if they exist, are identical up to renaming of free variables.

---

[1] Jacques Herbrand. "Recherches sur la théorie de la démonstration". In: (1930).

[2] Dag Prawitz. "An Improved Proof Procedure". In: *Theoria* 26.2 (1960), pp. 102–139.

[3] James R Guard. *Automated logic for semi-automated mathematics*. Tech. rep. DTIC Document, 1964.

[4] John Alan Robinson. "A Machine-Oriented Logic Based on the Resolution Principle". In: 12.1 (1965), pp. 23–41.

[5] Wayne Snyder and Jean Gallier. "Higher-order unification revisited: Complete sets of transformations". In: *Journal of Symbolic Computation* 8.1 - 2 (1989), pp. 101–140. ISSN: 0747-7171.

Entry 14 by: Tomer Libal

Ordered Resolution (1969)

$$\frac{D \vee B \quad C \vee \neg A}{(D \vee C)\sigma} \textit{ Resolution}$$

$$\frac{C \vee L_1 \vee \cdots \vee L_n}{(C \vee L_1)\sigma} \textit{ Factoring}$$

C, D are (possibly empty) clauses, $L_1, \ldots, L_n$ are literals, A, B are atoms, A and B, or $L_1, \ldots, L_n$, respectively, are unifiable with most general unifier σ.
The literals $\neg A$, B, and L_1 are maximal in the respective premises.

Clarifications: Ordered resolution is a refutational saturation calculus for first-order clauses (disjunctions of possibly negated atoms). It works on a set N of clauses that is saturated by successively computing inferences with premises in N and adding the conclusion of the inference to N, until the empty clause (i. e., false) is derived.

History: The idea to use a syntactic ordering on literals to restrict the number of possible inferences was developed independently by Maslov [1, 2, 4] for the *inverse method* (resolution can be seen as the dual form of a special case of the inverse method) and by Kowalski and Hayes [3] for resolution itself (the requirements for the ordering differ slightly).

Remarks: The ordered resolution calculus is refutationally complete for sets of first-order clauses.

---

[1] S. Ju. Maslov. "An inverse method of establishing deducibility in classical predicate calculus". In: 159 (1964), pp. 17–20.

[2] S. Ju. Maslov. "The inverse method for establishing deducibility for logical calculi". In: *Trudy Mat. Inst. Steklov* 98 (1968), pp. 26–87.

[3] R[obert] Kowalski and P[atrick] J. Hayes. "Semantic Trees in Automatic Theorem Proving". In: *Machine Intelligence 4*. Ed. by Bernard Meltzer and Donald Michie. Edinburgh University Press, 1969, pp. 87–101.

[4] S. Ju. Maslov. "Proof-search Strategies for Methods of the Resolution Type". In: *Machine Intelligence 6*. Ed. by Bernard Meltzer and Donald Michie. Edinburgh University Press, 1971. Chap. 6, pp. 77–90.

Entry 15 by: Uwe Waldmann

Paramodulation (1969)

$$\frac{D \vee u \approx u' \quad C \vee L[v]}{(D \vee C \vee L[u'])\sigma} \textit{ Paramodulation}$$

$$\frac{D \vee B \quad C \vee \neg A}{(D \vee C)\sigma} \textit{ Resolution}$$

$$\frac{C \vee L_1 \vee \cdots \vee L_n}{(C \vee L_1)\sigma} \textit{ Factoring}$$

$$\frac{}{x \approx x} \textit{ Reflexivity}$$

C, D are (possibly empty) equational clauses, $L, L_1, \ldots, L_n$ are literals, A, B are atoms, u, u', v are terms; u and v, A and B, or $L_1, \ldots, L_n$, respectively, are unifiable with most general unifier σ.

Clarifications: Paramodulation is a refutational saturation calculus for first-order clauses (disjunctions of possibly negated atoms) with equality (denoted by $\approx$). It works on a set N of clauses that is saturated by successively computing inferences with premises in N and adding the conclusion of the inference to N, until the empty clause (i. e., false) is derived.

History: Handling the equality axioms in the resolution calculus {13} is impractical due to the huge search space generated in particular by the transitivity axiom. The paramodulation calculus developed by Robinson and Wos [1] extends resolution by specific inference rules that render explicit inferences with the equality axioms unnecessary. The original completeness proof also assumed the presence of so-called functional-reflexive axioms of the form $f(x_1, \ldots, x_n) \approx f(x_1, \ldots, x_n)$; this was later shown to be superfluous by Brand [2]. Many refinements were developed in the sequel, aiming in particular at reducing the number of possible inferences, see {31}.

Remarks: The paramodulation calculus is refutationally complete for first-order logic with equality.

---

[1] G[eorge] Robinson and L[arry] Wos. "Paramodulation and Theorem-proving in First-Order Theories with Equality". In: *Machine Intelligence 4*. Ed. by Bernard Meltzer and Donald Michie. Edinburgh University Press, 1969. Chap. 8, pp. 135–150.

[2] D. Brand. "Proving Theorems with the Modification Method". In: *SIAM Journal on Computing* 4.4 (1975), pp. 412–430.

Entry 16 by: Uwe Waldmann

(Unfailing) Completion (1970/1986)

Standard Completion:

$$\frac{E \cup \{s \mathrel{\dot{\approx}} t\},\ R}{E,\ R \cup \{s \to t\}}\ \textit{Orient} \quad \text{if } s \succ t$$

$$\frac{E,\ R}{E \cup \{s \approx t\},\ R}\ \textit{Deduce} \quad \text{if } \langle s,t\rangle \in \mathrm{CP}(R)$$

$$\frac{E \cup \{s \approx s\},\ R}{E,\ R}\ \textit{Delete}$$

$$\frac{E \cup \{s \mathrel{\dot{\approx}} t\},\ R}{E \cup \{u \mathrel{\dot{\approx}} t\},\ R}\ \textit{Simplify-Equation} \quad \text{if } s \to_R u$$

$$\frac{E,\ R \cup \{s \to t\}}{E \cup \{u \approx t\},\ R}\ \textit{Left-Simplify-Rule} \quad \text{if } s \to_R u \text{ using } l \to r \in R \text{ such that } s \sqsupset l$$

$$\frac{E,\ R \cup \{s \to t\}}{E,\ R \cup \{s \to u\}}\ \textit{Right-Simplify-Rule} \quad \text{if } t \to_R u$$

plus, for Unfailing Completion:

$$\frac{E,\ R}{E \cup \{s \approx t\},\ R}\ \textit{UC-Deduce} \quad \text{if } \langle s,t\rangle \in \mathrm{CP}(E \cup R)$$

E is a set of equations, R is a set of rewrite rules, s,t,u,l,r are terms, $s \mathrel{\dot{\approx}} t$ represents $s \approx t$ or $t \approx s$, CP(...) denotes the set of (ordered) critical pairs of a set of (equations and) rules, $\succ$ is a reduction ordering that is total on ground terms.

Clarifications: Standard completion tries to convert a set of equations into an equivalent terminating and confluent set of rewrite rules; it may fail, however, for certain inputs E and $\succ$. Adding the *UC-Deduce* rule turns standard completion into a refutationally complete calculus for equational theories.

History: Standard completion was developed by Knuth and Bendix [1]; the presentation as an inference system given here and the extension to unfailing completion are due to Bachmair, Dershowitz, and Hsiang [3, 4]. An extension of completion to completion modulo associativity and/or commutativity was presented in [2].

Remarks: To prove that an equation $s \approx t$ is entailed by E, unfailing completion is applied to $E \cup \{eq(x,x) \approx true,\ eq(\hat{s},\hat{t}) \approx false\}$, where $\hat{s}$ and $\hat{t}$ are skolemized versions of s and t. Unfailing completion derives $true \approx false$ if and only if $E \models s \approx t$.

---

[1] Donald E. Knuth and Peter B. Bendix. "Simple Word Problems in Universal Algebras". In: *Computational Problems in Abstract Algebra*. Ed. by J. Leech. Oxford, United Kingdom: Pergamon Press, 1970, pp. 263–297.

[2] Gerald E. Peterson and Mark E. Stickel. "Complete Sets of Reductions for Some Equational Theories". In: *Journal of the ACM* 28.2 (1981), pp. 233–264.

[3] Leo Bachmair, Nachum Dershowitz, and Jieh Hsiang. "Orderings for Equational Proofs". In: *[First Annual] Symposium on Logic in Computer Science*. Cambridge, Massachusetts, USA: IEEE Computer Society Press, 1986, pp. 346–357.

[4] Leo Bachmair. *Canonical Equational Proofs*. Boston, MA, USA: Birkhäuser, 1991.

Entry 17 by: Uwe Waldmann

Second Order λ-Calculus (System F) (1971)

$$\frac{(x:T)\in E}{\Gamma;E\vdash x:T}\ \text{assumption}$$

$$\frac{\Gamma;E,(x:T)\vdash e:S}{\Gamma;E\vdash(\lambda x:T.e):(T\to S)}\to I \qquad \frac{\Gamma;E\vdash f:(T\to S)\quad \Gamma;E\vdash e:T}{\Gamma;E\vdash(fe):S}\to E$$

$$\frac{\Gamma X;E\vdash e:T}{\Gamma;E\vdash(\Lambda X:Tp.e):(\forall X:Tp.T)}\ \forall I^{*} \qquad \frac{\Gamma;E\vdash f:(\forall X:Tp.T)\quad \Gamma\vdash S:Tp}{\Gamma;E\vdash fS:[S/X]T}\ \forall E$$

* X must be not free in the type of any free term variable in E.

Clarifications: The presentation from [4] with minor corrections is used. $X,Y,Z...$ are type-variables and $x,y,...$ term variables. Expressions are type ($T := X|(T\to S)|(\forall X:Tp.T)$) or terms ($e := x|(ee)|(eT)|(\lambda x:T.e)|(\Lambda X:Tp.e)$). $\forall,\Lambda$ and λ are variable binders. All expressions are considered up to renaming of bound variables (α-conversion). An unbound variable is free. $FV(R)$ is the set of free variables for any (type or term) expression; $[e/x]$, $[S/X]$ mean capture-avoiding substitution in term- and type-expressions respectively (defined by induction). A context is a finite set Γ of type variables; ΓX stands for $\Gamma\cup X$. A type T is legal in Γ iff $FV(T)\subseteq FV(\Gamma)$. A type assignment in Γ is a finite list $E=(x_1:T_1),...,(x_n:T_n)$ where any T_i is legal in Γ. The typing relation $\Gamma;E\vdash e:T$, where E is a type assignment legal in Γ, e is a term and T is a type, is defined by the rules above. The *conversion relation* between well-typed terms is very important. It is defined by the following axioms: (β) $(\lambda x:T.f)e=[e/x]f$; (β_2) $(\Lambda X:Tp.e)S=[S/X]e$; (η) $\lambda x:T.(ex)=e$ if $x\notin FV(e)$; (η_2) $\Lambda X:Tp.(eX)=e$ if $X\notin FV(e)$, and by usual rules that turn "=" into congruence. The system $\mathbf{F}_c$ is obtained if one more equality axiom is added: (**C**) $eT=eT'$ for $\Gamma;E\vdash e:\forall X.S$ and $X\notin FV(S)$.

History: It was introduced in [1, 3] and included in the λ-cube [6]. It is important for functional programming and inspired works on higher order type systems and many extensions (e.g. $\mathbf{F}_c$ [7], **F** with subtyping [5, 8]).

Remarks: A strong normalization theorem for **F** was proved by Girard [2]. It implies a normalization theorem and consistency for second order arithmetic PA_2. For $\mathbf{F}_c$, a *genericity theorem* holds [7].

[1] J.-Y. Girard. "Une extension de l'interprétation fonctionelle de Gödel à l'analyse et son application à l'élimination des coupures dans et la théorie des types". In: *Proc. 2nd Scandinavian Logic Symposium.* North-Holland (1971).

[2] J.-Y. Girard. "Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur". PhD thesis. Université Paris VII, 1972.

[3] J.C. Reynolds. "Towards a Theory of Type Structure". In: *LNCS* 19 (1974).

[4] Andrea Asperti and Giuseppe Longo. *Categories, Types and Structures*. Cambridge, Mass., London, England: The MIT Press, 1991.

[5] L. Cardelli, S. Martini, J.C. Mitchell, and A. Scedrov. "An Extension of System F with Subtyping". In: *Lecture Notes in Computer Science* 526 (1991).

[6] H.P.Barendregt. "Introduction to generalized type systems". In: *J. of Functional Programming* 2 (1991).

[7] G. Longo, K. Milsted, and S. Soloviev. "The Genericity Theorem and the Notion of Parametricity in the Polymorphic λ-calculus". In: *Theoretical Computer Science* 121 (1993).

[8] G. Longo, K. Milsted, and S. Soloviev. "Coherence and Transitivity of Subtyping as Entailment". In: *Journal of Logic and Computation* 10 (2000).

Entry 18 by: Giuseppe Longo, Kathleen Milsted, Sergei Soloviev

Higher-Order Pre-Unification (1975)

$$\frac{\{\langle u, u\rangle\} \cup S}{S}\ \mathit{delete} \qquad \frac{\{\langle \lambda\overline{x_k}.z(\overline{x_k}), \lambda\overline{x_k}.v\rangle\} \cup S}{\{\langle\langle z, \lambda\overline{x_k}.v\rangle\rangle\} \cup \sigma(S)\downarrow_\beta}\ \mathit{varelim}$$

$$\frac{\{\langle \lambda\overline{x_k}.a(\overline{v_n}), \lambda\overline{x_k}.a(\overline{u_n}\rangle\} \cup S}{\{\langle \lambda\overline{x_k}.v_1, \lambda\overline{x_k}.u_1\rangle, \ldots, \langle \lambda\overline{x_k}.v_n, \lambda\overline{x_k}.u_n\rangle\} \cup S}\ \mathit{decomp}$$

$$\frac{\{\langle \lambda\overline{x_k}.y(\overline{u_n}), \lambda\overline{x_k}.b(\overline{v_m})\rangle\} \cup S}{\{\langle y\uparrow_\eta, t\uparrow_\eta\rangle, \langle \lambda\overline{x_k}.y(\overline{u_n}), \lambda\overline{x_k}.b(\overline{v_m})\rangle\} \cup S}\ \mathit{imitate} \qquad \frac{\{\langle \lambda\overline{x_k}.y(\overline{u_n}), \lambda\overline{x_k}.a(\overline{v_m})\rangle\} \cup S}{\{\langle y\uparrow_\eta, s\uparrow_\eta\rangle, \langle \lambda\overline{x_k}.y(\overline{u_n}), \lambda\overline{x_k}.a(\overline{v_m})\rangle\} \cup S}\ \mathit{project}$$

Where $a \in \Sigma$ or $a \in \overline{x_k}$; $b \in \Sigma$; z does not occur in v; $\sigma = [\lambda\overline{x_k}.v/x]$; $t = \lambda\overline{x_n}.b(\overline{y_m(\overline{x_n})})$; $s = \lambda\overline{x_n}.x_i(\overline{y_l(\overline{x_n})})$ for $0 < i \leq n$ and $l = \mathtt{ty}(x_i)$.

Clarifications: Σ is the term signature. $\overline{o_p} = o_1, \ldots, o_p$. $z, x, \overline{x_k}$ and $y, \overline{y_m}$ are variables. $v, \overline{v_n}$ and $u, \overline{u_n}$ are terms. $\langle v, u\rangle$ and $\langle\langle v, u\rangle\rangle$ are unsolved and solved, respectively, pairs of λ-terms. S is a set of such pairs and σ is a substitution. $\downarrow_\beta$ denotes β-normalization and $\uparrow_\eta$ denotes η-expansion. $\mathtt{ty}(a) = n$ for a symbol a of type $\beta_1 \to \ldots \to \beta_n \to \gamma$. The set S must originaly contain terms in β-normalized and η-expanded form.

History: In contrast to the first-order case, the question whether higher-order terms are unifiable is undecidable already in the second-order case [4]. The first complete procedure for higher-order unification was given by Jensen and Pietrzykowski [3]. The use of pre-unifiers, introduced by Huet, enabled the search to be less redundant and more efficient.

Remarks: Huet [2] introduced the procedure without assuming the axiom of functional extensionality and showed that assuming this axiom makes the procedure non-redundant. The above set of rules assumes extensionality. The set S is considered pre-solved if it contains only solved or "flex-flex" pairs where a "flex" term is a term whose head is a free variable. The solved pairs in S' are the substitution components [1] of a pre-unifier of S which can always be extended into a unifier. The application of $\mathit{imitate}$ and $\mathit{project}$ is a "don't-know" non-determinism, while the choice of delete, $\mathit{varelim}$ or decomp is a "don't-care" non-determinism. Nevertheless, by choosing an appropriate strategy, the application of the above rules always terminates on a unifiable set of pairs of terms and enumerates (with extensionality) a complete and minimal set of pre-unifiers.

[1] John Alan Robinson. "A Machine-Oriented Logic Based on the Resolution Principle". In: 12.1 (1965), pp. 23–41.

[2] Gérard Huet. "A Unification Algorithm for Typed λ-Calculus". In: *Theoretical Computer Science* 1 (1975), pp. 27–57.

[3] D. C. Jensen and Tomasz Pietrzykowski. "Mechanizing ω-Order Type Theory Through Unification". In: *Theor. Comput. Sci.* 3.2 (1976), pp. 123–171.

[4] Warren Goldfarb. "The Undecidability of the Second-Order Unification Problem". In: *Theoretical Computer Science* 13 (1981), pp. 225–230.

Entry 19 by: Tomer Libal

Resolution for Modal Logic K (RK) (1982)

Rules for computing resolvents

$$\text{(A1)}\ \frac{}{\Sigma(p,\neg p)\longrightarrow\bot}\qquad \text{(A2)}\ \frac{}{\Sigma(\bot,A)\longrightarrow\bot}\qquad (\Gamma\Box)\ \frac{\Gamma(A)\longrightarrow B}{\Gamma(\Box A)\longrightarrow\Box B}$$

$$(\Gamma\Diamond 1)\ \frac{\Sigma(A,B)\longrightarrow C}{\Gamma(\Diamond(A\wedge B\wedge E))\longrightarrow\Diamond(A\wedge B\wedge C\wedge E)}\qquad (\Gamma\Diamond 2)\ \frac{\Gamma(A)\longrightarrow B}{\Gamma(\Diamond(A\wedge E))\longrightarrow\Diamond(A\wedge B\wedge E)}$$

$$(\Gamma\vee)\ \frac{\Gamma(A)\longrightarrow B}{\Gamma(A\vee C)\longrightarrow B\vee C}\qquad (\Sigma\vee)\ \frac{\Sigma(A,B)\longrightarrow C}{\Sigma(A\vee D_1,B\vee D_2)\longrightarrow C\vee D_1\vee D_2}$$

$$(\Sigma\Box\Diamond)\ \frac{\Sigma(A,B)\longrightarrow C}{\Sigma(\Box A,\Diamond(B\wedge E))\longrightarrow\Diamond(B\wedge C\wedge E)}\qquad (\Sigma\Box\Box)\ \frac{\Sigma(A,B)\longrightarrow C}{\Sigma(\Box A,\Box B)\longrightarrow\Box C}$$

Simplification rules

$$(S_1)\ \ \Diamond\bot\approx\bot\qquad (S_3)\ \ \bot\wedge E\approx\bot\qquad (S_2)\ \ \bot\vee A\approx A\qquad (S_4)\ \ A\vee A\vee B\approx A\vee B$$

Inference rules

$$\text{(R1)}\ \frac{C}{D}\ \text{if}\ \Gamma(C)\Rightarrow D\qquad\qquad \text{(R2)}\ \frac{C_1\qquad C_2}{D}\ \text{if}\ \Sigma(C_1,C_2)\Rightarrow D$$

Clarifications: A, B, C and D denote formulas in disjunctive normal form (DNF) whereas E denotes a formula in conjunctive normal form (CNF). A formula is in DNF if it has the general form $L_1\vee\ldots\vee L_n\vee\Box A_1\vee\ldots\vee\Box A_p\vee\Diamond E_1\vee\ldots\vee\Diamond E_q$, where L_i are literals, A_i are in DNF and E_i are in CNF. A formula is in CNF if it is a conjunction of formulas in DNF. The relation $\approx$ is the least congruence satisfying all simplification rules. The normal form A of a formula A' is the least formula such that $A'\approx A$. We write $\Sigma(A,B)\Rightarrow C$ (respectively $\Gamma(A)\Rightarrow C$) if there exist C' such that $\Sigma(A,B)\longrightarrow C'$ (resp. $\Gamma(A)\longrightarrow C'$) and C is the normal form of C'.

History: Introduced in [1]. The current presentation is inspired by [4]. The method is at the core of the MOLOG language [3]. With slight variations of the rules, some other modal logics like S4 or S5 can be obtained [4]. The method has been adaptated to first-order modal logic [5]. An alternative non-clausal resolution method is presented in [2] (for LTL).

Remarks: The method is sound and complete with respect to the classical modal logic K.

[1] Luis Fariñas del Cerro. "A Simple Deduction Method for Modal Logic". In: *Inf. Process. Lett.* 14.2 (1982), pp. 49–51.

[2] Martín Abadi and Zohar Manna. "Nonclausal Temporal Deduction". In: *Logics of Programs*. Ed. by Rohit Parikh. Vol. 193. LNCS. Springer, 1985, pp. 1–15.

[3] Pierre Bieber, Luis Fariñas del Cerro, and Andreas Herzig. "MOLOG: a Modal PROLOG". In: *CADE 9*. Ed. by Ewing L. Lusk and Ross A. Overbeek. Vol. 310. LNCS. Springer, 1988, pp. 762–763.

[4] Patrice Enjalbert and Luis Fariñas del Cerro. "Modal Resolution in Clausal Form". In: *Theor. Comput. Sci.* 65.1 (1989), pp. 1–33.

[5] Marta Cialdea. "Resolution for Some First-Order Modal Systems". In: *Theor. Comput. Sci.* 85.2 (1991), pp. 213–229.

Entry 20 by: Joseph Boudou, Luis Fariñas del Cerro

Expansion Proofs (1983)

Expansion trees, *eigenvariables*, and the function $\mathsf{Sh}(-)$ (read *shallow formula of*), that maps an expansion tree to a formula, are defined as follows:

1. If A is $\top$ (true), $\bot$ (false), or a literal, then A is an expansion tree with top node A, and $\mathsf{Sh}(A) = A$.
2. If E is an expansion tree with $\mathsf{Sh}(E) = [y/x]A$ and y is not an eigenvariable of any node in E, then $E' = \forall x.A +^y E$ is an expansion tree with top node $\forall x.A$ and $\mathsf{Sh}(E') = \forall x.A$. The variable y is called an *eigenvariable* of (the top node of) E'. The set of eigenvariables of all nodes in an expansion tree is called the *eigenvariables of* the tree.
3. If $\{t_1, \ldots, t_n\}$ (with $n \geq 0$) is a set of terms and $E_1, \ldots, E_n$ are expansion trees with pairwise disjoint eigenvariable sets and with $\mathsf{Sh}(E_i) = [t_i/x]A$ for $i \in \{1, \ldots, n\}$, then $E' = \exists x.A +^{t_1} E_1 \ldots +^{t_n} E_n$ is an expansion tree with top node $\exists x.A$ and $\mathsf{Sh}(E') = \exists x.A$. The terms $t_1, \ldots, t_n$ are known as the *expansion terms* of (the top node of) E'.
4. If E_1 and E_2 are expansion trees that share no eigenvariables and $\circ \in \{\wedge, \vee\}$, then $E_1 \circ E_2$ is an expansion tree with top node $\circ$ and $\mathsf{Sh}(E_1 \circ E_2) = \mathsf{Sh}(E_1) \circ \mathsf{Sh}(E_2)$.

In the expansion tree $\forall x.A +^x E$ (resp. in $\exists x.A +^{t_1} E_1 \ldots +^{t_n} E_n$), we say that x (resp. t_i) *labels* the top node of E (resp. E_i, for any $i \in \{1, \ldots, n\}$). A term t *dominates* a node in an expansion tree if it labels a parent node of that node in the tree.

For an expansion tree E, the quantifier-free formula $\mathsf{Dp}(E)$, called the *deep formula of* E, is defined as:

- $\mathsf{Dp}(E) = E$ if E is $\top$, $\bot$, or a literal;
- $\mathsf{Dp}(E_1 \circ E_2) = \mathsf{Dp}(E_1) \circ \mathsf{Dp}(E_2)$ for $\circ \in \{\wedge, \vee\}$;
- $\mathsf{Dp}(\forall x.A +^y E) = \mathsf{Dp}(E)$; and
- $\mathsf{Dp}(\exists x.A +^{t_1} E_1 \cdots +^{t_n} E_n) = \mathsf{Dp}(E_1) \vee \ldots \vee \mathsf{Dp}(E_n)$ if $n > 0$, and $\mathsf{Dp}(\exists x.A) = \bot$.

Let $\mathcal{E}$ be an expansion tree and let $<^0_{\mathcal{E}}$ be the binary relation on the occurrences of expansion terms in $\mathcal{E}$ defined by $t <^0_{\mathcal{E}} s$ if there is an x which is free in s and which is the eigenvariable of a node dominated by t. Then $<_{\mathcal{E}}$, the transitive closure of $<^0_{\mathcal{E}}$, is called the *dependency relation* of $\mathcal{E}$.

An expansion tree $\mathcal{E}$ is said to be an *expansion proof* if $<_{\mathcal{E}}$ is acyclic and $\mathsf{Dp}(\mathcal{E})$ is a tautology; in particular, $\mathcal{E}$ is an *expansion proof of* $\mathsf{Sh}(\mathcal{E})$.

Clarifications: The soundness and completeness theorem for expansion trees is the following. A formula B is a theorem of first-order logic if and only if there is an expansion proof Q such that $\mathsf{Sh}(Q) = B$.

History: Expansion trees and proofs [2, 1] generalize Herbrand's disjunctions and Gentzen's mid-sequents to the non-prenex case. They were originally defined for higher-order classical logic and used to prove soundness of skolemization and a generalization of Herbrand's theorem for this logic. Expansion trees are an early example of a matrix-based proof system emphasizing parallelism in a manner similar to that found in proof nets {26}. That parallelism is explicitly analyzed in [3] using a multi-focused version of LKF {67}.

[1] Dale Miller. "Proofs in Higher-order Logic". PhD thesis. Carnegie-Mellon University, Aug. 1983.

[2] Dale Miller. "A Compact Representation of Proofs". In: *Studia Logica* 46.4 (1987), pp. 347–370.

[3] Kaustuv Chaudhuri, Stefan Hetzl, and Dale Miller. "A Multi-Focused Proof System Isomorphic to Expansion Proofs". In: *J. of Logic and Computation* (June 2014).

Entry 21 by: Dale Miller

Bledsoe's Natural Deduction - Prover (1973-1978)

SPLIT: basic rules of Natural Deduction(see {5}), for example
To prove $A \wedge B$, prove A and prove B
To prove $p \rightarrow A \wedge B$, prove $(p \rightarrow A) \wedge (p \rightarrow B)$
To prove $p \vee q \rightarrow A$, prove $(p \rightarrow A) \wedge (q \rightarrow A)$
To prove $\exists x P(x) \rightarrow D$, prove $P(y) \rightarrow D$, where y is a new variable
REDUCE: conversion rules, for example
To prove $x \in A \cap B$, prove $x \in A \wedge x \in B$
To prove $S \in \mathcal{P}(A)$, prove $S \subset A \wedge S \in \mathcal{U}$
To prove $x \in \sigma F$, prove $\exists y(y \in F \wedge x \in y)$
DEFINITIONS, example
$A \subset B$ is defined by $\forall x(x \in A \rightarrow x \in B)$ and is replaced by $x \in A \rightarrow x \in B$ or by $x_o \in A \rightarrow x_o \in B$, depending on the position of the formula in the theorem.
IMPLY: in addition to SPLIT and REDUCE rules,
- search for substitutions which unify some hypotheses and a conclusion and compose them until obtaining the empty substitution (theorem proved) or failing
- forward chaining : if P and P' are unified by θ ($P\theta$=$P'\theta$), then a hypothesis $P' \wedge (P \rightarrow Q)$ is converted into $P' \wedge (P \rightarrow Q) \wedge Q\theta$
- PEEK forward chaining : if $P\theta$=$P'\theta$ and A has the definition $(P \rightarrow Q)$, then a hypothesis $P' \wedge A$ is converted into $P' \wedge A \wedge Q\theta$
- backward chaining : if $A \rightarrow D$ and $D\theta = C\theta$, replace the conclusion C by $A\theta$

Clarifications: Bledsoe's natural deduction may be seen as both an extension and a restriction of formal natural deduction {5}. In SPLIT and REDUCE, there is reduction but not expansion. Some subroutines convert expressions into forms convenient for applying the rules. The notions of hypothesis and conclusion are privileged.

History: After having applied the rules of IMPLY and REDUCE, the first version of **Prover** [1] called a resolution program if necessary. Then, in [2], these calls to resolution are completely replaced by IMPLY. **Prover** has been working in set theory, limit theorems, topology and program verification.

Remarks: The system is sound but not complete. Bledsoe emphasizes the fact that, with these methods, provers may succeed because they proceed in a natural human-like way [3].

---

[1] W. W. Bledsoe. "Splitting and reduction heuristics in automatic theorem proving". In: *Artificial Intelligence* 2 (1971), pp. 55–77.
[2] W. W. Bledsoe, R. S. Boyer, and W. H. Henneman. "Computer proofs of Limit theorems". In: *Artificial Intelligence* 3 (1972), pp. 27–60.
[3] W. W. Bledsoe. "Non-resolution theorem proving". In: *Artificial Intelligence* 9 (1977), pp. 1–35.

Entry 22 by: Dominique Pastre

Natural Knowledge Bases - Muscadet (1984)

Some of the rules given to the system :
Basic rules of Natural Deduction (similar to Bledsoe's SPLIT rules {22}).
Flatten : Replace $P(f(x))$ by $\exists y(y : f(x) \wedge P(y))$ or by $\forall y(y : f(x) \Rightarrow P(y))$ depending on the position (positive or negative) of the formula in the theorem to be proved and in the definitions and lemmas.
Rules automatically built by metarules from definitions :
If $A \subset B$ and $x \in A$ then $x \in B$ If $x \in \sigma E$, then $\exists y(y \in E \wedge x \in y)$
If $C : A \cap B$ and $x \in C$, then $x \in A$ If $C : A \cap B$, $x \in A$ and $x \in B$, then $x \in C$
in place of (and more general than) given REDUCE conversion rules of {22}.
and from universal hypotheses :
Universal hypotheses are removed and replaced by local rules (for a sub-theorem).
This replaces and generalizes PEEK forward-chaining of {22}.

Clarifications: "If $C : A \cap B$" expresses that C is $A \cap B$ which has already been introduced. Flattening is used to recursively create and name objects such as $f(x)$, and in a certain manner to "eliminate" functional symbols since the expression $y : f(x)$ will be handled as if it was a predicate expression $F(x)$.
Rules are conditional actions. Actions may be defined by packs of rules. Metarules build rules from definitions, lemmas and universal hypotheses.

History: **Muscadet** [1, 2] is a knowledge-based system. Facts are hypotheses and the conclusion of a theorem or a sub-theorem to be proved, and all sorts of facts which give relevant information during the proof search process. Universal hypotheses are handled as local definitions (no skolemization). **Muscadet** worked in set theory, mappings and relations, topology and topological linear spaces, elementary geometry, discrete geometry, cellular automata, and TPTP problems. It attended CASC competitions. It is open software, freely available.
Muscadet is efficient for everyday mathematical problems which are expressed in a natural manner, and problems which involve many axioms, definitions or lemmas, but not for problems with only one large conjecture and few definitions.

Remarks: The system is sound but not complete (because of the use of many selective rules and heuristics). It displays proofs easily readable by a human reader.

---

[1] D. Pastre. "MUSCADET : An Automatic Theorem Proving System using Knowledge and Metaknowledge in Mathematics". In: *Artificial Intelligence* 38.3 (1989), pp. 257–318.
[2] D. Pastre. "Automated Theorem Proving in Mathematics". In: *Annals on Artificial Intelligence and Mathematics* 8.3-4 (1993), pp. 425–447.
[3] D. Pastre. *Muscadet version 4.1 : user's manual*. 2011, pp. 1–22. URL: `http://www.normalesup.org/~pastre/muscadet/manual-en.pdf`.

Entry 23 by: Dominique Pastre

Intuitionistic Linear Logic (ILL) (1987)

Structural

$$\frac{}{A \vdash A} \qquad \frac{\Gamma \vdash A \quad A,\Delta \vdash B}{\Gamma,\Delta \vdash B}\ (cut) \qquad \frac{\Gamma,A,B,\Delta \vdash C}{\Gamma,B,A,\Delta \vdash C}$$

Multiplicative

$$\frac{}{\vdash 1} \qquad \frac{\Gamma \vdash A}{\Gamma,1 \vdash A} \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma,\Delta \vdash A \otimes B} \qquad \frac{\Gamma,A,B \vdash C}{\Gamma,A \otimes B \vdash C}$$

$$\frac{\Gamma,A \vdash B}{\Gamma \vdash A \multimap B} \qquad \frac{\Gamma \vdash A \quad B,\Delta \vdash C}{\Gamma,A \multimap B,\Delta \vdash C}$$

Additive

$$\frac{}{\Gamma \vdash \top} \qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} \qquad \frac{\Gamma,A_i \vdash B}{\Gamma,A_1 \& A_2 \vdash B}$$

$$\frac{}{\Gamma,0 \vdash A} \qquad \frac{\Gamma \vdash A_i}{\Gamma \vdash A_1 \oplus A_2} \qquad \frac{\Gamma,A \vdash C \quad \Gamma,B \vdash C}{\Gamma,A \oplus B \vdash C}$$

Exponential

$$\frac{!\Gamma \vdash A}{!\Gamma \vdash !A} \qquad \frac{\Gamma \vdash B}{\Gamma,!A \vdash B} \qquad \frac{\Gamma,A \vdash B}{\Gamma,!A \vdash B} \qquad \frac{\Gamma,!A,!A \vdash B}{\Gamma,!A \vdash B}$$

Clarifications: Succedents are single formulas. Antecedents are ordered list of formulas. If Γ is the list $A_1,\ldots,A_n$ of formulas, $!\Gamma$ denotes the list $!A_1,\ldots,!A_n$. First order quantifiers can be added with rules similar to **LJ** {7}. Conversely, removing the exponential rules leads to the intuitionistic multiplicative additive linear logic (IMALL). And by further removing the additive rules, the intuitionistic multiplicative linear logic (IMLL) [1] is obtained.

History: Introduced by Girard and Lafont in [2] as intuitionistic variant of **LL** {25}. **ILL** has multiple applications in categorical logic.

Remarks: Enjoys cut elimination [2].

---

[1] Grigorii Efroimovich Mints. "Closed categories and the theory of proofs". In: *Zapiski Nauchnykh Seminarov POMI* 68 (1977), pp. 83–114.

[2] Jean-Yves Girard and Yves Lafont. "Linear Logic and Lazy Computation". In: *Theory and Practice of Software Development*. 1987, pp. 52–66.

Entry 24 by: Joseph Boudou, Sergei Soloviev

Linear Sequent Calculus LL (1987)

$$\frac{}{\vdash A^\perp, A} \qquad \frac{\vdash \Gamma, A \qquad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta} \qquad \frac{\vdash \Gamma}{\vdash \sigma(\Gamma)}$$

$$\frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B} \qquad \frac{}{\vdash 1} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \qquad \frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B} \qquad \frac{}{\vdash \Gamma, \top}$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} \qquad \frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}$$

$$\begin{array}{cccc} & (A \otimes B)^\perp = A^\perp \parr B^\perp & & 1^\perp = \perp \\ (X^\perp)^\perp = X & (A \parr B)^\perp = A^\perp \otimes B^\perp & & \perp^\perp = 1 \\ (!A)^\perp = ?(A^\perp) & (A \oplus B)^\perp = A^\perp \,\&\, B^\perp & & 0^\perp = \top \\ (?A)^\perp = !(A^\perp) & (A \,\&\, B)^\perp = A^\perp \oplus B^\perp & & \top^\perp = 0 \end{array}$$

Γ and Δ are lists of formulas.
σ is a permutation.

Clarifications: If $\Gamma = A_1, \ldots, A_n$ then $?\Gamma = ?A_1, \ldots, ?A_n$. Negation is not a connective. It is defined using De Morgan's laws so that $(A^\perp)^\perp = A$. The linear implication can be defined as $A \multimap B = A^\perp \parr B$.

History: Linear Logic and its sequent calculus **LL** [1] come from the analysis of intuitionistic logic through Girard's decomposition of the intuitionistic implication into the linear implication: $A \to B = !A \multimap B$.

Remarks: Cut elimination holds. **LL** is sound and complete with respect to phase semantics [1]. **LL** is not decidable [2]. Sequent calculi **LK** {6} and **LJ** {7} can be translated into **LL**.

[1] Jean-Yves Girard. "Linear logic". In: *Theoretical Computer Science* 50 (1987), pp. 1–102.

[2] Patrick Lincoln, John Mitchell, Andre Scedrov, and Natarajan Shankar. "Decision problems for propositional linear logic". In: *Annals of pure and applied logic* 56.1-3 (1992), pp. 239–311.

Entry 25 by: Olivier Laurent

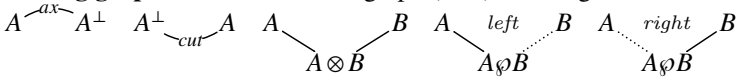
Proof Nets for MLL^- (1987)

Links : $\dfrac{}{A^\perp \quad A}\,axiom$ $\qquad$ $\dfrac{A \qquad A^\perp}{cut}$ $\qquad$ $\dfrac{A \qquad B}{A \otimes B}$ $\qquad$ $\dfrac{A \qquad B}{A \parr B}$

Proof Structure $\mathcal{R}(R, L)$ **:** a nonempty set R of formula occurrences, with a set L of links, such that each $A \in R$ is a conclusion of *exactly one* link and a premise of *at most one* link. If A is not a premise, then it is a **conclusion of** $\mathcal{R}$. (*Cut* links behave like *times* links with conclusion $A \otimes A^\perp$.)

Switching s: a choice for every *par* link ℓ of one premise, $s(\ell)$= '*left*' or '*right*'.

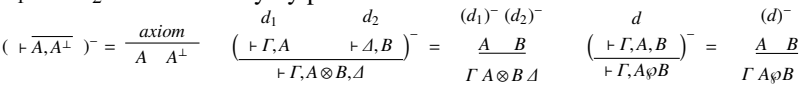
Switching graph $s\mathcal{R}$: an undirected graph (R, E) with edges E as follows:



Proof net: A proof structure $\mathcal{R}$ such that for every switching s the graph $s\mathcal{R}$ is *acyclic* and *connected* (Danos Regnier's *correctness criterion* [2]).

Clarifications:

1. The forest of sub-formulas of a multiset $\Gamma = C_1, \ldots, C_n$, with a partition of the leaves in unordered pairs $(p, p^\perp)$ is a cut-free proof-structure. Also $\dfrac{A \quad \overline{\quad axiom \quad}\quad A^\perp}{cut}$ is a proof structure.
2. Proof nets are canonical representations of $\mathbf{MLL}^-$ sequent calculus proofs and solve the proof identity problem for $\mathbf{MLL}^-$ in linear time. The *desequentialization map* $(\ )^-$ identifies sequent calculus derivations d_1 and d_2 that differ only by permutations of inferences:

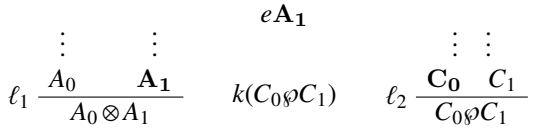


3. $\mathcal{R}_1(R_1, L_1)$ is a *subnet* of $\mathcal{R}_2(R_2, L_2)$ if $R' \subseteq R$ and $L_1 = L_2|_{R_1}$.

   Lemma 1. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be subnets of $\mathcal{R}$ with $\mathcal{R}_1 \cap \mathcal{R}_2 \neq \emptyset$. Then $\mathcal{S} = \mathcal{R}_1 \cup \mathcal{R}_2$ is a subnet of $\mathcal{R}$. **Proof.** Since any $s\mathcal{R}$ is acyclic, so is its subgraph $s\mathcal{S}$. Given $A \in R_1$, $B \in R_2$ and $C \in (R_1 \cap R_2)$, A is connected to C in $\mathcal{R}_1$ and C is connected to B in $\mathcal{R}_2$, so A is connected to C in $\mathcal{S}$. **qed**

   The *empire* eA, for $A \in \mathcal{R}$, is the *largest subnet having A as a conclusion*. If $s_A\mathcal{R}$ is the subgraph of $s\mathcal{R}$ with the vertex A as root, then $eA = \bigcap_s s_A\mathcal{R}$. The *kingdom* kA of A is the *smallest* subnet having A as conclusion.

   Lemma 2. Let ℓ_1 and ℓ_2 be links in $\mathcal{R}$ with conclusions $A_0 \otimes A_1$ and $C_0 \parr C_1$, respectively. If $C_i \in eA_j$ but $C_0 \parr C_1 \notin eA_j$ then $A_0 \otimes A_1 \in k(C_0 \parr C_1)$.



Entry 26 by: Gianluigi Bellin

Proof. Let $C_0 \in eA_1$; clearly $C_0 \in k(C_0 ⅋ C_1)$ so $\mathcal{S} = eA_1 \cup k(C_0 ⅋ C_1) \neq \emptyset$ and by Lemma 1 is a subnet. Suppose $A_0 \otimes A_1$ does not belong to $k(C_0 ⅋ C_1)$; then $\mathcal{S}$ has A_1 as conclusion and is larger than eA_1, a contradiction. **qed**

Sequentialization Theorem. If $\mathcal{R}$ is a proof net for $\mathbf{MLL}^-$ with conclusions *Gamma*, then a sequent calculus derivation d of $\vdash$ *Gamma* can be constructed such that $(d)^- = \mathcal{R}$.

Proof sketch. By induction on the number of links of eA. Terminal *par* links can be deleted and the result follows from the induction hypothesis. Suppose the non-atomic conclusions of eA are $A_0 \otimes B_0, \ldots, A_n \otimes B_n$. We need to find a *splitting link* ℓ_i with conclusion $A_i \otimes B_i$, such that by removing ℓ_i the net splits in two disjoint components $e(A_i)$ and $e(B_i)$. We choose an ℓ_i such that $A_i \otimes B_i$ is not included in $k(A_j \otimes B_j)$ for $j \neq i$. If all conclusions of eA_i are conclusions of eA, we are done. Otherwise let ℓ be a link such that a premise C is in eA_i, but the conclusion is not. Then ℓ must be a *par* link with conclusion, say, $C ⅋ D$. By Lemma 2 $A_i \otimes B_i \in k(C ⅋ D)$. But $C ⅋ D$ must occur above a link ℓ_j with conclusion $A_j \otimes B_j$. It follows that $(A_i \otimes B_i) \in k(C ⅋ D) \subset k(A_j \otimes B_j)$ contrary to the choice of ℓ_i. **qed**

History: Proof nets for $\mathbf{MLL}^-$ were introduced by J-Y.Girard in 1987 [1]. Simplifications were given by Danos and Regnier [2] and in [3]. Since then many systems of proof nets were found for larger fragments of linear logic, with additives (D. Hughes and R. van Glabbek) and for variants of linear logic, with *Mix* (A. Fleury, C. Retoré, G. Bellin) and F. Lamarche's *essential nets* for intuitionistic linear logic). In 1999 S. Guerrini showed that correctness of multiplicative proof-nets without units is linear. For **MLL** *with the units* proof-nets are non-canonical with respect to permutation of inferences in the sequent calculus. In 2014 W. Heijltjes and R. Houston showed that the identity problem for **MLL** proofs is PSPACE complete.

---

[1] J-Y. Girard. "Linear Logic". In: *Theoretical Computer Science* 50 (1987), pp. 1–202.

[2] V. Danos and L. Regnier. "The structure of multiplicatives". In: *Arch. Math. Logic* 28 (1989).

[3] G. Bellin and J. van de Wiele. "Subnets of Proof-nets in $\mathbf{MLL}^-$". In: *Advances in Linear Logic*. London Math. Soc. L.N.S. 222 CUP, 1995, pp. 249–270.

Entailment for Structured Specifications (1988)

$$\frac{SP \vdash \varphi_1 \quad \cdots \quad SP \vdash \varphi_n \qquad \{\varphi_1,\ldots,\varphi_n\} \vdash_{Sig[SP]} \varphi}{SP \vdash \varphi}$$

$$\frac{}{\langle \Sigma, \Phi\rangle \vdash \varphi}\ \varphi \in \Phi$$

$$\frac{SP_1 \vdash \varphi}{SP_1 \cup SP_2 \vdash \varphi} \qquad \frac{SP_2 \vdash \varphi}{SP_1 \cup SP_2 \vdash \varphi}$$

$$\frac{SP \vdash \varphi}{SP \textbf{ with } \sigma \vdash \sigma(\varphi)} \qquad \frac{SP \vdash \sigma(\varphi)}{SP \textbf{ hide via } \sigma \vdash \varphi}$$

Clarifications: $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}\colon \mathbf{Sign} \to \mathbf{Set}, \mathbf{Mod}\colon \mathbf{Sign}^{op} \to \mathbf{Cat}, \langle \models_\Sigma \subseteq |\mathbf{Mod}(\Sigma)| \times \mathbf{Sen}(\Sigma)\rangle_{\Sigma \in |\mathbf{Sign}|}\rangle$ is an institution that defines the logical system used for specifications, SP, SP_1 and SP_2 are structured Σ-specifications over $\mathbf{INS}$, where Σ is a signature in the category $\mathbf{Sign}$, $\varphi, \varphi_1, \ldots, \varphi_n$ are Σ-sentences, i.e. elements in $\mathbf{Sen}(\Sigma)$, Φ is a set of Σ-sentences, and $\sigma(\varphi)$ denotes $\mathbf{Sen}(\sigma)(\varphi)$, the translation of the sentence φ along $\sigma\colon \Sigma \to \Sigma'$. Structured specifications in $\mathbf{INS}$ are built from basic specifications $\langle \Sigma, \Phi\rangle$, the union of Σ-specifications $SP_1 \cup SP_2$, the translation "SP **with** σ" of SP along a signature morphism $\sigma\colon \Sigma \to \Sigma'$, and hiding "$SP$ **hide via** σ" for hiding the symbols in SP not occurring in the image of $\sigma\colon \Sigma' \to \Sigma$. $Sig[SP]$ is the signature of SP. Translations of Σ-sentences and Σ'-models along $\sigma\colon \Sigma \to \Sigma'$ are required to preserve satisfaction: for any $\varphi \in \mathbf{Sen}(\Sigma)$ and $M' \in |\mathbf{Mod}(\Sigma')|$, $M' \models_{\Sigma'} \mathbf{Sen}(\sigma)(\varphi) \Leftrightarrow \mathbf{Mod}(\sigma)(M') \models_\Sigma \varphi$. Finally, $\langle \vdash_\Sigma \subseteq Pow(\mathbf{Sen}(\Sigma)) \times \mathbf{Sen}(\Sigma)\rangle_{\Sigma \in |\mathbf{Sign}|}$ is a sound entailment relation for the satisfaction relation $\langle \models_\Sigma\rangle_{\Sigma \in |\mathbf{Sign}|}$.

The judgment $SP \vdash \varphi$ is meant to capture the property that φ is satisfied in all models of SP.

History: The first systems for proving entailment in structured specifications were given by Sannella and Burstall [1], Sannella and Tarlecki [2], and Wirsing [3]. The above presentation can be found in [6], Sect. 9.2.

Remarks: The system is sound; completeness is shown in [3] for the first-order instance and in [5, 6] for an institution $\mathbf{INS}$ which is finitely exact, admits propositional operators, satisfies Craig interpolation, and has a complete entailment relation $\langle \vdash_\Sigma\rangle_{\Sigma \in |\mathbf{Sign}|}$. [7] shows that this is the most powerful sound proof system that is compositional in the structure of specifications. [4] provides additional rules for observability operators.

[1] Donald Sannella and Rod M. Burstall. "Structured Theories in LCF". In: *CAAP'83, Trees in Algebra and Programming, 8th Colloquium, Proc.* Vol. 159. LNCS. Springer, 1983, pp. 377–391.

[2] Donald Sannella and Andrzej Tarlecki. "Specifications in an Arbitrary Institution". In: *Information and Computation* 76.2/3 (1988), pp. 165–210.

[3] Martin Wirsing. "Structured Specifications: Syntax, Semantics and Proof Calculus". In: *Logic and Algebra of Specification, NATO Advanced Institute, 1991*. Vol. 94. Springer, 1993, pp. 411–442.

[4] Rolf Hennicker. *Structured Specifications with Behavioural Operators: Semantics, Proof Methods and Applications*. Habilitation thesis. LMU Munich, 1997.

[5] Tomasz Borzyszkowski. "Logical systems for structured specifications". In: *Theoretical Computer Science* 286.2 (2002), pp. 197–245.

[6] Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2012.

[7] Donald Sannella and Andrzej Tarlecki. "Property-oriented semantics of structured specifications". In: *Mathematical Structures in Computer Science* 24.2 (2014).

Entry 27 by: Rolf Hennicker, Donald Sannella, Andrzej Tarlecki, Martin Wirsing

Pure Type Systems (1989)

$$\frac{}{\vdash c : s}\ axiom\ ((c:s)\in\mathcal{A}) \qquad \frac{\Gamma\vdash A:s}{\Gamma,x:A\vdash x:A}\ start\ (x\notin\Gamma)$$

$$\frac{\Gamma\vdash M:B \qquad \Gamma\vdash A:s}{\Gamma,x:A\vdash M:B}\ weakening\ (x\notin\Gamma) \qquad \frac{\Gamma\vdash A:s_1 \qquad \Gamma,x:A\vdash B:s_2}{\Gamma\vdash \Pi x:A.B:s_3}\ product\ ((s_1,s_2,s_3)\in\mathcal{R})$$

$$\frac{\Gamma,x:A\vdash M:B \qquad \Gamma\vdash \Pi x:A.B:s}{\Gamma\vdash \lambda x:A.M:\Pi x:A.B}\ abstraction \qquad \frac{\Gamma\vdash M:\Pi x:A.B \qquad \Gamma\vdash N:A}{\Gamma\vdash M\ N:B[x:=N]}\ application$$

$$\frac{\Gamma\vdash M:A \qquad \Gamma\vdash B:s \qquad A\equiv_\beta B}{\Gamma\vdash M:B}\ conversion$$

Clarifications: *Pure type systems* (PTS) are a general class of typed λ calculus. They represent logical systems through the Curry-Howard correspondence and the "propositions as types" interpretation. The syntax is given by the grammar:

$$\mathcal{T} ::= \mathcal{V} \mid C \mid \Pi\mathcal{V}:\mathcal{T}.\mathcal{T} \mid \lambda\mathcal{V}:\mathcal{T}.\mathcal{T} \mid \mathcal{T}\ \mathcal{T}$$

where $\mathcal{V}$ is a set of variables and C is a set of constants. A PTS is parameterized by a *specification* $(\mathcal{S},\mathcal{A},\mathcal{R})$ where $\mathcal{S}\subseteq C$ is the set of *sorts*, $\mathcal{A}\subseteq C\times\mathcal{S}$ is the set of *axioms*, and $\mathcal{R}\subseteq\mathcal{S}\times\mathcal{S}\times\mathcal{S}$ is the set of *rules*.

History: Pure type systems were independently introduced by Berardi and Terlouw as a generalization of systems of the λ cube, and further developed and popularized by Barendregt, Geuvers, Nederhof [1, 2, 3, 4]. Many important systems can be expressed as PTSs, including simply typed λ calculus ($\lambda\!\!\rightarrow$), $\lambda\Pi$ calculus {41} (λP), system F {18} ($\lambda 2$), and the calculus of constructions (λC):

$$\mathcal{S}=\{*,\Box\} \qquad \mathcal{A}=\{(*,\Box)\} \qquad \mathcal{R}_{\rightarrow}=\{(*,*,*)\}$$

$$\mathcal{R}_P=\mathcal{R}_{\rightarrow}\cup\{(*,\Box,\Box)\} \qquad \mathcal{R}_2=\mathcal{R}_{\rightarrow}\cup\{(\Box,*,*)\} \qquad \mathcal{R}_C=\mathcal{R}_P\cup\mathcal{R}_2\cup\{(\Box,\Box,\Box)\}$$

as well as intuitionistic higher-order logic (λHOL). Pure type systems form the basis of many proof assistants such as Automath, Lego, Coq, Agda, and Matita.

Remarks: Soundness and decidability of type checking in PTSs are closely related to *strong normalization* (SN), i.e. the property that all well-typed terms terminate. Not all pure type systems are SN. Examples of PTSs that are *not* SN (and are therefore inconsistent) are Girard's system U and the universal PTS $\lambda *$:

$$\mathcal{S}=\{*\} \qquad \mathcal{A}=\{(*,*)\} \qquad \mathcal{R}=\{(*,*,*)\}$$

[1] Henk Barendregt. "Introduction to generalized type systems". In: *Journal of Functional Programming* 1.2 (1991), pp. 125–154.

[2] Herman Geuvers and Mark-Jan Nederhof. "Modular proof of strong normalization for the calculus of constructions". In: *Journal of Functional Programming* 1.2 (1991), pp. 155–189.

[3] Henk Barendregt. "Lambda Calculi with Types". In: *Handbook of Logic in Computer Science*. Ed. by Samson Abramsky, Dov M. Gabbay, and Thomas S. E. Maibaum. Vol. 2. Oxford University Press, 1992, pp. 117–309.

[4] Herman Geuvers. "Logics and type systems". PhD thesis. University of Nijmegen, 1993.

Entry 28 by: Ali Assaf

Full Intuitionistic Linear Logic (FILL) (1990)

$$\frac{}{x : A \vdash x : A}\ Ax \qquad \frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma', y : A \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta \mid [t/y]\Delta'}\ Cut$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, x : \top \vdash \mathsf{let}\, x \,\mathsf{be}\, * \,\mathsf{in}\, \Delta}\ \top_L \qquad \frac{}{\cdot \vdash * : \top}\ \top_R$$

$$\frac{}{x : \bot \vdash \cdot}\ \bot_L \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \circ : \bot \mid \Delta}\ \bot_R$$

$$\frac{\Gamma, x : A, y : B \vdash \Delta}{\Gamma, z : A \otimes B \vdash \mathsf{let}\, z \,\mathsf{be}\, x \otimes y \,\mathsf{in}\, \Delta}\ \otimes_L \qquad \frac{\Gamma \vdash t_1 : A \mid \Delta \quad \Gamma' \vdash t_2 : B \mid \Delta'}{\Gamma, \Gamma' \vdash t_1 \otimes t_2 : A \otimes B \mid \Delta \mid \Delta'}\ \otimes_R$$

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma', x : B \vdash t_i : C_i}{\Gamma, y : A \multimap B, \Gamma' \vdash [y\, t/x] t_i : C_i \mid \Delta}\ \multimap_L \qquad \frac{\Gamma, x : A \vdash t : B \quad x \notin \mathsf{FV}(\Delta)}{\Gamma \vdash \lambda x.t : A \multimap B \mid \Delta}\ \multimap_R$$

$$\frac{\Gamma, x : A \vdash t_i : C_i \quad \Gamma', y : B \vdash t_j : D_j}{\Gamma, \Gamma', z : A \parr B \vdash \mathsf{let\text{-}pat}\, z\, (x \parr -)\, t_i : C_i \mid \mathsf{let\text{-}pat}\, z\, (- \parr y)\, t_j : D_j}\ \parr_L$$

$$\frac{\Gamma \vdash \Delta \mid t_1 : A \mid t_2 : B \mid \Delta'}{\Gamma \vdash \Delta \mid t_1 \parr t_2 : A \parr B \mid \Delta'}\ \parr_R$$

Clarifications: Both the left-hand and right-hand sides of sequents above are multisets of formulas, denoted Γ and Δ. The terms annotating formulas are standard terms used in the simply typed λ-calculus. Capture avoiding substitution is denoted by $[t/x]t'$, and uniformly replaces every occurrence of x in t' with t. The definition of the let-pattern function used in the rule $\parr_L$ is defined as follows:

$$\mathsf{let\text{-}pat}\, z\, (x \parr -)\, t = t \quad \text{where } x \notin \mathsf{FV}(t) \qquad \mathsf{let\text{-}pat}\, z\, (- \parr y)\, t = t \quad \text{where } y \notin \mathsf{FV}(t) \qquad \mathsf{let\text{-}pat}\, z\, p\, t = \mathsf{let}\, z \,\mathsf{be}\, p \,\mathsf{in}\, t$$

We denote vectors of terms (resp. types) by t_i (resp. A_j). The function $\mathsf{FV}(\Delta)$ constructs the set of all free variables in each term found in Δ.

History: The original formulation of FILL by Valeria de Paiva in her thesis [1] did not satisfy cut-elimination, as shown by Schellinx. Martin Hyland and Valeria de Paiva [2] added a term assignment system to cope with the notion of dependency in the right implication rule and obtain cut-elimination. However, there was still a mistake in the par rule in [2], which was corrected independently, with different proof methods, by Bierman [3], Bellin [4], Brauner/dePaiva [5], dePaiva/Ritter [6]. The version here is the minimal modification suggested by Bellin, (who used proofnets), but using a traditional term assignment, as described in Eades/dePaiva [7].

---

[1] Valeria de Paiva. "The Dialectica Categories". PhD thesis. University of Cambridge, 1990.

[2] Martin Hyland and Valeria de Paiva. "Full intuitionistic linear logic (extended abstract)". In: *Annals of Pure and Applied Logic* 64.3 (1993), pp. 273–291.

[3] Gavin Bierman. "A note on full intuitionistic linear logic". In: *Annals of Pure and Applied Logic* 79.3 (1996), pp. 281–287.

[4] Gianluigi Bellin. "Subnets of proof-nets in multiplicative linear logic with MIX". In: *Mathematical Structures in Computer Science* 7 (Dec. 1997), pp. 663–669.

Entry 29 by: Harley Eades III, Valeria de Paiva

[5] Torben Braüner and Valeria de Paiva. "A formulation of linear logic based on dependency-relations". In: *Computer Science Logic*. Ed. by Mogens Nielsen and Wolfgang Thomas. Vol. 1414. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1998, pp. 129–148.

[6] Valeria de Paiva and Eike Ritter. "A Parigot-style Linear Lambda-calculus for Full Intuitionistic Linear Logic". In: *Theory and Applications of Categories* 17.3 (2006), pp. 127–152.

[7] Harley Eades III and Valeria de Paiva. "Multiple Conclusion Intuitionistic Linear Logic and Cut Elimination". `http://metatheorem.org/papers/FILL-cut-report.pdf`.

Signed Analytic Calculi for Finite-Valued Logics (1990)

$$\frac{}{\vdash S_1{:}\varphi, \ldots S_l{:}\varphi, \Delta}\ \text{Axiom} \quad \text{where } \bigcup_{k=0}^{l} S_k = N \qquad \frac{\{\vdash S^i_1{:}\varphi_1, \ldots, S^i_r{:}\varphi_r, \Delta\}_i}{\vdash S{:}C(\varphi_1, \ldots, \varphi_r), \Delta}\ (S{:}C)_r$$

Clarifications: A one-sided sequent calculus for a generic finite-valued logic with truth value set $N = \{0, \ldots, n-1\}$ for any $n \geq 2$. Let C be an r-ary propositional connective with matrix $M_C : N^r \to N$ and $S \subseteq N$. We call S a *sign* and—for any propositional formula φ over propositional variables Σ and connectives $C_1, \ldots, C_m$—the expression $S{:}\varphi$ a *signed formula*. Interpretations $I : \Sigma \to N$ are given the usual homomorphic extension to propositional formulas. Sequents have only a succedent which is a finite multiset of signed formulas. The calculus is designed such that $\vdash S{:}\varphi$ is derivable iff $I(\varphi) \in S$ for all I. This generalizes two-valued validity, where $N = \{0, 1\}$ and $S = \{1\}$. The calculus has only two (generic) rules. Let $\{\overline{S}^i\}_i = \{S^i_1, \ldots, S^i_r\}_i$ be a finite family of sets, each over r many signs. A generic rule for a connective C and sign S has $|\{\overline{S}^i\}_i| = m \geq 1$ many premises. Any rule where $\emptyset \neq \{\overline{s} \mid M_C(\overline{s}) \in S\} = \bigcap_i \left(\bigcup_{j=1}^{r} (N \times \cdots \times N \times S^i_j \times N \times \cdots \times N)\right)$ is admissible. (If $\{\overline{s} \mid M_C(\overline{s}) \in S\} = \emptyset$, then no rule is defined.) The intuition: the intersection of all premises must contain those tuples in $\overline{s} \in N^r$ such that the range of M_C is in S. Hence, each $\overline{s}$ must be contained in at least one $N \times \cdots \times N \times S^i_j \times N \times \cdots \times N$. The second rule is a generic axiom that detects tautologies of the form $N{:}\varphi$.

An example is classical binary conjunction $\wedge$, where $n = 2$ and $M_\wedge = \min$. Conjunction on the right in signed logic becomes $\{1\} : \varphi_1 \wedge \varphi_2, \Delta$. We must characterize $\{\overline{s} \mid M_\wedge(\overline{s}) \in \{1\}\} = \{(1, 1)\}$. This is achieved by $(\{1\} \times N) \cap (N \times \{1\})$. Hence, an admissible rule has two premises: $\{1\}{:}\varphi_1, \Delta$ and $\{1\}{:}\varphi_2, \Delta$. Now consider a three-valued logic ($n = 3$) and a unary connective d such that $d(2) = 2$ and $d(s) = 0$ for $s \neq 2$. One admissible rule for $\{0, 2\}{:}d(\varphi)$ has a single premise $N{:}\varphi$. There is no rule for unsatisfiable formulas like $\{1\}{:}d(\varphi)$. One admissible—but not the simplest possible—rule for $\{2\}{:}d(\varphi)$ has the two premises $\{0, 2\}{:}\varphi, \Delta$ and $\{1, 2\}{:}\varphi, \Delta$.

History: The idea to describe generic calculi for finite-valued logics with the help of formulas that are signed with truth value sets appears first in [1] for a tableau system. Predecessors with only single truth values as signs were described by Suchoń, Carnielli, and others. The concept was re-discovered independently by Baaz & Fermueller, Doherty, and Murray & Rosenthal. Details and references are in [2, 3].

Remarks: The calculus is sound and complete for any finite-valued logic. A decision procedure in NP (for fixed n) is obtained in a standard manner. The meta theory of finite truth value sets can be formulated in classical propositional logic and is therefore applicable to virtually any proof system. It also works for certain infinite-valued logics (e.g., Łukasiewicz logic) and can be lifted to certain first-order quantifiers.

[1] Reiner Hähnle. "Towards an Efficient Tableau Proof Procedure for Multiple-Valued Logics". In: *Selected Papers from Computer Science Logic, CSL'90, Heidelberg, Germany*. Ed. by E. Börger, H. Kleine Büning, M. M. Richter, and W. Schönfeld. Vol. 533. LNCS. Springer, 1991, pp. 248–260.

[2] Matthias Baaz, Christian G. Fermüller, and Gernot Salzer. "Automated Deduction for Many-Valued Logics". In: *Handbook of Automated Reasoning*. Ed. by A. Robinson and A. Voronkov. Vol. II. Elsevier, 2001. Chap. 20, pp. 1355–1402.

[3] Reiner Hähnle. "Advanced Many-Valued Logics". In: *Handbook of Philosophical Logic*. Ed. by D. M. Gabbay and F. Guenthner. 2nd. Vol. 2. Kluwer, Dordrecht, Aug. 2001, pp. 297–395.

Entry 30 by: Reiner Hähnle

Superposition (1990/1994)

$$\frac{C \vee \neg u \approx v}{C\sigma} \textit{ Equality Resolution}$$

$$\frac{D \vee u \approx u' \quad C \vee \neg t[v] \approx t'}{(D \vee C \vee \neg t[u'] \approx t')\sigma} \textit{ Negative Superposition}$$

$$\frac{D \vee u \approx u' \quad C \vee t[v] \approx t'}{(D \vee C \vee t[u'] \approx t')\sigma} \textit{ Positive Superposition}$$

plus either

$$\frac{D \vee u \approx u' \quad C \vee s \approx s' \vee t \approx t'[v]}{(D \vee C \vee s \approx s' \vee t \approx t'[u'])\sigma} \textit{ Merging Paramodulation}$$

$$\frac{C \vee s \approx u \vee t \approx v}{(C \vee s \approx u)\sigma} \textit{ Ordered Factoring}$$

or

$$\frac{C \vee v \approx v' \vee u \approx u'}{(C \vee \neg u' \approx v' \vee v \approx v')\sigma} \textit{ Equality Factoring}$$

C, D are (possibly empty) equational clauses, $s, s', t, t', u, u', v, v'$ are terms, u and v (and, for *Ordered Factoring* and *Merging Paramodulation*, s and t) are unifiable with most general unifier σ. In all binary inferences, v is not a variable.

Except for the last but one literal in *Equality Factoring* and *Merging Paramodulation* inferences, every literal involved in some inference is maximal in the respective premise (strictly maximal, if the literal is positive and the inference is binary). In every literal involved in some inference (except *Equality Resolution*), the lhs is strictly maximal. Optionally, ordering restrictions can be overridden by *selection functions*.

For simplicity, it is assumed that the equality predicate $\approx$ is the only predicate symbol in the signature. Non-equational atoms $P(t_1, \ldots, t_n)$ can be encoded as equations $P(t_1, \ldots, t_n) \approx \mathit{true}$.

Entry 31 by: Uwe Waldmann

Clarifications: Superposition is a refutational saturation calculus for first-order clauses (disjunctions of possibly negated atoms) with equality. The inference rules are supplemented by a redundancy criterion that permits to delete clauses that are unnecessary for deriving a contradiction during the saturation, see {32}.

History: The superposition calculus [1, 2] by Bachmair and Ganzinger refines the paramodulation calculus {16}. It uses a syntactic ordering on terms and literals to restrict the paramodulation inference rules in such a way that only (strictly) maximal sides of (strictly) maximal literals participate in inferences, thus combining the restrictions of ordered resolution {15} and unfailing completion {17}. In order to preserve refutational completeness, one new inference rule must be added – either the *Merging Paramodulation* rule [1] or the *Equality Factoring* rule originally due to Nieuwenhuis (which then subsumes *Ordered Factoring*).

The superposition calculus is the basis of most current automated theorem provers for full first-order logic with equality, such as E, SPASS, or Vampire. The calculus and the *model construction* technique used to prove its refutational completeness have been a prototype for numerous refinements, such as constraint superposition {37}, theory superposition {46}, or hierarchic superposition {38}.

Remarks: The superposition calculus is refutationally complete for first-order logic with equality. For certain fragments of first-order logic with equality, there exist strategies that guarantee termination of the calculus, turning superposition into a decision procedure for these fragments.

---

[1] Leo Bachmair and Harald Ganzinger. "Completion of First-Order Clauses with Equality by Strict Superposition (Extended Abstract)". In: *Conditional and Typed Rewriting Systems, 2nd International Workshop*. Ed. by Stéphane Kaplan and Mitsuhiro Okada. LNCS 516. Springer, 1990, pp. 162–180.

[2] Leo Bachmair and Harald Ganzinger. "Rewrite-based Equational Theorem Proving with Selection and Simplification". In: *Journal of Logic and Computation* 4.3 (1994), pp. 217–247.

Saturation With Redundancy (1990)

Primary Rules

$$\frac{N \quad N \models C}{N \cup \{C\}} \textit{ Deduction}$$

$$\frac{N \cup \{C\} \quad C \text{ } \mathcal{R}\text{-redundant w. r. t. } N}{N} \textit{ Deletion}$$

Derived Rules

$$\frac{N \cup \{C\} \quad N \cup \{C\} \models M \quad C \text{ } \mathcal{R}\text{-redundant w. r. t. } N \cup M}{N \cup M} \textit{ Simplification}$$

is a shorthand for

$$\frac{\dfrac{N \cup \{C\} \quad N \cup \{C\} \models M}{N \cup \{C\} \cup M} \textit{ Deduction}^{+} \quad C \text{ } \mathcal{R}\text{-redundant w. r. t. } N \cup M}{N \cup M} \textit{ Deletion}$$

N and M are finite sets of formulas, C is a formula.

Clarifications: This is a meta-inference system for refutational calculi that is parameterized by (1) an entailment relation $\models$, (2) an inference system $\mathcal{I}$ and (3) a redundancy criterion $\mathcal{R}$ for formulas and inferences, such that $\mathcal{I}$-inferences are sound w. r. t. $\models$, and such that $\mathcal{I}$-inferences whose conclusion is contained in N are $\mathcal{R}$-redundant w. r. t. N. Note that the *Deduction* rule is not restricted to adding the conclusions of $\mathcal{I}$-inferences from N; fairness, however, requires that every $\mathcal{I}$-inference from persisting formulas must become $\mathcal{R}$-redundant at some point (for instance, by adding its conclusion).

History: In theorem proving calculi with a redundancy concept, closure under the inference rules can be replaced by a refined notion of saturation that allows to alternate between derivation of new formulas and elimination of irrelevant formulas (e. g., tautologies and subsumed formulas). The system was introduced by Bachmair and Ganzinger [1] for superposition {31}; it can be used for most other superposition-like calculi, such as constraint superposition {37}, superposition modulo theories {46}, or hierarchic superposition {38}, with appropriate choices for $\models$, $\mathcal{I}$, and $\mathcal{R}$.

[1] Leo Bachmair and Harald Ganzinger. "Completion of First-Order Clauses with Equality by Strict Superposition (Extended Abstract)". In: *Conditional and Typed Rewriting Systems, 2nd International Workshop*. Ed. by Stéphane Kaplan and Mitsuhiro Okada. LNCS 516. Springer, 1990, pp. 162–180.

Entry 32 by: Uwe Waldmann

Constructive Classical Logic LC (1991)

$$\frac{}{\vdash \neg P\,;\,P} \qquad \frac{\vdash \Gamma\,;\,P \qquad \vdash \Delta,\neg P\,;\,\Pi}{\vdash \Gamma,\Delta\,;\,\Pi} \qquad \frac{\vdash \Gamma,N\,; \qquad \vdash \Delta,\neg N\,;\,\Pi}{\vdash \Gamma,\Delta\,;\,\Pi}$$

$$\frac{\vdash \Gamma\,;\,\Pi}{\vdash \sigma(\Gamma)\,;\,\Pi} \qquad \frac{\vdash \Gamma\,;\,P}{\vdash \Gamma,P\,;} \qquad \frac{\vdash \Gamma,A,A\,;\,\Pi}{\vdash \Gamma,A\,;\,\Pi} \qquad \frac{\vdash \Gamma\,;\,\Pi}{\vdash \Gamma,A\,;\,\Pi}$$

$$\frac{}{\vdash\,;\,V} \qquad \frac{}{\vdash \Gamma,\neg F\,;\,\Pi}$$

$$\frac{\vdash \Gamma\,;\,P \qquad \vdash \Delta\,;\,Q}{\vdash \Gamma,\Delta\,;\,P\wedge Q} \qquad \frac{\vdash \Gamma,M\,;\,\Pi \qquad \vdash \Delta,N\,;\,\Pi}{\vdash \Gamma,\Delta,M\wedge N\,;\,\Pi}$$

$$\frac{\vdash \Gamma\,;\,P \qquad \vdash \Delta,N\,;}{\vdash \Gamma,\Delta\,;\,P\wedge N} \qquad \frac{\vdash \Gamma,M\,; \qquad \vdash \Delta\,;\,Q}{\vdash \Gamma,\Delta\,;\,M\wedge Q}$$

$$\frac{\vdash \Gamma,A,B\,;\,\Pi}{\vdash \Gamma,A\vee B\,;\,\Pi}\ A\vee B \text{ negative} \qquad \frac{\vdash \Gamma\,;\,P}{\vdash \Gamma\,;\,P\vee Q} \qquad \frac{\vdash \Gamma\,;\,Q}{\vdash \Gamma\,;\,P\vee Q}$$

$$\neg\neg X = X \qquad \neg(A\wedge B) = \neg A\vee\neg B \qquad \neg(A\vee B) = \neg A\wedge\neg B$$

Formulas: $A,B ::= P \mid N$
Positive formulas: $P,Q ::= X \mid V \mid F \mid P\wedge Q \mid P\wedge N \mid M\wedge Q \mid P\vee Q$
Negative formulas: $M,N ::= \neg X \mid \neg V \mid \neg F \mid M\vee N \mid M\vee Q \mid P\vee N \mid M\wedge N$

Γ and Δ are lists of formulas, and Π consists of 0 or 1 positive formula.
σ is a permutation.

Clarifications: Negation is not a connective. It is defined using De Morgan's laws so that $\neg\neg A = A$. There are two atomic formulas for truth (a positive one V and a negative one $\neg F$) and two atomic formulas for falsity (a positive one F and a negative one $\neg V$). Sequents have the shape $\vdash \Gamma\,;\,\Pi$ where Π is called the stoup.

History: **LC** [2] comes from the analysis of classical logic inside the coherent semantics of linear logic [1] together with the use of the focusing property [3].

Remarks: Cut elimination holds. **LK** {6} can be translated into **LC**, but not in a canonical manner. **LC** satisfies constructive properties such as the disjunction property: if $\vdash\,;\,P\vee Q$ is provable then $\vdash\,;\,P$ or $\vdash\,;\,Q$ as well. **LC** admits a denotational semantics through correlation spaces [2] (a variant of coherence spaces [1]).

[1] Jean-Yves Girard. "Linear logic". In: 50 (1987), pp. 1–102.
[2] Jean-Yves Girard. "A new constructive logic: classical logic". In: *Mathematical Structures in Computer Science* 1.3 (1991), pp. 255–296.
[3] Jean-Marc Andreoli. "Logic Programming with Focusing Proofs in Linear Logic". In: *Journal of Logic and Computation* 2.3 (1992), pp. 297–347.

Entry 33 by: Olivier Laurent

Refinement of Structured Specifications (1991)

$$\frac{\text{for all } \varphi \in \Phi,\ SP \vdash \varphi}{SP \vdash \langle Sig[SP], \Phi\rangle}$$

$$\frac{SP \vdash SP_1 \qquad SP \vdash SP_2}{SP \vdash SP_1 \cup SP_2}$$

$$\frac{SP' \textbf{ hide via } \sigma \vdash SP}{SP' \vdash SP \textbf{ with } \sigma}$$

$$\frac{\widehat{SP} \vdash SP'}{SP \vdash SP' \textbf{ hide via } \sigma}\ \sigma\colon SP \to \widehat{SP} \text{ admits model expansion}$$

Clarifications: $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}\colon \mathbf{Sign} \to \mathbf{Set}, \mathbf{Mod}\colon \mathbf{Sign}^{op} \to \mathbf{Cat}, \langle \models_\Sigma \subseteq |\mathbf{Mod}(\Sigma)| \times \mathbf{Sen}(\Sigma)\rangle_{\Sigma \in |\mathbf{Sign}|}\rangle$ is an institution that defines the logical system used for specifications, and SP, SP_1, SP_2, SP' and $\widehat{SP}$ are structured specifications over **INS**. Structured specifications in **INS** are built from basic specifications $\langle \Sigma, \Phi\rangle$ where $\Sigma \in |\mathbf{Sign}|$ and $\Phi \subseteq \mathbf{Sen}(\Sigma)$, the union of Σ-specifications $SP_1 \cup SP_2$, the translation "SP **with** σ" of SP along a signature morphism $\sigma\colon \Sigma \to \Sigma'$, and hiding "$SP$ **hide via** σ" for hiding the symbols in SP not occurring in the image of $\sigma\colon \Sigma' \to \Sigma$. $Sig[SP]$ is the signature of SP and $Mod[SP] \subseteq |\mathbf{Mod}(Sig[SP])|$ is the class of models of SP. A signature morphism $\sigma\colon Sig[SP] \to Sig[SP']$ is a specification morphism $\sigma\colon SP \to SP'$ if for every $M' \in Mod[SP']$, $\mathbf{Mod}(\sigma)(M') \in Mod[SP]$. Then σ admits model expansion if $\mathbf{Mod}(\sigma)\colon Mod[SP'] \to Mod[SP]$ is surjective. The judgment $SP \vdash \varphi$ is entailment for structured specifications which is required to be sound: $SP \vdash \varphi$ implies $M \models_{Sig[SP]} \varphi$ for every $M \in Mod[SP]$.

The judgment $SP \vdash SP'$ is meant to capture that SP refines (or entails) SP', that is, $Sig[SP] = Sig[SP']$ and $Mod[SP] \subseteq Mod[SP']$.

History: The first proof systems for refinement of structured specifications were given by Farrés-Casals [1] and Wirsing [2]. The above presentation can be found in [4], Sect. 9.3.

Remarks: The calculus is sound; it is complete if the underlying entailment system for structured specifications is complete [2, 4]. [3] provides additional rules for observability operators to support refinement by observational abstraction.

---

[1] Jordi Farrés-Casals. "Proving Correctness of Constructor Implementations". In: *Mathematical Foundations of Computer Science 1989, MFCS'89, Porabka-Kozubnik, Poland, August 28 - September 1, 1989, Proceedings*. Vol. 379. Lecture Notes in Computer Science. Springer, 1989, pp. 225–235.

[2] Martin Wirsing. "Structured Specifications: Syntax, Semantics and Proof Calculus". In: *Logic and Algebra of Specification, Proceedings of the NATO Advanced Institute, 1991*. Vol. 94. Springer, 1993.

[3] Rolf Hennicker. *Structured Specifications with Behavioural Operators: Semantics, Proof Methods and Applications*. Habilitation thesis. LMU Munich, 1997.

[4] Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2012.

Entry 34 by: Rolf Hennicker, Donald Sannella, Andrzej Tarlecki, Martin Wirsing

Resolution for Propositional Linear Time Temporal Logic (LTL) (1991/2001)

$$\frac{\Box(\mathbf{start} \rightarrow (C \vee A)) \quad \Box(\mathbf{start} \rightarrow (D \vee \neg A))}{\Box(\mathbf{start} \rightarrow (C \vee D))}\ \textit{(initial)} \qquad \frac{\Box(P \rightarrow \bigcirc(C \vee A)) \quad \Box(Q \rightarrow \bigcirc(D \vee \neg A))}{\Box(P \wedge Q \rightarrow \bigcirc(C \vee D))}\ \textit{(step)}$$

$$\frac{\Box(P \rightarrow \bigcirc\mathbf{false})}{\Box(\mathbf{start} \rightarrow \neg P) \quad \Box(\mathbf{true} \rightarrow \bigcirc\neg P)}\ \textit{(conversion)}$$

$$\frac{\Box(\mathcal{A}_0 \rightarrow \bigcirc\mathcal{B}_0) \quad \cdots \quad \Box(\mathcal{A}_n \rightarrow \bigcirc\mathcal{B}_n) \quad \Box(P \rightarrow \Diamond L)}{\Box(P \rightarrow \left[\bigwedge_{i=0}^{n}(\neg\mathcal{A}_i)\right] \mathcal{W}\, L)}\ \textit{(temporal),}$$

where for all $0 \leq i \leq n$
$\vdash \mathcal{B}_i \rightarrow \neg L$ and $\vdash \mathcal{B}_i \rightarrow \bigvee_{j=0}^{n} \mathcal{A}_j$

Here A is a proposition, L is a literal, C and D are (possibly empty) disjunctions of literals, P and Q are (possibly empty) conjunctions of literals and $\mathcal{A}_i \rightarrow \bigcirc\mathcal{B}_i$ are *merged step clauses* (conjunctions of step clauses), see [4] for details. Derivation terminates if either $\Box(\mathbf{start} \rightarrow \mathbf{false})$ is derived (unsatisfiable) or no new clause can be derived (satisfiable).

Clarifications: The clausal temporal resolution calculus is for formulae of discrete propositional linear-time temporal logic with finite past and infinite future [1, 2]. Any LTL formula is first translated, in a satisfiability preserving way, into the following normal form: $\Box(\mathbf{start} \rightarrow C)$, *an initial clause*, $\Box(P \rightarrow \bigcirc C)$, *a step clause*, and $\Box(P \rightarrow \Diamond L)$, *a sometime clause*. This removes many of the operators, may add new propositional symbols, for example to rename subformulae. The logical constant **start** only holds in the first moment in time. The resolvent of the temporal resolution rule needs further translation into the normal form. In the rules **true** stands for an empty conjunction of literals; **false** stands for an empty disjunction of literals.

History: First proposed in [3] earlier versions of the normal form and the calculus used clauses with past-time formulae on the left-hand sides and present or future formulae on the right-hand side. The above version of the resolution rules are from [4] and the calculus has been implemented in the prover TRP++ [6]. A different presentation is provided in [5] using a *Divided Separated Normal Form*. The calculus has been extended to the monodic fragment of First-Order temporal logic in {57}.

Remarks: Soundness, completeness and termination are shown in [4].

[1] Arthur Prior. *Past, Present, and Future*. Oxford University Press, 1967.

[2] Hans Kamp. "Tense logic and the theory of linear order". PhD thesis. Los Angeles, USA: UCLA, 1968.

[3] Michael Fisher. "A Resolution Method for Temporal Logic". In: *Proc. IJCAI 1991*. Morgan Kaufman, 1991, pp. 99–104.

[4] Michael Fisher, Clare Dixon, and Martin Peim. "Clausal Temporal Resolution". In: *ACM Transactions on Computational Logic* 2.1 (2001), pp. 12–56. DOI: 10.1145/371282.371311.

[5] Anatoli Degtyarev, Michael Fisher, and Boris Konev. "A Simplified Clausal Resolution Procedure for Propositional Linear-Time Temporal Logic". In: *Proc. TABLEAUX 2002*. Vol. 2381. LNCS. Springer, 2002, pp. 85–99. DOI: 10.1007/3-540-45616-3_7.

[6] Ullrich Hustadt and Boris Konev. "TRP++ 2.0: A temporal resolution prover". In: *Proc. CADE-19*. Vol. 2741. LNAI. Springer, 2003, pp. 274–278. DOI: 10.1007/978-3-540-45085-6_21.

Entry 35 by: Clare Dixon, Michael Fisher, Ullrich Hustadt, Boris Konev

Two-sided Linear Sequent Calculus (1992)

$$\frac{}{B \vdash B}\ \mathit{Init} \qquad \frac{\Gamma \vdash B \mid \Delta \quad \Gamma', B \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \mathit{Cut} \qquad \frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}\ \mathbf{1}_L$$

$$\frac{}{\vdash \mathbf{1}}\ \mathbf{1}_R \qquad \frac{}{\bot \vdash}\ \bot_L \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \bot, \Delta}\ \bot_R$$

$$\frac{\Gamma \vdash A \vdash B, \Delta \vdash C}{\Gamma, A \multimap B, \Delta \vdash C}\ \multimap_L \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ \multimap_R \qquad \frac{\Gamma, B, C \vdash \Delta}{\Gamma, B \otimes C \vdash \Delta}\ \otimes_L$$

$$\frac{\Gamma_1 \vdash B, \Delta_1 \quad \Gamma_2 \vdash C, \Delta_2}{\Gamma_1, \Gamma_2 \vdash B \otimes C, \Delta_1, \Delta_2}\ \otimes_R \qquad \frac{\Gamma_1, B \vdash \Delta_1 \quad \Gamma_2, C \vdash \Delta_2}{\Gamma_1, \Gamma_2, B \parr C \vdash \Delta_1, \Delta_2}\ \parr_L \qquad \frac{\Gamma \vdash B, C, \Delta}{\Gamma \vdash B \parr C, \Delta}\ \parr_R$$

$$\frac{}{\Gamma, \mathbf{0} \vdash \Delta}\ \mathbf{0}_L \qquad \frac{}{\Gamma \vdash \top, \Delta}\ \top_R \qquad \frac{\Gamma, B_i \vdash \Delta}{\Gamma, B_1 \,\&\, B_2 \vdash \Delta}\ \&_L\ (i = 1,2)$$

$$\frac{\Gamma \vdash B, \Delta \quad \Gamma \vdash C, \Delta}{\Gamma \vdash B \,\&\, C, \Delta}\ \&_R \qquad \frac{\Gamma, B \vdash \Delta \quad \Gamma, C \vdash \Delta}{\Gamma, B \oplus C \vdash \Delta}\ \oplus_L \qquad \frac{\Gamma \vdash B_i, \Delta}{\Gamma \vdash B_1 \oplus B_2, \Delta}\ \oplus_R\ (i = 1,2)$$

$$\frac{\Gamma, B[t/x] \vdash \Delta}{\Gamma, \forall x.B \vdash \Delta}\ \forall_L \qquad \frac{\Gamma \vdash B[y/x], \Delta}{\Gamma \vdash \forall x.B, \Delta}\ \forall_R \qquad \frac{\Gamma, B[y/x] \vdash \Delta}{\Gamma, \exists x.B \vdash \Delta}\ \exists_L$$

$$\frac{\Gamma \vdash B[t/x], \Delta}{\Gamma \vdash \exists x.B, \Delta}\ \exists_R \qquad \frac{!\Gamma, B \vdash ?\Delta}{!\Gamma, ?B \vdash ?\Delta}\ ?_L \qquad \frac{!\Gamma \vdash B, ?\Delta}{!\Gamma \vdash !B, ?\Delta}\ !_R$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash ?B, \Delta}\ ?_W \qquad \frac{\Gamma \vdash ?B, ?B, \Delta}{\Gamma \vdash ?B, \Delta}\ ?_C \qquad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash ?B, \Delta}\ ?_D$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, !B \vdash \Delta}\ !_W \qquad \frac{\Gamma, !B, !B \vdash \Delta}{\Gamma, !B \vdash \Delta}\ !_C \qquad \frac{\Gamma, B \vdash \Delta}{\Gamma, !B \vdash \Delta}\ !_D$$

Clarifications: This is an alternate formalization of the sequent style formalization of Linear Logic {25}.

History: This formalization first appeared in [1].

---

[1] A. S. Troelstra. *Lectures on Linear Logic*. Vol. 29. CLSI Lecture Notes. Center for the Study of Language and Information, Stanford, 1992.

Entry 36 by: Elaine Pimentel, Harley Eades III

Constraint Superposition (1992/1995)

$$\frac{C \vee \neg u \approx v \;[\![T]\!]}{C \;[\![T \wedge T'']\!]}\ \textit{Equality Resolution} \qquad \frac{C \vee v \approx v' \vee u \approx u' \;[\![T]\!]}{C \vee \neg u' \approx v' \vee u \approx u' \;[\![T \wedge T'']\!]}\ \textit{Equality Factoring}$$

$$\frac{D \vee u \approx u' \;[\![T']\!] \quad C \vee \neg t[v] \approx t' \;[\![T]\!]}{D \vee C \vee \neg t[u'] \approx t' \;[\![T \wedge T' \wedge T'']\!]}\ \textit{Neg. Sup.} \qquad \frac{D \vee u \approx u' \;[\![T']\!] \quad C \vee t[v] \approx t' \;[\![T]\!]}{D \vee C \vee t[u'] \approx t' \;[\![T \wedge T' \wedge T'']\!]}\ \textit{Pos. Sup.}$$

C, D are (possibly empty) equational clauses, T, T', T'' are constraints (i. e., first-order formulas over terms and the binary predicate symbols = and $\succ$), t, t', u, u', v, v' are terms. In binary inferences, v is not a variable. The constraint T'' is the conjunction of the unifiability constraint $u = v$ and the ordering constraints that state that the literals involved in the inference are maximal in their premises (except for the last but one literal in *Equality Factoring* inferences), that positive literals involved in a *(Positive or Negative) Superposition* inference are strictly maximal in the respective premise, and that in every literal involved in the inference (except *Equality Resolution*), the lhs is strictly maximal.

Clarifications: Constraint superposition is a refutational saturation calculus for first-order clauses (disjunctions of possibly negated atoms) with equality (denoted by $\approx$). A constrained clause $C\;[\![T]\!]$ represents those ground instances $C\theta$ for which $T\theta$ evaluates to $true$; the initially given clauses are supposed to have a trivial constraint, that is, $C\;[\![true]\!]$. The inference rules are supplemented by a redundancy criterion that permits to delete constrained clauses that are unnecessary for deriving a contradiction during the saturation (cf. {32}). In particular, every constrained clause with an unsatisfiable constraint is redundant.

History: The idea to use constrained formulas in automated reasoning originated in [1]. There are several reasons to switch from standard superposition {31} to superposition with constrained clauses [2, 3, 5]. First, ordering constraints make it possible to pass on information about the instances for which an inference is actually needed to the derived clauses. Second, working with unifiability constraints rather than computing and applying unifiers avoids future superposition inferences into the substitution part (basic strategy). Finally, in theory calculi, such as [4], unifiability constraints allow to encode a multitude of theory unifiers compactly.

Remarks: This calculus is refutationally complete for first-order logic with equality, provided that the initially given clauses have only trivial constraints (for ordering constraints, this requirement can be relaxed slightly).

[1] Claude Kirchner, Hélène Kirchner, and Michaël Rusinowitch. "Deduction with Symbolic Constraints". In: *Revue d'Intelligence Artificielle* 4.3 (1990), pp. 9–52.

[2] Robert Nieuwenhuis and Albert Rubio. "Basic Superposition is Complete". In: *ESOP'92, 4th European Symposium on Programming*. Ed. by Bernd Krieg-Brückner. LNCS 582. Springer, 1992, pp. 371–389.

[3] Robert Nieuwenhuis and Albert Rubio. "Theorem Proving with Ordering Constrained Clauses". In: *11th International Conference on Automated Deduction*. Ed. by Deepak Kapur. LNAI 607. Springer, 1992, pp. 477–491.

[4] Robert Nieuwenhuis and Albert Rubio. "AC-superposition with constraints: no AC-unifiers needed". In: *Twelfth International Conference on Automated Deduction*. Ed. by Alan Bundy. LNAI 814. Springer, 1994, pp. 545–559.

[5] Robert Nieuwenhuis and Albert Rubio. "Theorem Proving with Ordering and Equality Constrained Clauses". In: *Journal of Symbolic Computation* 19.4 (1995), pp. 321–351.

Entry 37 by: Uwe Waldmann

Hierarchic Superposition (1992/2013)

Abstraction

$$\frac{C[t]}{C[x] \vee \neg x \approx t} \; \textit{Abstraction}$$

applied exhaustively until no literal contains operator symbols from both Σ_{Base} and Σ_{Ext}, followed by saturation under

$$\frac{M \quad M \models_{\text{Base}} \bot}{\bot} \; \textit{Constraint Refutation}$$

and the rules of the standard superposition calculus {31}, where the latter are restricted in such a way that only extension literals participate in inferences and that all unifying substitutions must be simple.
C is an equational clause, t is a term, x is a fresh variable, M is a finite set of clauses over Σ_{Base}.

Clarifications: Hierarchic superposition is a refutational saturation calculus for first-order clauses with equality modulo a base specification (e. g., some kind of arithmetic), for which a decision procedure is available that can be used as a "black-box" in the *Constraint Refutation* rule. The inference rules are supplemented by a redundancy criterion that permits to delete clauses that are unnecessary for deriving a contradiction during the saturation, see {32}.

History: The hierarchic superposition calculus [1, 2] works in the framework of hierarchic specifications consisting of a base part and an extension, where the models of the hierarchic specification are those models of the extension clauses that are conservative extensions of some base model. The calculus is refutationally complete, provided that the set of clauses is sufficiently complete after abstraction and that the base specification is compact. An improved variant of the calculus was given in [3]; this calculus uses a weaker form of abstraction that is guaranteed to preserve sufficient completeness but requires an additional abstraction step after each inference.

---

[1] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. "Theorem Proving for Hierarchic First-Order Theories". In: *Algebraic and Logic Programming, Third International Conference*. Ed. by Giorgio Levi and Hélène Kirchner. LNCS 632. Springer, 1992, pp. 420–434.

[2] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. "Refutational Theorem Proving for Hierarchic First-Order Theories". In: *Applicable Algebra in Engineering, Communication and Computing* 5.3/4 (1994), pp. 193–212.

[3] Peter Baumgartner and Uwe Waldmann. "Hierarchic Superposition With Weak Abstraction". In: *Automated Deduction, CADE-24, 24nd International Conference on Automated Deduction*. Ed. by Maria Paola Bonacina. LNAI 7898. Springer, 2013, pp. 39–57.

Entry 38 by: Uwe Waldmann

Classical Natural Deduction ($\lambda\mu$-calculus) (1992)

Structural subsystem

$$\frac{A^a \in \Gamma}{a : \Gamma \vdash A \mid \Delta}\ Ax$$

$$\frac{c : \Gamma \vdash A^\alpha, \Delta}{\mu\alpha.c : \Gamma \vdash A \mid \Delta}\ Focus \qquad \frac{p : \Gamma \vdash A \mid \Delta \qquad A^\alpha \in \Delta}{[\alpha]p : \Gamma \vdash \Delta}\ Unfocus$$

Introduction rules

$$\frac{p : \Gamma \vdash A_1 \wedge A_2 \mid \Delta}{\pi_1(p) : \Gamma \vdash A_i \mid \Delta}\ \wedge^i_E \qquad \frac{p_1 : \Gamma \vdash A_1 \mid \Delta \quad p_2 : \Gamma \vdash A_2 \mid \Delta}{(p_1, p_2) : \Gamma \vdash A_1 \wedge A_2 \mid \Delta}\ \wedge_I$$

$$\frac{p : \Gamma \vdash A_1 \vee A_2 \mid \Delta \quad p_1 : \Gamma, A_1^{a_1} \vdash C \mid \Delta \quad p_2 : \Gamma, A_2^{a_2} \vdash C \mid \Delta}{\mathsf{case}\ p\ \mathsf{of}\ [a_1 \Rightarrow p_1 \mid a_2 \Rightarrow p_2] : \Gamma \vdash C \mid \Delta}\ \vee_E$$

$$\frac{q : \Gamma \vdash A_i \mid \Delta}{\iota_i(q) : \Gamma \vdash A_1 \vee A_2 \mid \Delta}\ \vee^i_I$$

$$\frac{p : \Gamma \vdash A \to B \mid \Delta \quad q : \Gamma \vdash A \mid \Delta}{p\,q : \Gamma \vdash B \mid \Delta}\ \to_E \qquad \frac{p : \Gamma, A^a \vdash B \mid \Delta}{\lambda a.p : \Gamma \vdash A \to B \mid \Delta}\ \to_I$$

$$\frac{p : \Gamma \vdash \exists x A \mid \Delta \quad q : \Gamma, A[y/x]^a \vdash C \mid \Delta}{\mathsf{dest}\ p\ \mathsf{as}\ (y, a)\ \mathsf{in}\ q : \Gamma \vdash C \mid \Delta}\ \exists_E \qquad \frac{p : \Gamma \vdash A[t/x] \mid \Delta}{(t, p) : \Gamma \vdash \exists x A \mid \Delta}\ \exists_I$$

$$\frac{p : \Gamma \vdash \forall x A \mid \Delta}{p\,t : \Gamma \vdash A[t/x] \mid \Delta}\ \forall_E \qquad \frac{p : \Gamma \vdash A[y/x] \mid \Delta}{\lambda y.p : \Gamma \vdash \forall x A \mid \Delta}\ \forall_I$$

$$\frac{p : \Gamma \vdash \bot \mid \Delta}{\mathtt{efq}\ p : \Gamma \vdash C \mid \Delta}\ \bot_E \qquad \frac{}{\Gamma \vdash () : \top \mid \Delta}\ \top_I$$

Clarifications: There are two kinds of sequents: first $p : \Gamma \vdash A \mid \Delta$ with a distinguished formula on the right for typing the so-called *unnamed* term p, second $c : \Gamma \vdash \Delta$ with no distinguished formula for typing the so-called *named* term c. The syntax of the underlying $\lambda\mu$-calculus is:

$$\begin{array}{lll} c & ::= & [\alpha]p \\ p, q & ::= & a \mid \mu\alpha.c \mid (p,p) \mid \pi_i(p) \mid \iota_i(p) \mid \mathsf{case}\ p\ \mathsf{of}\ [a_1 \Rightarrow p_1 \mid a_2 \Rightarrow p_2] \\ & \mid & \lambda a.p \mid p\,q \mid \lambda x.p \mid p\,t \mid (t,p) \mid \mathsf{dest}\ p\ \mathsf{as}\ (x,a)\ \mathsf{in}\ q \mid () \mid \mathtt{efq}\ p \end{array}$$

The variables used for referring to assumptions in Γ and to conclusions in Δ range over distinct classes (denoted by Latin and Greek letters respectively). In the rules $\exists_E$ (resp. $\forall_I$), y is assumed fresh in Γ, Δ and $\exists x A$ (resp. $\forall x A$).

History: This system, defined in Parigot [4], highlights that classical logic in natural deduction can be obtained from allowing several conclusions with contraction and weakening on the right of the sequent, as in Gentzen's LK. Additionally, the system assigns to this form of classical reasoning a computational content, based on the μ and bracket operator which provides with a fine-grained decomposition of the operators `call-cc` (from Scheme/ML) or $\mathcal{C}$ (from [2]) that were known at this time to provide computational content to classical logic [3], as well as a decomposition of Prawitz's classical elimination rule of negation [1].

Entry 39 by: Hugo Herbelin

The original presentation [4] only contains implication as well as first-order and second-order universal quantification à la Curry (i.e. without leaving trace of the quantification in the proof-term, what corresponds to computationally interpreting quantification as an intersection type). The presentation above has quantification à la Church (i.e. with an explicit trace in the proof term) what makes the calculus compatible with several reduction strategies such as both call-by-name or call-by-value (see e.g. [8]). Variants with multiplicative disjunctions can be found in [7] or [6], or multiplicative conjunctions in [8].

A standard variant originating in [5] uses only one kind of sequents, interpreting $c : \Gamma \vdash \Delta$ as $c : \Gamma \vdash \bot \mid \Delta$ (and hence removing $\bot_E$ and merging the syntactic categories c and p into one). This variant is logically equivalent to the original presentation (in the presence of $\bot$), but not computationally equivalent [9].

---

[1] Dag Prawitz. *Natural Deduction, a Proof-Theoretical Study*. Almqvist and Wiksell, Stockholm, 1965.

[2] Matthias Felleisen, Daniel P. Friedman, Eugene Kohlbecker, and Bruce F. Duba. "Reasoning with continuations". In: *First Symposium on Logic and Computer Science*. 1986, pp. 131–141.

[3] Timothy G. Griffin. "The Formulae-as-Types Notion of Control". In: *Conf. Record 17th Annual ACM Symp. on Principles of Programming Languages, POPL '90, San Francisco, CA, USA, 17-19 Jan 1990*. ACM, 1990, pp. 47–57.

[4] Michel Parigot. "Lambda-mu-calculus: An algorithmic interpretation of classical natural deduction". In: *Logic Programming and Automated Reasoning: International Conference LPAR '92 Proceedings, St. Petersburg, Russia*. Springer-Verlag, 1992, pp. 190–201.

[5] Philippe de Groote. "On the Relation between the lambda-mu Calculus and the Syntactic Theory of Sequential Control". In: *Logic Programming and Automated Reasoning, Proc. of the 5th International Conference, LPAR'94*. Ed. by F. Pfenning. Berlin, Heidelberg: Springer, 1994, pp. 31–43.

[6] David Pym and Eike Ritter. "On the Semantics of Classical Disjunction". In: *Journal of Pure and Applied Algebra* 159 (2001), pp. 315–338.

[7] Peter Selinger. "Control categories and duality: on the categorical semantics of the lambda-mu calculus". In: *Mathematical Structures in Computer Science* 11.2 (2001), pp. 207–260.

[8] Hugo Herbelin. "C'est maintenant qu'on calcule: au cœur de la dualité". Habilitation thesis. University Paris 11, Dec. 2005.

[9] Hugo Herbelin and Alexis Saurin. *λ-calculus and Λ-calculus: a Capital Difference*. Manuscript. 2010.

Logic of Epistemic Inconsistency (1993)

$$\frac{\begin{array}{c}\Pi\\ \alpha\end{array}}{\alpha!}\ !I \qquad \frac{\alpha!}{\alpha}\ !E \qquad \frac{\alpha}{\alpha?}\ ?I \qquad \frac{\alpha? \quad \begin{array}{c}[\alpha]\\ \Pi\\ \beta\end{array}}{\beta}\ ?E \qquad \frac{\begin{array}{c}[\alpha]\\ \vdots\\ \bot\end{array}}{\neg\alpha}\ \neg I$$

$$\frac{A \quad \neg A}{\bot}\ \neg E \qquad \frac{(\neg\alpha)?}{\neg(\alpha?)}\ \neg?I \qquad \frac{\neg(\alpha?)}{(\neg\alpha)?}\ \neg?E \qquad \frac{\begin{array}{c}[\sim\alpha]\\ \vdots\\ \bot\end{array}}{\alpha}\ \bot\sim \qquad \frac{\begin{array}{c}[\neg\alpha]\\ \vdots\\ \bot\end{array}}{\alpha}\ \bot\neg$$

$$\frac{\neg\alpha\vee\neg\beta}{\neg(\alpha\wedge\beta)}\ \neg\wedge I \qquad \frac{\neg(\alpha\vee\beta)}{\neg\alpha\wedge\neg\beta}\ \neg\wedge E \qquad \frac{\neg\alpha\wedge\neg\beta}{\neg(\alpha\vee\beta)}\ \neg\vee I \qquad \frac{\neg(\alpha\wedge\beta)}{\neg\alpha\vee\neg\beta}\ \neg\vee E \qquad \frac{\alpha}{\neg\neg\alpha}\ \neg\neg I$$

$$\frac{\neg\neg\alpha}{\alpha}\ \neg\neg E \qquad \frac{\exists x\neg\alpha}{\neg\forall x\alpha}\ \neg\forall I \qquad \frac{\neg\forall x\alpha}{\exists x\neg\alpha}\ \neg\forall E \qquad \frac{\forall x\neg\alpha}{\neg\exists x\alpha}\ \neg\exists I \qquad \frac{\exists x\neg\alpha}{\neg\forall x\alpha}\ \neg\exists E$$

$$\frac{\alpha\wedge\neg\beta}{\neg(\alpha\to\beta)}\ \neg\to I \qquad \frac{\neg(\alpha\to\beta)}{\alpha\wedge\neg\beta}\ \neg\to E$$

Clarifications: The syntax of LEI is given by the following BNF rule

$$\phi ::= p \in Atomic \mid \bot \mid \phi\wedge\phi \mid \phi\vee\phi \mid \phi\to\phi \mid \neg\phi \mid\sim\phi \mid \exists x\phi \mid \forall x\phi.$$

LEI is intended to deal with the concept of plausibility. The ? and ! post-fixed operators symbolize credulous and skeptical plausibility. The skeptical plausibility can be defined as the dual of ?, that is $\alpha! :=\sim((\sim\alpha)?)$ LEI is a paraconsistent logic. It has two negation symbols $\sim$ (the classical negation) and $\neg$ (the paraconsistent negation). The $\sim$ can be defined as $\sim\alpha := \alpha\to\bot$. The intuitionistic absurd rule $\neg E$ is restricted to ?-free formulas, which are represented by Roman capital letters. Introduction and elimination rules for $\wedge$, $\vee$, $\to$, $\forall$ and $\exists$ are the usual ones. The $?E$ and $!I$ rule has a quite involved restriction for its application. We need the following definitions.

By a *connection* in a deduction Π between two formula occurrences α and β, we understand a sequence $\alpha_1,\ldots,\alpha_n$ of formula occurrences in Π such that $\alpha_1=\alpha, \alpha_n=\beta$, and one of the following conditions holds for each $i\le n$:

1. α_i is not the major premise of an application of $\vee E$, $\exists E$ and $?E$, and α_{i+1} stands immediately below α_i; or vice versa;
2. α_i is a premise of an application of $\to E$, $\neg E$ or $\sim E$, and α_{i+1} is side connected with α_i;
3. α_i is the major premise of an application of $\vee E$, $\exists E$ and $?E$, and α_{i+1} is a hypothesis discharged by this application; or vice versa;
4. α_i is a consequence of an application of $\to I$, $\neg I$, $\sim I$, $\bot\neg$, $\bot\sim$, and α_{i+1} is a hypothesis discharged by this application; or vice versa;

Two formula occurrences α and β are said to be *modally independent* in a deduction Π iff every connection in Π between α and β contains an occurrence of a ?-closed formula.

Entry 40 by: Ana Teresa Martins, Francicleber Ferreira

The $?E$ can only be applied when each occurrence of α is modally independent in Π from β and any (open) assumption of Π. The $!I$ requires that α is modally independent from any (open) assumption of Π.

History: The Logic of Epistemic Inconsistency was proposed by Pequeno and Buchsbaum [1]. The natural deduction system for LEI was created by Martins and Pequeno [2].

Remarks: Completeness and correctness for LEI has been proved in [3]. Normalization for the system above has been proved in [4].

---

[1] Tarcísio Pequeno and Arthur Buchsbaum. "The Logic of Epistemic Inconsistency". In: *2nd International Conference on the Principles of Knowledge Representation and Reasoning*. 1991, pp. 453–60.

[2] Ana Teresa Martins and Tarcísio Pequeno. "Proof-theoretical considerations about the Logic of Epistemic Inconsistency". In: *Logique et Analise* 143-4 (1993), pp. 245–260.

[3] Ana Teresa Martins. "A Syntactical and Semantical Uniform Treatment for the IDL & LEI Nonmonotonic System". Ph.D. thesis. Departamento de Informática, Universidade Federal de Pernambuco, 1997.

[4] Ana Teresa Martins, Lilia Ramalho Martins, and Felipe Ferreira Morais. "Natural Deduction and Weak Normalization for the Paraconsistent Logic of Epistemic Inconsistency". In: *Handbook of Paraconsistency: Studies in Logic and Cognitive Systems*. Ed. by J.-Y. Béziau, W. A Carnielli, and D. Gabbay. Vol. 9. Elsevier/North-Holland, 2007, pp. 355–382.

Typed LF for Type Theories (1994)

$$\frac{}{<>\vdash \mathbf{valid}} \quad \frac{\Gamma\vdash K\,\mathbf{kind} \quad x\notin FV(\Gamma)}{\Gamma,x:K\vdash \mathbf{valid}} \quad \frac{\Gamma,x:K,\Gamma'\vdash \mathbf{valid}}{\Gamma,x:K,\Gamma'\vdash x:K} \quad (1)$$

$$\frac{\Gamma\vdash \mathbf{valid}}{\Gamma\vdash \mathbf{Type}\,\mathbf{kind}} \quad \frac{\Gamma\vdash A:\mathbf{Type}}{\Gamma\vdash El(A)\,\mathbf{kind}} \quad (5)$$

$$\frac{\Gamma\vdash k:K \quad \Gamma\vdash K=K'}{\Gamma\vdash k:K'} \quad \frac{\Gamma\vdash k=k':K \quad \Gamma\vdash K=K'}{\Gamma\vdash k=k':K'} \quad (2)^* \qquad \frac{\Gamma,x:K,\Gamma'\vdash J \quad \Gamma\vdash k:K}{\Gamma,[k/x]\Gamma'\vdash [k/x]J} \quad (3)^{**}$$

$$\frac{\Gamma\vdash K\,\mathbf{kind} \quad \Gamma,x:K\vdash K'\,\mathbf{kind}}{\Gamma\vdash (x:K)K'\,\mathbf{kind}} \quad \frac{\Gamma\vdash K_1=K_2 \quad \Gamma,x:K_1\vdash K_1'=K_2'}{\Gamma\vdash (x:K_1)K_1'=(x:K_2)K_2'}$$

$$\frac{\Gamma,x:K\vdash k:K'}{\Gamma\vdash [x:K]k:(x:K)K'} \quad \frac{\Gamma\vdash K_1=K_2 \quad \Gamma,x:K_1\vdash k_1=k_2:K}{\Gamma\vdash [x:K_1]k_1=[x:K_2]k_2:(x:K_1)K}$$

$$\frac{\Gamma\vdash f:(x:K)K' \quad \Gamma\vdash k:K}{\Gamma\vdash f(k):[k/x]K'} \quad \frac{\Gamma\vdash f=f':(x:K)K' \quad \Gamma\vdash k_1=k_2:K}{\Gamma\vdash f(k_1)=f'(k_2):[k_1/x]K'}$$

$$\frac{\Gamma,x:K\vdash k':K' \quad \Gamma\vdash k:K}{\Gamma\vdash ([x:K]k')(k)=[k/x]k':[k/x]K'} \quad \frac{\Gamma\vdash f:(x:K)K' \quad x\notin FV(f)}{\Gamma\vdash [x:K]f(x)=f:(x:K)K'} \quad (4)$$

Clarifications: We follow [3]. Terms of **LF** are of the forms $\mathbf{Type}, El(A)$, $(x:K)K'$ (dependent product), $[x:K]K'$ (abstraction), $f(k)$, and judgements of the forms $\Gamma\vdash \mathbf{valid}$ (validity of context), $\Gamma\vdash K\mathbf{kind}$, $\Gamma\vdash k:K$, $\Gamma\vdash k=k':K$, $\Gamma\vdash K=K'$. Rule groups: (1) rules for contexts and assumptions; (2)* equality rules (reflexivity, symmetry and transitivity rules are ommitted); (3)** substitution rules (J denotes the right side of any of the five forms of judgement); (4) rules for dependent product kinds; (5) and the kind **Type**.

History: First defined in [3], ch. 9, **LF** is a typed version of Martin-Löf's logical framework [1]. In difference from Edinburgh LF it may be used to specify type theories. *E.g.*, theories specified in **LF** were used as basis of proof-assistants Lego and Plastic. Later the system was extended to include coercive subtyping [4, 5, 6].

Remarks: The proof-theoretical analysis of **LF** above was used in meta-theoretical studies of larger theories defined on its basis, *e.g.*, UTT (Unifying Theory of dependent Types) that includes inductive schemata, second order logic SOL with impredicative type *Prop* and a hierarchy of predicative universes [3]. H. Goguen defined a typed operational semantics for UTT and proved strong normalization theorem [2]. For **LF** with coercive subtyping conservativity results were obtained [4, 5, 6].

[1] B. Nordström, G. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An Introduction*. Oxford, UK: Oxford University Press, 1990.

[2] H. Goguen. "A Typed Operational Semantics for Type Theory". PhD thesis. Univ. of Edinburgh, 1994.

[3] Zhaohui Luo. *Computation and Reasoning. A Type Theory for Computer Science*. Oxford, UK: Clarendon Press, 1994.

[4] Z. Luo. "Coercive subtyping". In: *Journal of Logic and Computation* 9.1 (1999), pp. 105–130.

[5] S. Soloviev and Z. Luo. "Coercion Completion and Conservativity in Coercive Subtyping". In: *Annals of Pure and Applied Logic* 113.1-3 (2002), pp. 297–322.

[6] Z. Luo, S. Soloviev, and T. Xue. "Coercive Subtyping: Theory and Implementation". In: *Information and Computation* 223 (2013), pp. 18–42.

Entry 41 by: Zhaohui Luo, Sergei Soloviev

$\overline{\lambda}$-calculus (1994)

CUT-FREE SYSTEM

$$\frac{}{\Gamma;\cdot : A \vdash \cdot() : A}\ Ax \qquad \frac{\Gamma;\cdot : A \vdash \cdot(l) : C \quad (a : A) \in \Gamma}{\Gamma \vdash a(l) : C}\ Cont$$

$$\frac{\Gamma \vdash p : A \quad \Gamma;\cdot : B \vdash \cdot(l) : C}{\Gamma \mid (p,l) : A \to B \vdash C}\ \to_L \qquad \frac{\Gamma, a : A \vdash p : B}{\Gamma \vdash \lambda a.p : A \to B}\ \to_R$$

CUT RULES

$$\frac{\Gamma \vdash p : A \quad \Gamma;\cdot : A \vdash \cdot(l) : C}{\Gamma \vdash p(l) : C}\ Cut^1_H \qquad \frac{\Gamma;\cdot : A \vdash \cdot(l) : B \quad \Gamma;\cdot : B \vdash \cdot(l') : C}{\Gamma;\cdot : A \vdash \cdot(l@l') : C}\ Cut^2_H$$

$$\frac{\Gamma \vdash p : A \quad \Gamma, a : A, \Gamma' \vdash q : C}{\Gamma, \Gamma' \vdash q[p/a] : C}\ Cut^1_M \qquad \frac{\Gamma \vdash p : A \quad \Gamma, a : A, \Gamma';\cdot : B \vdash \cdot(l) : C}{\Gamma, \Gamma';\cdot : B \vdash \cdot(l[p/a])C}\ Cut^2_M$$

Clarifications: This calculus can be seen as an organization of the rules of Gentzen's intuitionistic sequent calculus in a way such that: there is computational interpretation of proofs as λ-calculus-like terms; there is a simple one-to-one correspondence between cut-free proofs and normal proofs of natural deduction.

The definition of the calculus is based on two kinds of sequents: the sequents $\Gamma \vdash p : A$ have a focus on the right and are annotated by a program p; the sequents $\Gamma;\cdot : A \vdash \cdot(l) : B$ have an extra focussed formula on the left annotated by a placeholder name $\cdot$ while the formula on the right is annotated by a program referring to this placeholder. The syntax of the underlying calculus is:

$$\begin{array}{lll} (l),(l') & ::= & () \mid (p,l) \mid (l@l') \mid (l[p/a]) \\ p,q & ::= & a(l) \mid \lambda a.p \mid p(l) \mid q[p/a] \end{array}$$

with $()$ and (p,l) denoting lists of arguments, $l@l'$ denoting concatenation of lists, $l[p/a]$ and $p[q/a]$ denoting explicit substitution, $x(l)$ and $p(l)$ denoting cut-free and non cut-free application, respectively. The first two items of each entry characterize the syntax of cut-free proofs.

History: The λ-calculus has been designed in [4, 5]. It can be seen as the direct counterpart for sequent calculus of what λ-calculus is for natural deduction, along the lines of the Curry-Howard correspondence between proofs and programs. The idea of focussing a specific formula of the sequent comes from Girard [1] which himself credits it to Andreoli [2] (see also {67}). With proof annotations removed, the calculus can be seen as the intuitionistic fragment LJT of the subsystem LKT of LK [3], with LKT and LKQ representing two dual ways to add asymmetric focus to LK.

Extensions to other connectives than implication can be given. Extensions to classical logic, namely a computational presentation of LKT, can be obtained by adding the μ and bracket operators of $\lambda\mu$-calculus {39} and by considering instead three kinds of sequents, $\Gamma \vdash p : A \mid \Delta$, or $\Gamma;\cdot : A \vdash \cdot(l) : B$, or $c : (\Gamma \vdash \Delta)$ (see [5]). A variant with implicit substitution is possible.

The symmetrization of λ-calculus led to $\mathbf{LK}_{\mu\tilde{\mu}}$ {50}.

---

[1] Jean-Yves Girard. "A new constructive logic: classical logic". In: *Math. Structures in Comp. Science* 1 (1991), pp. 255–296.

Entry 42 by: Hugo Herbelin

[2] Jean-Marc Andreoli. “Logic Programming with Focusing Proofs in Linear Logic”. In: *JLC* 2.3 (1992), pp. 297–347.

[3] V. Danos, J.-B. Joinet, and H. Schellinx. “LKT and LKQ: sequent calculi for second order logic based upon dual linear decompositions of classical implication”. In: *Advances in Linear Logic*. Ed. by J.-Y. Girard, Y. Lafont, and L. Regnier. London Mathematical Society Lecture Note Series 222. Cambridge University Press, 1995, pp. 211–224.

[4] Hugo Herbelin. “A Lambda-Calculus Structure Isomorphic to Gentzen-Style Sequent Calculus Structure”. In: *Computer Science Logic, 8th International Workshop, CSL ’94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*. Ed. by Leszek Pacholski and Jerzy Tiuryn. Vol. 933. LNCS. Springer, 1995, pp. 61–75. ISBN: 3-540-60017-5.

[5] Hugo Herbelin. “Séquents qu’on calcule: de l’interprétation du calcul des séquents comme calcul de λ-termes et comme calcul de stratégies gagnantes”. Ph.D. thesis. University Paris 7, Jan. 1995.

Full Intuitionistic Logic (FIL) (1995)

$$\frac{}{A(n) \Rightarrow A/\{n\}}\ ax \qquad \frac{}{\bot\,(n) \Rightarrow A_1/\{n\},\ldots,A_k/\{n\}}\ \bot\Rightarrow$$

$$\frac{\Gamma_1 \Rightarrow \Delta_1, A/S \quad A(n), \Gamma_1 \Rightarrow \Delta_1}{\Gamma_1, \Gamma_1 \Rightarrow \Delta_1, \Delta_1^*}\ cut \qquad \frac{\Gamma_1, A(m), B(n), \Gamma_1 \Rightarrow \Delta}{\Gamma_1, B(n), A(m), \Gamma_1 \Rightarrow \Delta}\ perm\Rightarrow$$

$$\frac{\Gamma \Rightarrow \Delta_1, A/S_1, B/S_1, \Delta_1}{\Gamma \Rightarrow \Delta_1, B/S_1, A/S_1, \Delta_1}\ \Rightarrow perm \qquad \frac{\Gamma \Rightarrow \Delta}{A(n), \Gamma \Rightarrow \Delta^*}\ weak\Rightarrow$$

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A/\{\}}\ \Rightarrow weak \qquad \frac{\Gamma, A(n), A(m) \Rightarrow \Delta}{\Gamma, A(k) \Rightarrow \Delta^*}\ cont\Rightarrow$$

$$\frac{\Gamma \Rightarrow \Delta, A/S_1, A/S_1}{\Gamma \Rightarrow \Delta, A/S_1 \cup S_1}\ \Rightarrow cont \qquad \frac{\Gamma_1, A(n) \Rightarrow \Delta_1 \quad \Gamma_1, B(m) \Rightarrow \Delta_1}{\Gamma_1, \Gamma_1, (A \vee B)(k) \Rightarrow \Delta_1^*, \Delta_1^*}\ \vee\Rightarrow$$

$$\frac{\Gamma \Rightarrow \Delta, A/S_1, B/S_1}{\Gamma \Rightarrow \Delta, (A \vee B)/S_1 \cup S_1}\ \Rightarrow \vee \qquad \frac{\Gamma, A(n), B(m) \Rightarrow \Delta}{\Gamma, (A \wedge B)(k) \Rightarrow \Delta^*}\ \wedge\Rightarrow$$

$$\frac{\Gamma \Rightarrow \Delta, A/S_1 \quad \Gamma \Rightarrow \Delta, B/S_1}{\Gamma \Rightarrow \Delta, (A \wedge B)/S_1 \cup S_1}\ \Rightarrow \wedge \qquad \frac{\Gamma_1 \Rightarrow \Delta_1, A/S \quad B(n), \Gamma_1 \Rightarrow \Delta_1}{(A \to B)(n), \Gamma_1, \Gamma_1 \Rightarrow \Delta_1, \Delta_1^*}\ \to\Rightarrow$$

$$\frac{\Gamma, A(n) \Rightarrow \Delta, B/S}{\Gamma \Rightarrow \Delta, (A \to B)/S - \{n\}}\ \Rightarrow\to$$

Clarifications: Sequents are of the form $\Gamma \Rightarrow \Delta$ where Γ is a multiset of pairs of formulas and natural number indicies, and Δ is a multiset of pairs of formulas and sets of natural number indicies. The set of natural number indicies for a particular conclusion, formula on the right, indicates which hypotheses the conclusion depends on. This dependency tracking is used to enforce intuitionism in the rule $\Rightarrow\to$. See [2] for more details.

History: The system FIL was announced in the abstract [1] but only published officially ten years later in [2]. The system was conceived after the remark in the paper describing FILL {29} that intuitionism is about proofs that resemble functions, not about a cardinality constraint in the sequent calculus. The system shows we can use a notion of *dependency between formulae* to enforce the constructive character of derivations. This is similar to an impoverished Curry-Howard term assignment.

---

[1] Valeria de Paiva and Luiz C. Pereira. "A New Proof System for Intuitionistic Logic". In: *Bulletin of Symbolic Logic* 1.1 (1995), p. 101.

[2] Valéria de Paiva and Luiz Pereira. "A Short Note on Intuitionistic Propositional Logic with Multiple Conclusions". In: *Manuscrito* 28.2 (2005), pp. 317–329.

Entry 43 by: Harley Eades III, Valeria de Paiva

T[C] (LF with Coercive Subtyping) (1996)

Basic subkinding rule

$$\frac{\Gamma \vdash A <_c B : \mathbf{Type}}{\Gamma \vdash El(A) <_c El(B)}$$

Subkinding for dependent product kinds

$$\frac{\Gamma \vdash K_1' = K_1 \quad \Gamma, x : K_1' \vdash K_2 <_c K_2' \quad \Gamma, x : K_1 \vdash K_2 : \mathbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_{[f:(x:K_1)K_2][x:K_1']c(fx)} (x : K_1')K_2'}$$

$$\frac{\Gamma \vdash K_1' <_c K_1 \quad \Gamma, x : K_1' \vdash [cx/x]K_2 = K_2' \quad \Gamma, x : K_1 \vdash K_2 : \mathbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_{[f:(x:K_1)K_2][x:K_1']f(cx)} (x : K_1')K_2'}$$

$$\frac{\Gamma \vdash K_1' <_{c_1} K_1 \quad \Gamma, x : K_1' \vdash [c_1 x/x]K_2 <_{c_2} K_2' \quad \Gamma, x : K_1 \vdash K_2 : \mathbf{kind}}{\Gamma \vdash (x : K_1)K_2 <_{[f:(x:K_1)K_2][x:K_1']c_2(f(c_1 x))} (x : K_1')K_2'}$$

Coercive application rules

$$\frac{\Gamma \vdash f : (x : K)K' \quad \Gamma \vdash k_0 : K_0 \quad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) : [c(k_0)/x]K'}$$

$$\frac{\Gamma \vdash f(k_0) = f(ck_0) : (x : K)K' \quad \Gamma \vdash k_0 : K_0 \quad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) = f'(k_0') : [c(k_0)/x]K'}$$

Coercive definition rule

$$\frac{\Gamma \vdash f : (x : K)K' \quad \Gamma \vdash k_0 : K_0 \quad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) : [c(k_0)/x]K'}$$

Structural rules

$$\frac{\Gamma \vdash A <_c B \quad \Gamma \vdash A = A' : Type \quad \Gamma \vdash B = B' \quad \Gamma \vdash c = c' : (El(A))El(B)}{\Gamma \vdash A' <_{c'} B'}$$

$$\frac{\Gamma \vdash A <_c A' \quad \Gamma \vdash A' <_{c'} A''}{\Gamma \vdash A_{c' \circ c} A''}$$

$$\frac{\Gamma, x : K, \Gamma' \vdash A <_c B \quad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]A <_{[k/x]c} [k/x]B} \qquad \frac{\Gamma, \Gamma' \vdash A <_c B \quad \Gamma, \Gamma'' \vdash \mathbf{valid}}{\Gamma, \Gamma'', \Gamma' \vdash A <_c B}$$

$$\frac{\Gamma, x : K, \Gamma' \vdash A <_c B \quad \Gamma \vdash K = K'}{\Gamma, x : K', \Gamma' \vdash A <_c B}$$

Clarifications: We follow [5]. In this entry the extensions $T[C]$ of the logical framework **LF** {41} are considered. Here T is a type theory specified in **LF** (formally, an extension of **LF**) and C is a (possibly infinite) set of subtyping judgements of the form $\Gamma \vdash A <_c B : Type$. The set C itself may be generated by some user-defined rules. As coercive definition rule above shows, coercive subtyping is considered as

Entry 44 by: Zhaohui Luo, Sergei Soloviev

an *abbreviation mechanism*, the expressions without coercions are considered as "abbreviations" of the expressions where coercions are inserted. For corcive subtyping as an abbreviation mechanism, one of central questions is the concervativity of the extension $T[C]$ over T.

The system $T[C]$ is built by "layers" and in this sense may be considered as *hybrid*. Above the rules (except structural rules) of the *subkinding* level are given. The structure (and rules) of the subtyping level, as well as its connection with the subkinding level, are explained below.

First the intermediate system $T[C]_0$ is defined. The syntax of $T[C]_0$ is the same as the syntax of T (*i.e.*, type theory specified in **LF**). The rule

$$\frac{\Gamma \vdash A <_c B : Type \in C}{\Gamma \vdash A <_c B : Type}$$

is added, and the structural subtyping rules given below. They state that the subtyping relation $<$ (annotated by coercion terms c) is congruent, transitive, and closed under substitution, and satisfies the rules of weakening and contextual equality. Similar structural rules are included in the subkinding level above.

Main requirement to the set C (expressed in terms of $T[C]_0$) is *coherence*:

- If $\Gamma \vdash A <_c B : Type$ then $\Gamma \vdash A : Type$, $\Gamma \vdash B : Type$ and $\Gamma \vdash c : (El(A))El(B)$.
- $\Gamma \not\vdash A <_c A : Type$ for any Γ, A, c.
- If $\Gamma \vdash A <_c B : Type$ and $\Gamma \vdash A <_{c'} B : Type$ then $\Gamma \vdash c = c' : (El(A))El(B)$.

History: Coercive subtiping as an abbreviation mechanism was introduced in a conference paper [1]. It was described for type theories specified in Z. Luo's typed **LF** (extensions of **LF**) {41}, but the idea itself is much more general and may apply to other type theories. The approach was further developed in [2, 3, 4, 5].

Remarks: The main theorem (justifying the view of coercive subtyping as an abbreviation mechanism) is the conservativity of $T[C]$ w.r.t. the type theory T.

---

[1] Zhaohui Luo. "Coercive subtyping in type theory". In: *LNCS* (1997). Proc. of CSL'96.

[2] A. Jones, Z. Luo, and S. Soloviev. "Some proof-theoretic and algorithmic aspects of coercive subtyping". In: *LNCS* (1998). Proc. of the Annual Conf on Types and Proofs (TYPES'96).

[3] Z. Luo. "Coercive subtyping". In: 9.1 (1999), pp. 105–130.

[4] S. Soloviev and Z. Luo. "Coercion Completion and Conservativity in Coercive Subtyping". In: 113.1-3 (2002), pp. 297–322.

[5] Z. Luo, S. Soloviev, and T. Xue. "Coercive Subtyping: Theory and Implementation". In: 223 (2013), pp. 18–42.

Sequent Calculus G3c (1996)

$$\frac{}{P,\Gamma \vdash \Delta,P}\ \mathrm{Ax} \qquad \frac{}{\bot,\Gamma \vdash \Delta}\ \mathrm{L}\bot$$

$$\frac{A,B,\Gamma \vdash \Delta}{A \wedge B,\Gamma \vdash \Delta}\ \mathrm{L}\wedge \qquad \frac{\Gamma \vdash \Delta,A \quad \Gamma \vdash \Delta,B}{\Gamma \vdash \Delta,A \wedge B}\ \mathrm{R}\wedge$$

$$\frac{A,\Gamma \vdash \Delta \quad B,\Gamma \vdash \Delta}{A \vee B,\Gamma \vdash \Delta}\ \mathrm{L}\vee \qquad \frac{\Gamma \vdash \Delta,A,B}{\Gamma \vdash \Delta,A \vee B}\ \mathrm{R}\vee$$

$$\frac{\Gamma \vdash \Delta,A \quad B,\Gamma \vdash \Delta}{A \to B,\Gamma \vdash \Delta}\ \mathrm{L}\to \qquad \frac{A,\Gamma \vdash \Delta,B}{\Gamma \vdash \Delta,A \to B}\ \mathrm{R}\to$$

$$\frac{\forall xA,A[x/t],\Gamma \vdash \Delta}{\forall xA,\Gamma \vdash \Delta}\ \mathrm{L}\forall \qquad \frac{\Gamma \vdash \Delta,A[x/y]}{\Gamma \vdash \Delta,\forall xA}\ \mathrm{R}\forall$$

$$\frac{A[x/y],\Gamma \vdash \Delta}{\exists xA,\Gamma \vdash \Delta}\ \mathrm{L}\exists \qquad \frac{\Gamma \vdash \Delta,A[x/t],\exists xA}{\Gamma \vdash \Delta,\exists xA}\ \mathrm{R}\exists$$

P should be atomic in Ax and y should not be free in the conclusion of R$\forall$ and L$\exists$

Clarifications: Sequents are based on multisets. A formula $A[x/t]$ is the result of uniformly substituting the term t for the variable x in A, renaming bound variables to prevent clashes with the variables in t.

Remarks: **G3c** is sound and complete w.r.t. classical first-order logic. Weakening and contraction are depth-preserving admissible and all rules are depth-preserving invertible.

---

[1] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. 2nd ed. Vol. 43. Cambridge Tracts In Theoretical Computer Science. Cambridge University Press, 2000.

Entry 45 by: Björn Lellmann

Cancellative Superposition (1996)

Cancellative rules (for simplicity, the ground versions are given; the non-ground rules are obtained by lifting):

$$\frac{C \vee \neg t \approx t}{C} \textit{ Equality Resolution}$$

$$\frac{C \vee [\neg] nu + t \approx mu + s}{C \vee [\neg](n-m)u + t \approx s} \textit{ Cancellation}$$

$$\frac{D \vee mu + s \approx s' \quad C \vee [\neg] nu + t \approx t'}{D \vee C \vee [\neg](n-m)u + t + s' \approx t' + s} \textit{ Cancellative Superposition}$$

$$\frac{C \vee nu + s \approx s' \vee nu + t \approx t'}{C \vee \neg s + t' \approx s' + t \vee nu + t \approx t'} \textit{ Cancellative Equality Factoring}$$

plus, if there are any non-constant function symbols besides +, the rules of the standard superposition calculus {31} and

$$\frac{C \vee [\neg] w[nu + t] \approx w'}{C \vee \neg x \approx nu + t \vee [\neg] w[x] \approx w'} \textit{ Abstraction}$$

C, D are (possibly empty) equational clauses, s, s', t, t' are terms, u is an atomic term, n, m are positive integers. Every literal involved in some inference is maximal in the respective premise (except for the last but one literal in *Equality Factoring* inferences). A positive literal involved in a *Superposition* inference is strictly maximal in the respective clause. In every literal involved in a cancellative inference (except *Equality Resolution*), the term u is the maximal atomic term.

Clarifications: Cancellative superposition is a refutational saturation calculus for first-order clauses containing the axioms of cancellative abelian monoids or abelian groups. The inference rules are supplemented by a redundancy criterion that permits to delete clauses that are unnecessary for deriving a contradiction during the saturation, see {32}.

History: As a naïve handling of axioms like commutativity or associativity in an automated theorem prover leads to an explosion of the search space, there has been a lot of interest in incorporating specialized techniques into general proof systems to work efficiently within standard algebraic theories. The cancellative superposition calculus [9] shown above is one example of a saturation calculus with a built-in algebraic theory. By using dedicated inference rules, explicit inferences with the theory axioms become superfluous; moreover variable elimination techniques and strengthened ordering restrictions and redundancy criteria lead to a significant reduction of the search space. The cancellative superposition calculus is refutationally complete for first-order logic modulo cancellative abelian monoids.

Other examples for "white-box" theory integration include calculi for dealing with associativity and commutativity [1, 3, 8, 5], superposition modulo abelian groups [12], chaining calculi [2, 4, 7, 6], or superposition modulo divisible torsion-free abelian groups or ordered divisible abelian groups [11, 10].

[1] Gordon D. Plotkin. "Building-in Equational Theories". In: *Machine Intelligence 7*. Ed. by Bernard Meltzer and Donald Michie. American Elsevier, 1972. Chap. 4, pp. 73–90.

[2] James R. Slagle. "Automatic Theorem Proving with Built-In Theories Including Equality, Partial Ordering, and Sets". In: *Journal of the ACM* 19.1 (1972), pp. 120–135.

Entry 46 by: Uwe Waldmann

[3] James R. Slagle. “Automated Theorem-Proving for Theories with Simplifiers, Commutativity, and Associativity”. In: *Journal of the ACM* 21.4 (1974), pp. 622–642.

[4] Larry M. Hines. “Completeness of a Prover for Dense Linear Logics”. In: *Journal of Automated Reasoning* 8 (1992), pp. 45–75.

[5] Leo Bachmair and Harald Ganzinger. “Associative-Commutative Superposition”. In: *Conditional and Typed Rewriting Systems, 4th International Workshop, CTRS-94*. Ed. by Nachum Dershowitz and Naomi Lindenstrauss. LNCS 968. Springer, 1994, pp. 1–14.

[6] Leo Bachmair and Harald Ganzinger. “Ordered Chaining for Total Orderings”. In: *Twelfth International Conference on Automated Deduction*. Ed. by Alan Bundy. LNAI 814. Springer, 1994, pp. 435–450.

[7] Leo Bachmair and Harald Ganzinger. “Rewrite Techniques for Transitive Relations”. In: *Ninth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1994, pp. 384–393.

[8] Michaël Rusinowitch and Laurent Vigneron. “Automated Deduction with Associative-Commutative Operators”. In: *Applicable Algebra in Engineering, Communication and Computing* 6.1 (1995), pp. 23–56.

[9] Harald Ganzinger and Uwe Waldmann. “Theorem Proving in Cancellative Abelian Monoids (Extended Abstract)”. In: *Automated Deduction – CADE-13, 13th International Conference on Automated Deduction*. Ed. by Michael A. McRobbie and John K. Slaney. LNAI 1104. Springer, 1996, pp. 388–402.

[10] Uwe Waldmann. “Superposition and Chaining for Totally Ordered Divisible Abelian Groups (Extended Abstract)”. In: *Automated Reasoning, First International Joint Conference, IJCAR 2001*. Ed. by Rajeev Goré, Alexander Leitsch, and Tobias Nipkow. LNAI 2083. Springer, 2001, pp. 226–241.

[11] Uwe Waldmann. “Cancellative Abelian Monoids and Related Structures in Refutational Theorem Proving (Part I & II)”. In: *Journal of Symbolic Computation* 33.6 (2002), pp. 777–861.

[12] Guillem Godoy and Robert Nieuwenhuis. “Superposition with completely built-in Abelian groups”. In: *Journal of Symbolic Computation* 37.1 (2004), pp. 1–33.

Graph-based tableaux for modal logics (1997)

BOOLEAN RULES

$$\frac{\Gamma, A, \neg A \bullet}{\Gamma, A, \neg A, \bot \bullet}\ (\bot) \qquad \frac{\Gamma, A \wedge B \bullet}{\Gamma, A \wedge B, A, B \bullet}\ (\wedge) \qquad \frac{\Gamma, A_1 \vee A_2 \bullet}{\Gamma, A_1 \vee A_2, A_i \bullet}\ (\vee)$$

DIAMOND RULE

$$\frac{\Gamma, \Diamond A \bullet}{\Gamma, \Diamond A \bullet \longrightarrow \bullet A}\ (\Diamond)$$

PROPAGATION RULES

$$\frac{\Gamma, \Box A \bullet \longrightarrow \bullet \Delta}{\Gamma, \Box A \bullet \longrightarrow \bullet \Delta, A}\ (K)$$

$$\frac{\Gamma, \Box A \bullet}{\Gamma, \Box A, A \bullet}\ (T) \qquad \frac{\Gamma, \Box A \bullet \longrightarrow \bullet \Delta}{\Gamma, \Box A \bullet \longrightarrow \bullet \Delta, \Box A}\ (4) \qquad \frac{\Gamma \bullet \longrightarrow \bullet \Delta, \Box A}{\Gamma, A \bullet \longrightarrow \bullet \Delta, \Box A}\ (B)$$

$$\frac{\Gamma \bullet \begin{array}{l} \nearrow \bullet \Delta_1, \Box A \\ \searrow \bullet \Delta_2 \end{array}}{\Gamma \bullet \begin{array}{l} \nearrow \bullet \Delta_1, \Box A \\ \searrow \bullet \Delta_2, \Box A \end{array}}\ (5_{\rightarrow}) \qquad \frac{\Gamma \bullet \longrightarrow \bullet \Delta, \Box A}{\Gamma, \Box A \bullet \longrightarrow \bullet \Delta, \Box A}\ (5_{\uparrow}) \qquad \frac{\begin{array}{c} \Delta_1, \Box A \\ \Gamma \bullet \longrightarrow \bullet \end{array} \longrightarrow \bullet \Delta_2}{\begin{array}{c} \Delta_1, \Box A \\ \Gamma \bullet \longrightarrow \bullet \end{array} \longrightarrow \bullet \Delta_2, \Box A}\ (5_{\downarrow})$$

STRUCTURAL RULES

$$\frac{\Gamma \bullet}{\Gamma \bullet \longrightarrow \bullet \emptyset}\ (D) \qquad \frac{\Gamma \bullet \longrightarrow \bullet \Delta}{\Gamma \bullet \begin{array}{l} \nearrow \\ \searrow \end{array} \begin{array}{c} \bullet \Delta \\ \bullet \\ \emptyset \end{array}}\ (De)$$

$$\frac{\Gamma \bullet \begin{array}{l} \nearrow \bullet \Delta_1 \\ \searrow \bullet \Delta_2 \end{array}}{\Gamma \bullet \begin{array}{l} \nearrow \bullet \\ \searrow \bullet \end{array} \begin{array}{l} \Delta_1 \\ \\ \Delta_2 \end{array} \bullet \emptyset}\ (C_0) \qquad \frac{\Gamma \bullet \longrightarrow \bullet \Delta}{\Gamma \bullet \longrightarrow \begin{array}{c} \bullet \\ \Delta \end{array} \longrightarrow \bullet \emptyset}\ (C_1)$$

Clarifications: The method constructs a collection of rooted directed acyclic graphs with vertices labeled with sets of formulas. $\Gamma \bullet$ denotes a vertex labeled with Γ. To each branch of the tableau corresponds a graph. The only branching rule is $(\vee)$, for which i must be choosen among $\{1,2\}$. A branch is closed if the corresponding graph contains a vertex of the form $\Gamma, \bot \bullet$. For modal logic K, only rules $(\bot)$, $(\wedge)$, $(\vee)$, $(\Diamond)$ and (K) are used. To each additional axiom T, 4, B, 5, D, De and C corresponds a set of rules to add, as detailed in Table 47.1.

Entry 47 by: Joseph Boudou, Olivier Gasquet

Axiom	Model's property	Rules
$T = \Box A \to A$	reflexivity	(T)
$4 = \Box A \to \Box\Box A$	transitivity	(4)
$B = \Diamond\Box A \to A$	symmetry	(B)
$5 = \Diamond\Box A \to \Box A$	euclideanity	$(5_{\to})$, $(5_{\uparrow})$, $(5_{\downarrow})$
$D = \Box A \to \Diamond A$	seriality	(D)
$De = \Diamond A \to \Diamond\Diamond A$	density	(De)
$C = \Diamond\Box A \to \Box\Diamond A$	confluence	(C_0), (C_1)

Table 47.1 Correspondences between modal axioms and graph-based tableaux rules.

History: Tableaux methods for modal logics have a long history started by Kripke [1]. The present method, introduced in [2] and extended in [3], distinguish itself by its ability to deal with properties like confluence or density. Moreover, it can be easily adapted to multimodal logics. The method has been enhanced and implemented in the LoTREC prover [5].

Remarks: The method is sound and complete for any combination of axioms. Termination is more problematic and has been investigated in [3, 4, 5].

---

[1] Saul A. Kripke. "A completeness theorem in modal logic". In: *Journal of Symbolic Logic* 24.1 (1959), pp. 1–14.

[2] Marcos A. Castilho, Luis Fariñas del Cerro, Olivier Gasquet, and Andreas Herzig. "Modal Tableaux with Propagation Rules and Structural Rules". In: *Fundam. Inform.* 32.3-4 (1997), pp. 281–297.

[3] Luis Fariñas del Cerro and Olivier Gasquet. "General Framework for Pattern-Driven Modal Tableaux". In: *Logic Journal of the IGPL* 10.1 (2002), pp. 51–83.

[4] Olivier Gasquet, Andreas Herzig, and Mohamad Sahade. "Terminating modal tableaux with simple completeness proof". In: *Advances in Modal Logic*. College Publications, 2006, pp. 167–186.

[5] Olivier Gasquet, Andreas Herzig, Bilal Said, and François Schwarzentruber. *Kripke's Worlds - An Introduction to Modal Logics via Tableaux*. Studies in Universal Logic. Birkhäuser, 2014.

Synthetic Tableaux (2000)

The synthesizing rules are:

$$\frac{\neg A}{A \to B}\ \mathbf{r}^{\mathbf{1}}_{\to} \qquad \frac{B}{A \to B}\ \mathbf{r}^{\mathbf{2}}_{\to} \qquad \frac{\begin{array}{c}A\\ \neg B\end{array}}{\neg(A \to B)}\ \mathbf{r}^{\mathbf{3}}_{\to}$$

$$\frac{A}{A \vee B}\ \mathbf{r}^{\mathbf{1}}_{\vee} \qquad \frac{B}{A \vee B}\ \mathbf{r}^{\mathbf{2}}_{\vee} \qquad \frac{\begin{array}{c}\neg A\\ \neg B\end{array}}{\neg(A \vee B)}\ \mathbf{r}^{\mathbf{3}}_{\vee}$$

$$\frac{\neg A}{\neg(A \wedge B)}\ \mathbf{r}^{\mathbf{1}}_{\wedge} \qquad \frac{\neg B}{\neg(A \wedge B)}\ \mathbf{r}^{\mathbf{2}}_{\wedge} \qquad \frac{\begin{array}{c}A\\ B\end{array}}{A \wedge B}\ \mathbf{r}^{\mathbf{3}}_{\wedge} \qquad \frac{A}{\neg\neg A}\ \mathbf{r}_{\neg}$$

The premises of rules $\mathbf{r}^{\mathbf{3}}_{\to}$, $\mathbf{r}^{\mathbf{3}}_{\vee}$, $\mathbf{r}^{\mathbf{3}}_{\wedge}$ may occur in any order.

The branching rule:

$$\overbrace{p_i \quad \neg p_i}$$

Clarifications: A Synthetic Tableau for a formula A is a finite tree with the following properties: the tree is generated by the above rules (the root is empty), each formula labelling a node of the tree is a subformula of A or the negation of a subformula of A, each leaf is labelled with A or $\neg A$. The tableau is a proof of A if each leaf is labelled with A.

History: The method has been first presented in [1], [2], [4]. In [4], [3] and [5] it is also presented for some extensional many-valued logics and for some paraconsistent logics.

Remarks: The method is sound and complete with respect to Classical Propositional Logic and constitutes a decision procedure for CPL. The same holds with respect to the non-classical logics for which the method has been described, see [4], [3], [5].

[1] Mariusz Urbański. "Remarks on Synthetic Tableaux for Classical Propositional Calculus". In: *Bulletin of the Section of Logic* 30.4 (2001), pp. 194–204.

[2] Mariusz Urbański. "Synthetic Tableaux and Erotetic Search Scenarios: Extension and Extraction". In: *Logique et Analyse* 173–175 (2001), pp. 69–91.

[3] Mariusz Urbański. "Synthetic Tableaux for Łukasiewicz's Calculus Ł3". In: *Logique et Analyse* 177-178 (2002), pp. 155–173.

[4] Mariusz Urbański. *Tabele syntetyczne a logika pytań (Synthetic Tableaux and the Logic of Questions)*. Lublin: Wydawnictwo UMCS, 2002.

[5] Mariusz Urbański. "How to Synthesize a Paraconsistent Negation. The Case of CLuN". In: *Logique et Analyse* 185-188 (2004), pp. 319–333.

Entry 48 by: Dorota Leszczyńska-Jasion

Polarized Linear Sequent Calculus LLP (2000)

$$\frac{}{\vdash P^\perp, P} \qquad \frac{\vdash \Gamma, P \qquad \vdash \Delta, P^\perp, \Pi}{\vdash \Gamma, \Delta, \Pi} \qquad \frac{\vdash \Gamma, \Pi}{\vdash \sigma(\Gamma), \Pi}$$

$$\frac{\vdash \Gamma, P \qquad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q} \qquad \frac{\vdash \Gamma, N, M, \Pi}{\vdash \Gamma, N \parr M, \Pi} \qquad \frac{}{\vdash 1} \qquad \frac{\vdash \Gamma, \Pi}{\vdash \Gamma, \bot, \Pi}$$

$$\frac{\vdash \Gamma, P}{\vdash \Gamma, P \oplus Q} \qquad \frac{\vdash \Gamma, Q}{\vdash \Gamma, P \oplus Q} \qquad \frac{\vdash \Gamma, M, \Pi \qquad \vdash \Gamma, N, \Pi}{\vdash \Gamma, M \,\&\, N, \Pi} \qquad \frac{}{\vdash \Gamma, \top, \Pi}$$

$$\frac{\vdash \Gamma, P}{\vdash \Gamma, ?P} \qquad \frac{\vdash \Gamma, N}{\vdash \Gamma, !N} \qquad \frac{\vdash \Gamma, N, N, \Pi}{\vdash \Gamma, N, \Pi} \qquad \frac{\vdash \Gamma, \Pi}{\vdash \Gamma, N, \Pi}$$

$$\begin{array}{cccc} & (P \otimes Q)^\perp = P^\perp \parr Q^\perp & & 1^\perp = \bot \\ (!N)^\perp = ?(N^\perp) & (P \oplus Q)^\perp = P^\perp \,\&\, Q^\perp & & 0^\perp = \top \\ (X^\perp)^\perp = X & (N \parr M)^\perp = N^\perp \otimes M^\perp & & \bot^\perp = 1 \\ (?P)^\perp = !(P^\perp) & (N \,\&\, M)^\perp = N^\perp \oplus M^\perp & & \top^\perp = 0 \end{array}$$

Positive formulas: $P, Q ::= X \mid P \otimes Q \mid 1 \mid P \oplus Q \mid 0 \mid !N$
Negative formulas: $N, M ::= X^\perp \mid N \parr M \mid \bot \mid N \,\&\, M \mid \top \mid ?P$

Γ and Δ are lists of negative formulas.
Π consists of 0 or 1 positive formula.
σ is a permutation.

Clarifications: Negation is not a connective. It is defined using De Morgan's laws so that $(A^\perp)^\perp = A$. Negative connectives which turn negative formulas into negative formulas ($\parr$, $\bot$, & and $\top$) are the reversible connectives of **LL** {25}. Their dual, the positive connectives ($\otimes$, 1, $\oplus$, 0) have the focusing property [1], related here with the "at most one positive formula" property of sequents.

History: **LLP** [2] comes from the natural embedding of Girard's **LC** {33} into linear logic {25}. It is obtained by restricting **LL** to polarized formulas and then by generalizing the structural rules (contraction, weakening and context of promotion) to arbitrary negative formulas, not only those starting with a ?-connective.

Remarks: Cut elimination holds. In the categorical models of **LLP**, positive formulas are interpreted as $\otimes$-comonoids while negative formulas are interpreted as $\parr$-monoids.

[1] Jean-Marc Andreoli. "Logic Programming with Focusing Proofs in Linear Logic". In: 2.3 (1992), pp. 297–347.

[2] Olivier Laurent. "Étude de la polarisation en logique". Thèse de Doctorat. Université Aix-Marseille II, Mar. 2002.

Entry 49 by: Olivier Laurent

$\mathbf{LK}_{\mu\tilde{\mu}}$ **(2000)**

STRUCTURAL SUBSYSTEM

$$\frac{(a:A)\in\Gamma}{\Gamma\vdash a:A\mid\Delta}\ Ax_R \qquad \frac{\Gamma\vdash v:A\mid\Delta \quad \Gamma\mid e:A\vdash\Delta}{\langle v|e\rangle:(\Gamma\vdash\Delta)}\ Cut \qquad \frac{(\alpha:A)\in\Delta}{\Gamma\mid\alpha:A\vdash\Delta}\ Ax_L$$

$$\frac{c:(\Gamma,a:A\vdash\Delta)}{\Gamma\mid\tilde{\mu}a.c:A\vdash\Delta}\ Focus_L \qquad \frac{c:(\Gamma\vdash\alpha:A,\Delta)}{\Gamma\vdash\mu\alpha.c:A\mid\Delta}\ Focus_R$$

INTRODUCTION RULES

$$\frac{\Gamma\mid e:A_i\vdash\Delta}{\Gamma\mid\pi_i\cdot e:A_1\wedge A_2\vdash\Delta}\ \wedge^i_L \qquad \frac{\Gamma\vdash v_1:A_1\mid\Delta \quad \Gamma\vdash v_2:A_2\mid\Delta}{\Gamma\vdash(v_1,v_2):A_1\wedge A_2\mid\Delta}\ \wedge_R$$

$$\frac{\Gamma\mid e_1:A_1\vdash\Delta \quad \Gamma\mid e_2:A_2\vdash\Delta}{\Gamma\mid[e_1,e_2]:A_1\vee A_2\vdash\Delta}\ \vee_L \qquad \frac{\Gamma\vdash v:A_i\mid\Delta}{\Gamma\vdash\iota_i(v):A_1\vee A_2\mid\Delta}\ \vee^i_R$$

$$\frac{\Gamma\vdash v:A\mid\Delta \quad \Gamma\mid e:B\vdash\Delta}{\Gamma\mid v\cdot e:A\to B\vdash\Delta}\ \to_L \qquad \frac{\Gamma,a:A\vdash v:B\mid\Delta}{\Gamma\vdash\lambda a.v:A\to B\mid\Delta}\ \to_R$$

$$\frac{\Gamma\mid e:A[y]\vdash\Delta}{\Gamma\mid\tilde{\lambda}x.e:\exists xA[x]\vdash\Delta}\ \exists_L \qquad \frac{\Gamma\vdash v:A[t]\mid\Delta}{\Gamma\vdash t\cdot v:\exists xA[x]\mid\Delta}\ \exists_R$$

$$\frac{\Gamma\mid e:A[t]\vdash\Delta}{\Gamma\mid t\cdot e:\forall xA[x]\vdash\Delta}\ \forall_L \qquad \frac{\Gamma\vdash v:A[y]\mid\Delta}{\Gamma\vdash\lambda x.v:\forall xA[x]\mid\Delta}\ \forall_R$$

$$\frac{}{\Gamma\mid[\,]:\bot\vdash\Delta}\ \bot_L \qquad \frac{}{\Gamma\vdash():\top\mid\Delta}\ \top_R$$

Clarifications: There are three kinds of sequents: first $\Gamma\vdash v:A\mid\Delta$ with a distinguished formula on the right for typing the program v, second $\Gamma\mid e:A\vdash\Delta$ with a distinguished formula on the left for typing the evaluation context e, and finally $c:(\Gamma\vdash\Delta)$ with no distinguished formula for typing command c, i.e. the interaction of a program within an evaluation context. The typing contexts Γ and Δ are lists of named formulas so that a non-ambiguous correspondence with λ-calculus is possible (if it were sets or multisets, there were e.g. no way to distinguish the two distinct proofs of $x:A,x:A\vdash x:A\mid$). Weakening rules are implemented implicitly at the level of axioms. Contraction rules are derived, using a cut against an axiom. No exchange rule is needed. Not all cuts are eliminable: only those not involving an axiom rule are. Negation $\neg A$ can be defined as $A\to\bot$. In the rules $\exists_E$ and $\forall_R$, y is assumed fresh in Γ, Δ and $A[x]$. The syntax of the underlying λ-calculus is:

$$\begin{aligned} c &::= \langle v|e\rangle \\ e &::= \alpha \mid \tilde{\mu}a.c \mid \pi_i\cdot e \mid [e,e] \mid v\cdot e \mid (t,e) \mid \tilde{\lambda}x.e \mid [] \\ v &::= a \mid \mu\alpha.c \mid (v,v) \mid \iota_i(v) \mid \lambda a.v \mid \lambda x.v \mid (t,v) \mid () \end{aligned}$$

History: The purpose of this system is to provide with a λ-calculus-style computational meaning to Gentzen's LK {6} and to highlight how the symmetries of sequent calculus show computationally. Seeing the rules as typing rules, the left/right symmetry is a symmetry between programs and their evaluation contexts. At the level of cut elimination, giving priority to the left-hand side relates to call-by-name evaluation while giving priority to the right-hand side relates to call-by-value evaluation [1]. Thanks to the presence of two dual axiom

Entry 50 by: Hugo Herbelin

rules and implicit contraction rules, the system supports a tree-like sequent-free presentation like originally presented by Gentzen for natural deduction [5] (see {62}). The system can be seen as a symmetric variant of λ-calculus {42}.

The structural subsystem can be adapted to various sequent calculi. Restriction to intuitionistic logic can be obtained by demanding that the right-hand side has exactly one formula.

The presentation of this calculus with conjunctive and disjunctive additive connectives has been studied in [3, 5]. A variant with only commands, called $\mathcal{X}$, has been studied in [4], based on previous work in [2]. Various extensions of the system emphasizing different symmetries can be found in the literature.

Remarks: The system is obviously logically equivalent to Gentzen's **LK** when equipped with the corresponding connectives and observed through the sequents of the form $\Gamma \vdash \Delta$.

---

[1] Pierre-Louis Curien and Hugo Herbelin. "The duality of computation". In: *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming, ICFP 2000, Montreal, Canada, September 18-21, 2000*. SIGPLAN Notices 35(9). ACM, 2000, pp. 233–243. ISBN: 1-58113-202-6.

[2] Christian Urban. "Classical Logic and Computation". Ph.D. Thesis. University of Cambridge, Oct. 2000.

[3] Philip Wadler. "Call-by-value is dual to call-by-name". In: *Proceedings of ICFP 2003, Uppsala, Sweden, August 25-29, 2003*. Ed. by Colin Runciman and Olin Shivers. Vol. 38(9). SIGPLAN Notices. ACM, 2003, pp. 189–201. ISBN: 1-58113-756-7.

[4] Steffen van Bakel, Stephane Lengrand, and Pierre Lescanne. "The Language X: Circuits, Computations and Classical Logic". In: *Theoretical Computer Science, 9th Italian Conference, ICTCS 2005, Siena, Italy, October 12-14, 2005, Proceedings*. Ed. by Mario Coppo, Elena Lodi, and G. Michele Pinna. Vol. 3701. LNCS. Springer, 2005, pp. 81–96. ISBN: 3-540-29106-7.

[5] Hugo Herbelin. *C'est maintenant qu'on calcule: au cœur de la dualité*. Habilitation thesis. Dec. 2005.

Constructive Modal Logic S4 (CS4) (2000)

$$\frac{}{\Delta, A \vdash A}\ ax \qquad \frac{\Gamma \vdash A \quad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}\ cut \qquad \frac{}{\Gamma, \bot \vdash A}\ \bot\mathcal{L}$$

$$\frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}\ \vee\mathcal{L} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee\mathcal{R} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee\mathcal{R}$$

$$\frac{\Gamma, A \vdash C}{\Gamma, A \wedge B \vdash C}\ \wedge\mathcal{L} \qquad \frac{\Gamma, B \vdash C}{\Gamma, A \wedge B \vdash C}\ \wedge\mathcal{L} \qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge\mathcal{R}$$

$$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \to B \vdash C}\ \to\mathcal{L} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \to\mathcal{R} \qquad \frac{\Gamma, A \vdash B}{\Gamma, \Box A \vdash B}\ \Box\mathcal{L}$$

$$\frac{\Box\Gamma \vdash A}{\Box\Gamma, \Delta \vdash \Box A}\ \Box\mathcal{R} \qquad \frac{\Box\Gamma, A \vdash \Diamond B}{\Delta, \Box\Gamma, \Diamond A \vdash \Diamond B}\ \Diamond\mathcal{L} \qquad \frac{\Gamma \vdash A}{\Gamma \vdash \Diamond A}\ \Diamond\mathcal{R}$$

Clarifications: Left contexts, denoted Γ or Δ, are multisets of formulas. Furthermore, if $\Gamma = A_1, \ldots, A_n$, then $\Box\Gamma = \Box A_1, \ldots, \Box A_n$.

History: The intuitionistic system for S4 that we are calling constructive S4 (CS4) here, was originally described by Prawitz in his Natural Deduction book [1] in 1965. This system differs from what is more widely called now IS4, originally defined by Fisher-Servi [2] and thoroughly studied in Simpson's PhD thesis [3] in that it does not satisfy the distribution of possibility over disjunctions, either binary ($\Diamond(A \vee B) \to \Diamond A \vee \Diamond B$) or nullary ($\Diamond\bot \to \bot$). The calculus for CS4 was thoroughly investigated by Bierman and de Paiva in [4].

[1] Dag Prawitz. *Natural Deduction: A Proof-Theoretical Study*. Almqvist and Wiksell, Stockholm, 1965.

[2] G. Fisher-Servi. "Semantics for a class of intuitionistic modal calculi". In: *Italian Studies in the Philosophy of Science*. Ed. by Dalla Chiara and Maria Luisa. 1st ed. Vol. 47. Boston Studies in the Philosophy and History of Science. Springer Netherlands, 1981, pp. 59–72.

[3] Alex K Simpson. "The proof theory and semantics of intuitionistic modal logic". In: (1994).

[4] Gavin M. Bierman and Valeria CV de Paiva. "On an intuitionistic modal logic". In: *Studia Logica* 65.3 (2000), pp. 383–416.

Entry 51 by: Harley Eades III, Valeria de Paiva

Hybrid Logic (HL) (2001)

$$\frac{@_a\phi \quad @_a\psi}{@_a(\phi \wedge \psi)}\,(\wedge I) \qquad \frac{@_a(\phi \wedge \psi)}{@_a\phi}\,(\wedge E1) \qquad \frac{@_a(\phi \wedge \psi)}{@_a\psi}\,(\wedge E2)$$

$$\frac{\begin{array}{c}[@_a\phi]\\ \vdots\\ @_a\psi\end{array}}{@_a(\phi \to \psi)}\,(\to I) \qquad \frac{@_a(\phi \to \psi) \quad @_a\phi}{@_a\psi}\,(\to E) \qquad \frac{\begin{array}{c}[@_a\neg\phi]\\ \vdots\\ @_a\bot\end{array}}{@_a\phi}\,(\bot 1)^*$$

$$\frac{@_a\phi}{@_c@_a\phi}\,(@I) \qquad \frac{@_c@_a\phi}{@_a\phi}\,(@E) \qquad \frac{@_a\bot}{@_c\bot}\,(\bot 2)$$

$$\frac{\begin{array}{c}[@_a\Diamond c]\\ \vdots\\ @_c\phi\end{array}}{@_a\Box\phi}\,(\Box I)^\star \qquad \frac{@_a\Box\phi \quad @_a\Diamond e}{@_e\phi}\,(\Box E)$$

$$\frac{}{@_a a}\,(Ref) \qquad \frac{@_a c \quad @_a\phi}{@_c\phi}\,(Nom1)^* \qquad \frac{@_a c \quad @_a\Diamond b}{@_c\Diamond b}\,(Nom2)$$

$*$ ϕ is a propositional symbol (ordinary or a nominal).
$\star$ c does not occur in $@_a\Box\phi$ or in any undischarged assumptions other than the occurrences of $@_a\Diamond c$.

Clarifications: Hybrid logic is an extension of ordinary modal logic which allows explicit reference to individual points in a Kripke model. Formulas of HL are defined by $S ::= p \mid a \mid S \wedge S \mid S \to S \mid \bot \mid \Box S \mid @_a S$ where p ranges over ordinary propositional symbols and a ranges over nominals (a second sort of propositional symbols that refer to points in the model). As usual, $\neg\phi$ stands for $\phi \to \bot$ and $\Diamond\phi$ stands for $\neg\Box\neg\phi$.

History: This natural deduction system for classical HL was originally suggested in [1] and developed in [2]. A natural deduction system for intuitionistic hybrid logic can be found in the entry {54}. These and other proof systems are included in the book [4], which considers a spectrum of different hybrid logics (propositional, first-order, intensional first-order, and intuitionistic) and different types of proof systems for hybrid logic (natural deduction, Gentzen, tableau, and axiom systems). See [3] for a general introduction to hybrid logic.

Remarks: The system satisfies normalization, and normal derivations satisfy a version of the subformula property. Completeness is preserved when the system is extended with additional rules corresponding to first-order conditions on Kripke frames expressed by geometric theories.

---

[1] T. Braüner. "Natural Deduction for Hybrid Logic (Extended Abstract)". In: *Workshop Proceedings of Methods for Modalities 2*. Ed. by C. Areces and M. de Rijke. ILLC Amsterdam, 2001.

[2] T. Braüner. "Natural Deduction for Hybrid Logic". In: *Journal of Logic and Computation* 14 (2004), pp. 329–353.

[3] C. Areces and B. ten Cate. "Hybrid Logics". In: *Handbook of Modal Logic*. Ed. by P. Blackburn, J. van Benthem, and F. Wolter. Elsevier, 2007, pp. 821–868.

[4] T. Braüner. *Hybrid Logic and its Proof-Theory*. Vol. 37. Applied Logic Series. Springer, 2011.

Entry 52 by: Torben Braüner

Sequent Calculus TC (2002/2014)

$$\frac{\Gamma \vdash \Delta, \varphi\left\{\frac{s}{x}, \frac{t}{y}\right\}}{\Gamma \vdash \Delta, \left(TC_{x,y}\varphi\right)(s,t)} \ (sub)$$

$$\frac{\Gamma \vdash \Delta, \left(TC_{x,y}\varphi\right)(s,r) \quad \Gamma \vdash \Delta, \left(TC_{x,y}\varphi\right)(r,t)}{\Gamma \vdash \Delta, \left(TC_{x,y}\varphi\right)(s,t)} \ (trans)$$

$$\frac{\Gamma, \varphi(x,y) \vdash \Delta, \psi(x,y) \quad \Gamma, \psi\left\{\frac{u}{x}, \frac{v}{y}\right\}, \psi\left\{\frac{v}{x}, \frac{w}{y}\right\} \vdash \Delta, \psi\left\{\frac{u}{x}, \frac{w}{y}\right\}}{\Gamma, \left(TC_{x,y}\varphi\right)(s,t) \vdash \Delta, \psi\left\{\frac{s}{x}, \frac{t}{y}\right\}} \ (min)$$

Clarifications: The system is an extension of the sequent calculus for classical first-order logic, **LK** {6}. The letters Γ,Δ represent finite multisets of formulas, φ, ψ arbitrary formulas, x, y, u, v, w variables, and r, s, t terms. $\varphi\left\{\frac{t_1}{x_1}, ..., \frac{t_n}{x_n}\right\}$ stands for the result of simultaneously substituting t_i for x_i in φ ($i = 1, ..., n$). In all three rules the terms which are substituted should be free for substitution, and no forbidden capturing should occur. In Rule (*min*) x, y should not occur free in Γ and Δ, and u, v, w should not occur free in Γ, Δ, ϕ and ψ.

History: The sequent calculus was presented in [5], based on suggestions made in [4]. A similar sequent calculus for the reflexive transitive closure operator was also presented in [5]. Equivalent Hilbert-style systems for the reflexive transitive closure operator were suggested in [1, 2, 3].

Remarks: The sequent calculus generalizes Gentzen's calculus for Peano's Arithmetic. It is sound with respect to the intended semantics of the transitive closure operator. It is also sound and complete with respect to generalized Henkin-style semantics of the operator.

---

[1] Richard Milton Martin. "A Homogeneous System for Formal Logic". In: *Journal of Symbolic Logic* 8.1 (1943), pp. 1–23.

[2] Richard Milton Martin. "A Note on Nominalism and Recursive Functions". In: *Journal of Symbolic Logic* 14.1 (1949), pp. 27–31.

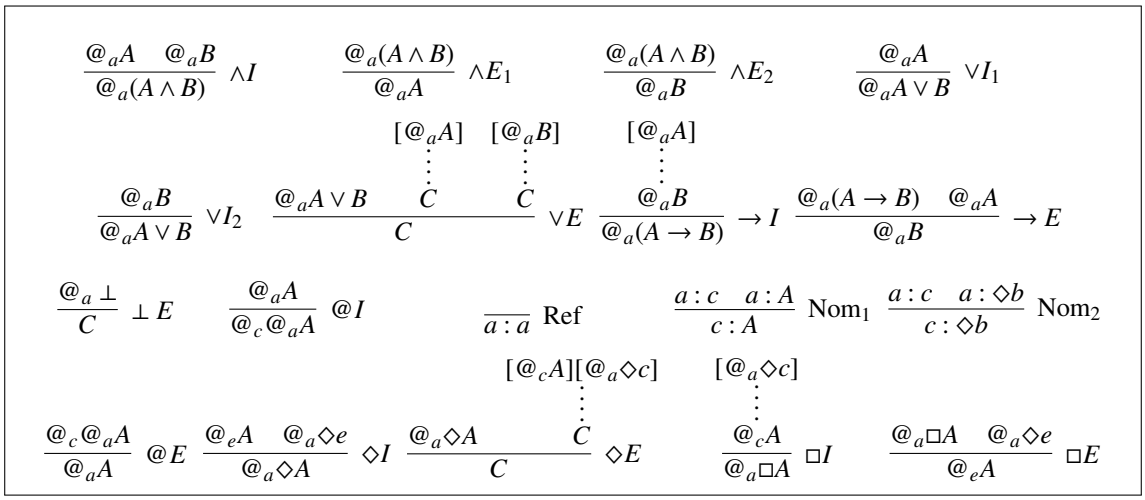
[3] John Myhill. "A Derivation of Number Theory from Ancestral Theory". In: *Journal of Symbolic Logic* 17.3 (1952), pp. 192–197.

[4] Arnon Avron. "Transitive Closure and the Mechanization of Mathematics". In: *Thirty Five Years of Automating Mathematics*. Ed. by F. D. Kamareddine. Vol. 28. Springer, Netherlands, 2003, pp. 149–171.

[5] Liron Cohen. "Ancestral Logic: A Proof Theoretical Study". In: *Logic, Language, Information, and Computation*. Ed. by Ulrich Kohlenbach et al. Vol. 8652. Lecture Notes in Computer Science. Springer, 2014, pp. 137–151.

Entry 53 by: Arnon Avron, Liron Cohen

Intuitionistic Hybrid Logic (IHL) (2003)



Clarifications: Formulas of IHL are defined by the following grammar:

$$A, B, C ::= p \mid a \mid A \wedge B \mid A \vee B \mid A \to B \mid \bot \mid \Box A \mid \Diamond A \mid @_aA$$

where a ranges over nominals and p propositional symbols. In the rule $\Diamond E$, c does not occur in $@_a\Diamond A$, in C, or in any undischarged assumptions other than the specified occurrences of $@_cA$ and $@_a\Diamond c$. Furthermore, in $\Box I$, c does not occur in $@_a\Box A$ or in any undischarged assumptions other than the specified occurrences of $@_a\Diamond c$. In the rule Nom_1, A, is any proposition (ordinary or nominal).

History: The Natural Deduction system for IHL was originally suggested in [2] and developed in [4]. This system adds nominals and satisfaction operators to a version of Intuitionistic Modal Logic described using labelled deduction, in the style of Simpson [1]. Axioms, or a Hilbert-style calculus version of the system, were provided in [3]. Some of the properties of the intuitionistic system, as well as a discussion of some of its applications to type systems in computing, appeared in [5].

---

[1] A. Simpson. "The Proof Theory and Semantics of Intuitionistic Modal logic". PhD thesis. University of Edinburgh, 1994.

[2] T. Braüner and V. de Paiva. "Towards Constructive Hybrid Logic (Extended Abstract)". In: *Workshop Proceedings of Methods for Modalities 3*. 15 pages. 2003.

[3] Torben Braüner. "Axioms for classical, intuitionistic, and paraconsistent hybrid logic". In: *Journal of Logic, Language and Information* 15.3 (2006), pp. 179–194.

[4] Torben Braüner and Valeria de Paiva. "Intuitionistic hybrid logic". In: *Journal of Applied Logic* 4.3 (2006), pp. 231–255.

[5] Torben Braüner. "Intuitionistic hybrid logic: Introduction and survey". In: *Inf. Comput.* 209.12 (2011), pp. 1437–1446. DOI: 10.1016/j.ic.2011.10.001.

Entry 54 by: Valeria de Paiva, Harley Eades III, Torben Braüner

Resolution for Monodic First-Order Temporal Logic (2003)

$$\frac{\mathcal{A} \to \bigcirc \mathcal{B}}{\neg \mathcal{A}}\ \textit{Step Resolution} \qquad \text{where } \mathcal{A} \to \bigcirc \mathcal{B} \text{ is a merged derived step clause and } \mathcal{U} \cup \{\mathcal{B}\} \models \bot$$

$$\frac{\forall x((\mathcal{A}_1 \wedge A_1(x)) \to \bigcirc(\mathcal{B}_1 \wedge B_1(x))) \quad \cdots \quad \forall x((\mathcal{A}_n \wedge A_n(x)) \to \bigcirc(\mathcal{B}_n \wedge B_n(x))) \quad \Diamond L(x)}{\forall x \bigwedge_{i=1}^{n}(\neg \mathcal{A}_i \vee \neg A_i(x))}\ \textit{Eventuality Resolution}$$

where $\forall x((\mathcal{A}_i \wedge A_1(x)) \to \bigcirc(\mathcal{B}_i \wedge B_i(x)))$, $1 \le i \le n$, are full merged step clauses such that for all i, $1 \le i \le n$, $\forall x((\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x)) \to \neg L(x))$ and $\forall x((\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x)) \to \bigvee_{j=1}^{n}(\mathcal{A}_j \wedge A_j(x))$ are valid.

$$\frac{\mathcal{A}_1 \to \bigcirc \mathcal{B}_1 \quad \cdots \quad \mathcal{A}_n \to \bigcirc \mathcal{B}_n \quad \Diamond L}{\bigwedge_{i=1}^{n} \neg \mathcal{A}_i}\ \textit{Ground Eventuality Resolution}$$

where $\Diamond L \in \mathcal{E}$, for a proposition or ground literal L, and $\mathcal{A}_i \to \bigcirc B_i$, $1 \le i \le n$, are merged derived step clauses such that for all i, $1 \le i \le n$, $(\mathcal{U} \wedge \mathcal{B}_i) \to \neg L$ and $(\mathcal{U} \wedge \mathcal{B}_i) \to \bigvee_{j=1}^{n} \mathcal{A}_j$ are valid.

Derivations terminate if $\mathcal{U} \cup \mathcal{I} \models \bot$, $\mathcal{U} \models \forall x \neg L(x)$ where $\Diamond L(x) \in \mathcal{E}$, or $\mathcal{U} \models \neg L$ where $\Diamond L \in \mathcal{E}$ for a proposition or ground literal L.

Clarifications: To determine the satisfiability of a formula of monodic first-order discrete linear time temporal logic over expanding domains, the formula is transformed into a *Monodic Temporal Problem in Divided Separated Normal Form* $\mathcal{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ where $\mathcal{U}$ and $\mathcal{I}$ are finite sets of arbitrary closed first-order formulae; $\mathcal{S}$ is a finite set of formulae of the form $p \to \bigcirc q$ (*original ground step clause*), where p and q are propositions, or $P(x) \to \bigcirc Q(x)$ (*original non-ground step clauses*), where P and Q are unary predicates and x is variable; and $\mathcal{E}$ is a set of *eventuality clauses* of the form $\Diamond l$, where l is a propositional literal, $\Diamond L(c)$, where $L(c)$ is a unary ground literal, or $\Diamond L(x)$, where $L(x)$ is a unary non-ground literal.

If $P_1(x) \to \bigcirc Q_1(x), \ldots, P_k(x) \to \bigcirc Q_k(x)$ are original non-ground step clauses, then $\forall x(P_1(x) \vee \cdots \vee P_k(x) \to \bigcirc \forall x(Q_1(x) \vee \cdots \vee Q_k(x))$, $\exists x(P_1(x) \wedge \cdots \wedge P_k(x) \to \bigcirc \exists x(Q_1(x) \wedge \cdots \wedge Q_k(x))$, and $P_i(c) \to Q_i(c)$, $1 \le i \le k$, where c is a constant occurring in $\mathcal{P}$, are *derived step clauses*. If $\Phi_1 \to \bigcirc \Psi_1, \ldots, \Phi_n \to \Psi_n$ are derived step clauses or original ground step clauses, then $\bigwedge_{i=1}^{n} \Phi_i \to \bigcirc \bigwedge_{i=1}^{n} \Psi_i$ is a *merged derived step clause*. If $\mathcal{A} \to \bigcirc \mathcal{B}$ is a merged derived step clause and $P_1(x) \to \bigcirc Q_1(x), \ldots, P_k(x) \to \bigcirc Q_k(x)$ are original step clauses, then $\forall x(\mathcal{A} \wedge \bigwedge_{i=1}^{k} P_i(x)) \to \bigcirc(\mathcal{B} \wedge \bigwedge_{i=1}^{k} Q_i(x))$ is a *full merged step clause*.

History: Related to {35}, this calculus was introduced in [1], a comprehensive description is given in [5]. It was extended to handling equality in [2] and a calculus closer to an implementable form was provided in [4]. It has been implemented in the prover TeMP [3]. Soundness and completeness are shown in [5].

[1] A. Degtyarev, M. Fisher, and B. Konev. "Monodic temporal resolution". In: *CADE-19*. Vol. 2741. LNAI. Springer, 2003, pp. 397–411. DOI: 10.1007/978-3-540-45085-6_35.

[2] B. Konev, A. Degtyarev, and M. Fisher. "Handling Equality in Monodic Temporal Resolution". In: *LPAR 2003*. Vol. 2850. LNCS. Springer, 2003, pp. 214–228.

[3] U. Hustadt, B. Konev, A. Riazanov, and A. Voronkov. "TeMP: A Temporal Monodic Prover". In: *IJCAR 2004*. Vol. 3097. LNCS. Springer, 2004, pp. 326–330. DOI: 10.1007/978-3-540-25984-8_23.

[4] B. Konev, A. Degtyarev, C. Dixon, M. Fisher, and U. Hustadt. "Mechanising First-Order Temporal Resolution". In: *Information and Computation* 199.1-2 (2005), pp. 55–86.

[5] A. Degtyarev, M. Fisher, and B. Konev. "Monodic temporal resolution". In: *ACM Trans. Comput. Log.* 7.1 (2006), pp. 108–150. DOI: 10.1145/1119439.1119443.

Entry 55 by: Clare Dixon, Michael Fisher, Ullrich Hustadt, Boris Konev

Model Evolution (2003)

$$\frac{\Lambda \vdash \Phi, L \vee C}{\Lambda, L\sigma \vdash \Phi, L \vee C \qquad \Lambda, (\overline{L}\sigma)^{\mathrm{sko}} \vdash \Phi, L \vee C}\ \textit{Split}$$

$$\frac{\Lambda, K, L \vdash \Phi}{\Lambda, K, L, L\sigma \vdash \Phi \qquad \Lambda, K, L, \overline{L}\sigma \vdash \Phi}\ \textit{Commit} \qquad \frac{\Lambda \vdash \Phi, L}{\Lambda, L \vdash \Phi, L}\ \textit{Assert} \qquad \frac{\Lambda \vdash \Phi, C}{\Lambda \vdash \square}\ \textit{Close}$$

Clarifications: Model Evolution is a refutationally complete calculus for first-order clause logic. The inference rules operate on sequents of the form $\Lambda \vdash \Phi$ where Λ is a set of literals and Φ is a clause set. Derivation trees are constructed top-down and start with the sequent $\neg v \vdash \psi$, where $\neg v$ is a pseudo-literal, representing the set of all negative literals, and ψ is the given clause set. The calculus derives (in the limit) a sequent $\Lambda \vdash \Phi$ such that the interpretation induced by Λ is a model of Φ unless ψ is unsatisfiable. All inference rules above are subject to certain applicability conditions, see [2]. Additional optional inference rules, not shown here, help improve performance in practice.

History: Model Evolution [2] improves the earlier FDPLL calculus [1], lifting the core of the propositional DPLL method to the first-order level. It has been extended by ordered paramodulation rules for equality reasoning [3], by lemma learning techniques inspired by modern CDCL SAT solvers [4] and by reasoning modulo background theories [5, 7]. It has been combined with the superposition calculus in [6].

[1] Peter Baumgartner. "FDPLL – A First-Order Davis-Putnam-Logeman-Loveland Procedure". In: *CADE-17 – The 17th International Conference on Automated Deduction*. Ed. by David McAllester. Vol. 1831. LNAI. Springer, 2000, pp. 200–219.

[2] Peter Baumgartner and Cesare Tinelli. "The Model Evolution Calculus". In: *CADE-19 – The 19th International Conference on Automated Deduction*. Ed. by Franz Baader. Vol. 2741. LNAI. Springer, 2003.

[3] Peter Baumgartner and Cesare Tinelli. "The Model Evolution Calculus with Equality". In: *CADE-20 – The 20th International Conference on Automated Deduction*. Ed. by Robert Nieuwenhuis. Vol. 3632. LNAI. Springer, 2005.

[4] Peter Baumgartner, Alexander Fuchs, and Cesare Tinelli. "Lemma Learning in the Model Evolution Calculus". In: *Logic for Programming, Artificial Intelligence and Reasoning (LPAR)*. Ed. by Miki Hermann and Andrei Voronkov. Vol. 4246. LNAI. Springer, 2006, pp. 572–586.

[5] Peter Baumgartner, Alexander Fuchs, and Cesare Tinelli. "ME(LIA) – Model Evolution With Linear Integer Arithmetic Constraints". In: *Proceedings of the 15th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR'08)*. Ed. by I. Cervesato, H. Veith, and A. Voronkov. Vol. 5330. LNAI. Springer, Nov. 2008, pp. 258–273.

[6] Peter Baumgartner and Uwe Waldmann. "Superposition and Model Evolution Combined". In: *CADE-22 – The 22nd International Conference on Automated Deduction*. Ed. by Renate Schmidt. Vol. 5663. LNAI. Springer, July 2009, pp. 17–34.

[7] Peter Baumgartner and Cesare Tinelli. "Model Evolution with Equality Modulo Built-in Theories". In: *CADE-23 – The 23nd International Conference on Automated Deduction*. Ed. by Nikolaj Bjørner and Viorica Sofronie-Stokkermans. Vol. 6803. LNAI. Springer, 2011, pp. 85–100.

Entry 56 by: Peter Baumgartner

Resolution for Monodic First-Order Temporal Logic (2003)

$$\frac{\mathcal{A} \to \bigcirc \mathcal{B}}{\neg \mathcal{A}} \textit{ Step Resolution} \qquad \text{where } \mathcal{A} \to \bigcirc \mathcal{B} \text{ is a merged derived step clause and } \mathcal{U} \cup \{\mathcal{B}\} \models \bot$$

$$\frac{\forall x((\mathcal{A}_1 \wedge A_1(x)) \to \bigcirc(\mathcal{B}_1 \wedge B_1(x))) \quad \cdots \quad \forall x((\mathcal{A}_n \wedge A_n(x)) \to \bigcirc(\mathcal{B}_n \wedge B_n(x))) \quad \Diamond L(x)}{\forall x \bigwedge_{i=1}^{n}(\neg \mathcal{A}_i \vee \neg A_i(x))} \textit{ Eventuality Resolution}$$

where $\forall x((\mathcal{A}_i \wedge A_1(x)) \to \bigcirc(\mathcal{B}_i \wedge B_i(x)))$, $1 \le i \le n$, are full merged step clauses such that for all i, $1 \le i \le n$, $\forall x((\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x)) \to \neg L(x))$ and $\forall x((\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x)) \to \bigvee_{j=1}^{n}(\mathcal{A}_j \wedge A_j(x))$ are valid.

$$\frac{\mathcal{A}_1 \to \bigcirc \mathcal{B}_1 \quad \cdots \quad \mathcal{A}_n \to \bigcirc \mathcal{B}_n \quad \Diamond L}{\bigwedge_{i=1}^{n} \neg \mathcal{A}_i} \textit{ Ground Eventuality Resolution}$$

where $\Diamond L \in \mathcal{E}$, for a proposition or ground literal L, and $\mathcal{A}_i \to \bigcirc B_i$, $1 \le i \le n$, are merged derived step clauses such that for all i, $1 \le i \le n$, $(\mathcal{U} \wedge \mathcal{B}_i) \to \neg L$ and $(\mathcal{U} \wedge \mathcal{B}_i) \to \bigvee_{j=1}^{n} \mathcal{A}_j$ are valid.

Derivations terminate if $\mathcal{U} \cup \mathcal{I} \models \bot$, $\mathcal{U} \models \forall x \neg L(x)$ where $\Diamond L(x) \in \mathcal{E}$, or $\mathcal{U} \models \neg L$ where $\Diamond L \in \mathcal{E}$ for a proposition or ground literal L.

Clarifications: To determine the satisfiability of a formula of monodic first-order discrete linear time temporal logic over expanding domains, the formula is transformed into a *Monodic Temporal Problem in Divided Separated Normal Form* $\mathcal{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ where $\mathcal{U}$ and $\mathcal{I}$ are finite sets of arbitrary closed first-order formulae; $\mathcal{S}$ is a finite set of formulae of the form $p \to \bigcirc q$ (*original ground step clause*), where p and q are propositions, or $P(x) \to \bigcirc Q(x)$ (*original non-ground step clauses*), where P and Q are unary predicates and x is variable; and $\mathcal{E}$ is a set of *eventuality clauses* of the form $\Diamond l$, where l is a propositional literal, $\Diamond L(c)$, where $L(c)$ is a unary ground literal, or $\Diamond L(x)$, where $L(x)$ is a unary non-ground literal.

If $P_1(x) \to \bigcirc Q_1(x), \ldots, P_k(x) \to \bigcirc Q_k(x)$ are original non-ground step clauses, then $\forall x(P_1(x) \vee \cdots \vee P_k(x) \to \bigcirc \forall x(Q_1(x) \vee \cdots \vee Q_k(x))$, $\exists x(P_1(x) \wedge \cdots \wedge P_k(x) \to \bigcirc \exists x(Q_1(x) \wedge \cdots \wedge Q_k(x))$, and $P_i(c) \to Q_i(c)$, $1 \le i \le k$, where c is a constant occurring in $\mathcal{P}$, are *derived step clauses*. If $\Phi_1 \to \bigcirc \Psi_1, \ldots, \Phi_n \to \Psi_n$ are derived step clauses or original ground step clauses, then $\bigwedge_{i=1}^{n} \Phi_i \to \bigcirc \bigwedge_{i=1}^{n} \Psi_i$ is a *merged derived step clause*. If $\mathcal{A} \to \bigcirc \mathcal{B}$ is a merged derived step clause and $P_1(x) \to \bigcirc Q_1(x), \ldots, P_k(x) \to \bigcirc Q_k(x)$ are original step clauses, then $\forall x(\mathcal{A} \wedge \bigwedge_{i=1}^{k} P_i(x)) \to \bigcirc(\mathcal{B} \wedge \bigwedge_{i=1}^{k} Q_i(x))$ is a *full merged step clause*.

History: Related to {35}, this calculus was introduced in [1], a comprehensive description is given in [5]. It was extended to handling equality in [2] and a calculus closer to an implementable form was provided in [4]. It has been implemented in the prover TeMP [3]. Soundness and completeness are shown in [5].

[1] A. Degtyarev, M. Fisher, and B. Konev. "Monodic temporal resolution". In: *CADE-19*. Vol. 2741. LNAI. Springer, 2003, pp. 397–411. DOI: 10.1007/978-3-540-45085-6_35.

[2] B. Konev, A. Degtyarev, and M. Fisher. "Handling Equality in Monodic Temporal Resolution". In: *LPAR 2003*. Vol. 2850. LNCS. Springer, 2003, pp. 214–228.

[3] U. Hustadt, B. Konev, A. Riazanov, and A. Voronkov. "TeMP: A Temporal Monodic Prover". In: *IJCAR 2004*. Vol. 3097. LNCS. Springer, 2004, pp. 326–330. DOI: 10.1007/978-3-540-25984-8_23.

[4] B. Konev, A. Degtyarev, C. Dixon, M. Fisher, and U. Hustadt. "Mechanising First-Order Temporal Resolution". In: *Information and Computation* 199.1-2 (2005), pp. 55–86.

[5] A. Degtyarev, M. Fisher, and B. Konev. "Monodic temporal resolution". In: *ACM Trans. Comput. Log.* 7.1 (2006), pp. 108–150. DOI: 10.1145/1119439.1119443.

Entry 57 by: Clare Dixon, Michael Fisher, Ullrich Hustadt, Boris Konev

Socratic Proofs for CPL (2003)

$$\frac{?(\Phi\ ;\ S\ '\ \alpha\ '\ T\ \vdash\ C\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ \alpha_1\ '\ \alpha_2\ '\ T\ \vdash\ C\ ;\ \Psi)}\ \mathbf{L}_\alpha \qquad \frac{?(\Phi\ ;\ S\ \vdash\ \alpha\ ;\ \Psi)}{?(\Phi\ ;\ S\ \vdash\ \alpha_1\ ;\ S\ \vdash\ \alpha_2\ ;\ \Psi)}\ \mathbf{R}_\alpha$$

$$\frac{?(\Phi\ ;\ S\ '\ \beta\ '\ T\ \vdash\ C\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ \beta_1\ '\ T\ \vdash\ C\ ;\ S\ '\ \beta_2\ '\ T\ \vdash\ C\ ;\ \Psi)}\ \mathbf{L}_\beta$$

$$\frac{?(\Phi\ ;\ S\ \vdash\ \beta\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ \beta_1^*\ \vdash\ \beta_2\ ;\ \Psi)}\ \mathbf{R}_\beta \qquad \frac{?(\Phi\ ;\ S\ '\ \neg\neg A\ '\ T\ \vdash\ C\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ A\ '\ T\ \vdash\ C\ ;\ \Psi)}\ \mathbf{L}_{\neg\neg}$$

Where:

α	α_1	α_2	β	β_1	β_2	β_1^*
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$	A
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B	$\neg A$
$\neg(A \rightarrow B)$	A	$\neg B$	$A \rightarrow B$	$\neg A$	B	A

Clarifications: The method of Socratic proofs is a method of transforming questions, but these are based on sequences of two-sided, single-conclusion sequents with sequences of formulas in both cedents. Φ, Ψ are finite (possibly empty) sequences of sequents. S, T are finite (possibly empty) sequences of formulas. The semicolon ';' is the concatenation sign for sequences of sequents, whereas ''' is the concatenation sign for sequences of formulas. A Socratic proof of sequent '$S \vdash A$' in $\mathbf{E}^*$ is a finite sequence of questions guided by the rules of $\mathbf{E}^*$, starting with '$?(S \vdash A)$' and ending with a question based on a sequence of basic sequents, where a *basic sequent* is a sequent containing the same formula in both of its cedents or containing a formula and its negation in the antecedent.

History: The method has been first presented in [1]. Calculus $\mathbf{E}^*$ is called *erotetic* calculus since it is a calculus of questions (*erotema* means *question* in Greek). Proof-theoretically, it may be viewed as a calculus of hypersequents with ';' understood conjunctively. It is grounded in Inferential Erotetic Logic (cf. [2]).

Remarks: A sequent '$S \vdash A$' has a Socratic proof in $\mathbf{E}^*$ iff A is CPL-entailed by the set of terms of S. The rules are invertible.

---

[1] Andrzej Wiśniewski. "Socratic Proofs". In: *Journal of Philosophical Logic* 33.3 (2004), pp. 299–326.

[2] Andrzej Wiśniewski. *Questions, Inferences, and Scenarios*. London: College Publications, 2013.

Entry 58 by: Dorota Leszczyńska-Jasion

Socratic Proofs for FOL (2004)

The rules of calculus $\mathbf{E}^*$ (see {58}) and the quantifier rules:

$$\frac{?(\Phi\ ;\ S\ '\ \forall x_i A\ '\ T \vdash C\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ \forall x_i A\ '\ A(x_i/\tau)\ '\ T \vdash C\ ;\ \Psi)}\ \mathbf{L}_\forall \qquad \frac{?(\Phi\ ;\ S \vdash \forall x_i A\ ;\ \Psi)}{?(\Phi\ ;\ S \vdash A(x_i/\tau)\ ;\ \Psi)}\ \mathbf{R}_\forall$$

$$\frac{?(\Phi\ ;\ S\ '\ \exists x_i A\ '\ T \vdash C\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ A(x_i/\tau)\ '\ T \vdash C\ ;\ \Psi)}\ \mathbf{L}_\exists \qquad \frac{?(\Phi\ ;\ S \vdash \exists x_i A\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ \forall x_i \neg A \vdash A(x_i/\tau)\ ;\ \Psi)}\ \mathbf{R}_\exists$$

In $\mathbf{L}_\forall$, $\mathbf{R}_\exists$: x_i is free in A and τ is any parameter. In $\mathbf{L}_\exists$, $\mathbf{R}_\forall$: x_i is free in A, τ is a parameter which does not occur in the sequent distinguished in the premise.

$$\frac{?(\Phi\ ;\ S\ '\ \kappa\ '\ T \vdash C\ ;\ \Psi)}{?(\Phi\ ;\ S\ '\ \kappa^*\ '\ T \vdash C\ ;\ \Psi)}\ \mathbf{L}_\kappa \qquad \frac{?(\Phi\ ;\ S \vdash \kappa\ ;\ \Psi)}{?(\Phi\ ;\ S \vdash \kappa^*\ ;\ \Psi)}\ \mathbf{R}_\kappa$$

Where:

κ	κ^*
$\neg\exists x_i A$	$\forall x_i \neg A$
$\neg\forall x_i A$	$\exists x_i \neg A$
$\forall x_i A$, provided that x_i is not free in A	A
$\exists x_i A$, provided that x_i is not free in A	A

Clarifications: For notational conventions, see entry {58}. Socratic proofs for FOL start with questions concerning *pure sequents*, i.e. sequents formed with sentences only and containing no parameters. A Socratic proof of a pure sequent '$S \vdash A$' in $\mathbf{E}^{\mathsf{PQ}}$ is a finite sequence of questions guided by the rules of $\mathbf{E}^{\mathsf{PQ}}$, starting with '$?(S \vdash A)$' and ending with a question based on a sequence of basic sequents, where a *basic sequent* is a sequent containing the same formula in both of its cedents or containing a formula and its negation in the antecedent.

History: The method has been first presented in [1], together with a constructive completeness proof. All the rules of $\mathbf{E}^{\mathsf{PQ}}$ are invertible and there are no structural rules. The erotetic calculus $\mathbf{E}^{\mathsf{PQ}}$ may be reconstructed into a sequent calculus $\mathbf{G}^{\mathsf{PQ}}$ for FOL with invertible rules and no structural rules. $\mathbf{G}^{\mathsf{PQ}}$ has been first described in [1] and later examined in [2].

Remarks: A pure sequent '$S \vdash A$' has a Socratic proof in $\mathbf{E}^{\mathsf{PQ}}$ iff A is FOL-entailed by the set of terms of S.

[1] Andrzej Wiśniewski and Vasilyi Shangin. "Socratic Proofs for Quantifiers". In: *Journal of Philosophical Logic* 35.2 (2006), pp. 147–178.

[2] Dorota Leszczyńska-Jasion, Mariusz Urbański, and Andrzej Wiśniewski. "Socratic Trees". In: *Studia Logica* 101.5 (2013), pp. 959–986.

Entry 59 by: Dorota Leszczyńska-Jasion

Socratic Proofs for Modal Propositional K (2004)

The rules of calculus $\mathbf{E}^{\mathsf{K}}$:

$$\frac{?(\Phi\ ;\ \vdash S\ '\ (\alpha)^{\phi(i)}\ '\ T\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S\ '\ (\alpha_1)^{\phi(i)}\ '\ T\ ;\ \vdash S\ '\ (\alpha_2)^{\phi(i)}\ '\ T\ ;\ \Psi)}\ \mathbf{R}_{\alpha}$$

$$\frac{?(\Phi\ ;\ \vdash S\ '\ (\beta)^{\phi(i)}\ '\ T\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S\ '\ (\beta_1)^{\phi(i)}\ '\ (\beta_2)^{\phi(i)}\ '\ T\ ;\ \Psi)}\ \mathbf{R}_{\beta} \qquad \frac{?(\Phi\ ;\ \vdash S\ '\ (\neg\neg A)^{\phi(i)}\ '\ T\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S\ '\ (A)^{\phi(i)}\ '\ T\ ;\ \Psi)}\ \mathbf{R}_{\neg\neg}$$

$$\frac{?(\Phi\ ;\ \vdash S\ '\ (\mu)^{\phi(i)}\ '\ T\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S\ '\ (\mu_0)^{\phi(i),j}\ '\ T\ ;\ \Psi)}\ \mathbf{R}_{\mu} \qquad \frac{?(\Phi\ ;\ \vdash S\ '\ (\pi)^{\phi(i)}\ '\ T\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S\ '\ (\pi)^{\phi(i)}\ '\ (\pi_0)^{j}\ '\ T\ ;\ \Psi)}\ \mathbf{R}_{\pi}$$

where:

α	α_1	α_2	β	β_1	β_2	β_1^*	μ	μ_0	π	π_0
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$	A	$\Box A$	A	$\neg\Box A$	$\neg A$
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B	$\neg A$	$\neg\Diamond A$	$\neg A$	$\Diamond A$	A
$\neg(A \rightarrow B)$	A	$\neg B$	$A \rightarrow B$	$\neg A$	B	A				

$\phi(i)$ is a finite sequence of numerals ending with i (an index of a formula)
$\phi(i), j$ is a concatenation of $\phi(i)$ and $\langle j \rangle$

In $\mathbf{R}_{\mu}$, numeral j must be new with respect to the sequent distinguished in the premise. In $\mathbf{R}_{\pi}$, the pair $\langle i, j \rangle$ is present in the premise sequent.

Clarifications: The method of Socratic proofs is a method of transforming questions but with a clear proof-theoretic interpretation (see also {58}, {59}). The rules act upon right-sided sequents, with sequences of *indexed* formulas in the succedents. The indices store the semantic information. A Socratic proof starts with a question concerning $?(\vdash (A)^1)$ and ends with a question based on a sequence of basic sequents, where a *basic sequent* is a sequent containing indexed formulas of the forms $B^{\phi(i)}$, $(\neg B)^{\psi(i)}$.

History: The proof system has been presented in [1], the completeness proof may be found in [2].

Remarks: A sequent $\vdash (A)^1$ has a Socratic proof in $\mathbf{E}^{\mathsf{K}}$ iff A is K-valid.

---

[1] Dorota Leszczyńska. "Socratic Proofs for some Normal Modal Propositional Logics". In: *Logique et Analyse* 47.185-188 (2004), pp. 259–285.

[2] Dorota Leszczyńska-Jasion. *The Method of Socratic Proofs for Normal Modal Propositional Logics*. Poznań: Adam Mickiewicz University Press, 2007.

Entry 60 by: Dorota Leszczyńska-Jasion

Socratic Proofs for Modal Propositional Logics (2004)

The rules of $\mathbf{E}^{\mathsf{L}}$ are the rules of $\mathbf{E}^{\mathsf{K}}$ (see {60}), where the proviso of applicability of $\mathbf{R}_{\mu}$ depends on the logic L and is a combination of some of the following clauses:

1. $\langle i, j\rangle$ is present in the premise sequent
2. $i = j$
3. $\langle j, i\rangle$ is present in the premise sequent
4. there is a sequence $i_1, \ldots, i_n$ such that $i_1 = i$, $i_n = j$ and each $\langle i_k, i_{k+1}\rangle$, where $1 \geq k \geq n-1$, is present in the premise sequent
5. there is a sequence $i_1, \ldots, i_n$ such that $i_1 = i$, $i_n = j$ and for each $\langle i_k, i_{k+1}\rangle$, where $1 \geq k \geq n-1$, $\langle i_k, i_{k+1}\rangle$ or $\langle i_{k+1}, i_k\rangle$ is present in the premise sequent
6. there are sequences $i_1, \ldots, i_n$ and $j_1, \ldots, j_m$ such that $i_1 = 1$, $i_n = i$, $j_1 = 1$, $j_m = j$ and for each $\langle i_k, i_{k+1}\rangle$, where $1 \geq k \geq n-1$, $\langle i_k, i_{k+1}\rangle$ is present in the premise sequent and for each $\langle j_l, j_{l+1}\rangle$, where $1 \geq l \geq m-1$, $\langle j_l, j_{l+1}\rangle$ is present in the premise sequent

L	proviso	L	proviso	L	proviso
K,KD	(1)	K4, KD4	(4)	K5, D5	(6)
KT	(1) or (2)	S4	(2) or (4)	K45, D45	(4) or (6)
KB, KDB	(1) or (3)	KB4	(5)		
KTB	(1) or (2) or (3)	S5	(2) or (5)		

Calculi for logics: KD, KDB, KD4, KD5, KD45 have also the following rule, where j is new:

$$\frac{?(\Phi\ ;\ \vdash S\ '\ (\pi)^{\phi(i)}\ '\ T\ ;\ \Psi)}{?(\Phi\ ;\ \vdash S\ '\ (\pi)^{\phi(i)}\ '\ (\pi_0)^{j}\ '\ T\ ;\ \Psi)}\ \mathbf{R}_{\pi\mathsf{D}}$$

Clarifications: See {60}, {58}, {59} for more comments.

History: The proof system has been presented in [1], the completeness proof may be found in [2], and extensions to some non-basic modal logics in [3].

Remarks: A sequent $\vdash (A)^1$ has a Socratic proof in $\mathbf{E}^{\mathsf{L}}$ iff A is L-valid.

[1] Dorota Leszczyńska. "Socratic Proofs for some Normal Modal Propositional Logics". In: 47.185-188 (2004), pp. 259–285.

[2] Dorota Leszczyńska-Jasion. *The Method of Socratic Proofs for Normal Modal Propositional Logics*. Adam Mickiewicz University Press, 2007.

[3] Dorota Leszczyńska-Jasion. "The Method of Socratic Proofs for Modal Propositional Logics: K5, S4.2, S4.3, S4M, S4F, S4R and G". In: *Studia Logica* 89.3 (2008), pp. 371–405.

Entry 61 by: Dorota Leszczyńska-Jasion

LK$\mu\tilde{\mu}$ in sequent-free tree form (2005)

STRUCTURAL SUBSYSTEM

$$\frac{\vdash A \quad A \vdash}{\vdash} \; Cut$$

$$\frac{\begin{array}{c}[\vdash A]\\ \vdots \\ \vdash\end{array}}{A \vdash} \; Focus_L \qquad \frac{\begin{array}{c}[A \vdash]\\ \vdots \\ \vdash\end{array}}{\vdash A} \; Focus_R$$

INTRODUCTION RULES

$$\frac{A_i \vdash}{A_1 \wedge A_2 \vdash} \; \wedge^i_L \qquad \frac{\vdash A_1 \quad \vdash A_2}{\vdash A_1 \wedge A_2} \; \wedge_R$$

$$\frac{A_1 \vdash \quad A_2 \vdash}{A_1 \vee A_2 \vdash} \; \vee_L \qquad \frac{\vdash A_i}{\vdash A_1 \vee A_2} \; \vee^i_R$$

$$\frac{\vdash A \quad B \vdash}{A \to B \vdash} \; \to_L \qquad \frac{\begin{array}{c}[\vdash A]\\ \vdots \\ \vdash B\end{array}}{\vdash A \to B} \; \to_R$$

$$\frac{A[y] \vdash}{\exists x A[x] \vdash} \; \exists_L \qquad \frac{\vdash A[t]}{\vdash \exists x A[x]} \; \exists_R$$

$$\frac{A[t] \vdash}{\forall x A[x] \vdash} \; \forall_L \qquad \frac{\vdash A[y]}{\vdash \forall x A[x]} \; \forall_R$$

$$\frac{}{\bot \vdash} \; \bot_L \qquad \frac{}{\vdash \top} \; \top_R$$

Clarifications: There are three kinds of nodes, $\vdash A$ for asserting formulas, $A \vdash$ for refuting formulas, and $\vdash$ for expressing a contradiction. Negation $\neg A$ can be defined as $A \to \bot$. In the rules $\exists_E$ and $\forall_R$, y is assumed fresh in all the unbracketed assumption formula upon which that the derivation of $A(y)$ depends.

History: The purpose of this system is to show that the original distinction in Gentzen [1] between natural deduction presented as a tree of formulas and sequent calculus presented as a tree of sequents is no longer relevant. It is known from at least Howard [4] that natural deduction can be presented with sequents. The above formulation shows that systems based on left and right introductions ("sequent-calculus style") can be presented as a sequent-free tree of formulas [6].

The terminology "sequent calculus" seems to have become popular from [2] followed then e.g. by [3] who were associating the term "sequents" to Gentzen's LJ and LK systems. The terminology having lost the connection to its etymology, this motivated some authors to use alternative terminologies such as "L" systems [7].

Remarks: As pointed out e.g. in [5] in the context of natural deduction, to obtain a computationally non-degenerate proof-as-program correspondence with a presentation of a calculus as a tree of formulas, the bracketed assumptions have to be annotated with the exact occurrence of the rule which bracketed them. Then, annotation by proof-terms can optionally be added as in {50}.

Entry 62 by: Hugo Herbelin

[1] Gerhard Gentzen. “Untersuchungen über das logische Schließen”. In: 39 (1935). English Translation in [**Szabo69**], “Investigations into logical deduction”, pages 68-131, pp. 176–210, 405–431.
[2] Dag Prawitz. *Natural Deduction, a Proof-Theoretical Study*. Almqvist and Wiksell, Stockholm, 1965.
[3] Anne S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*. Vol. 344. Lecture Notes in Mathematics. Berlin: Springer-Verlag, 1973.
[4] William A. Howard. “The formulae-as-types notion of constructions”. In: *to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Unpublished manuscript of 1969. Academic Press, 1980.
[5] Herman Geuvers. “Logics and Type Systems”. Section 3.2. Ph.D. Thesis. Katholieke Universiteit Nijmegen, Sept. 1993.
[6] Hugo Herbelin. *C’est maintenant qu’on calcule: au cœur de la dualité*. Habilitation thesis. Dec. 2005.
[7] Guillaume Munch-Maccagnoni. “Focalisation and Classical Realisability”. In: *Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal, September 7-11, 2009. Proceedings*. Ed. by Erich Grädel and Reinhard Kahle. Vol. 5771. Lecture Notes in Computer Science. Springer, 2009, pp. 409–423. DOI: 10.1007/978-3-642-04027-6_30. URL: http://dx.doi.org/10.1007/978-3-642-04027-6_30.

Conditional Labelled Sequent Calculi SeqS (2003-2007)

$(\mathbf{AX})$ $\Gamma, x : P \vdash \Delta, x : P$ $(P$ atomic$)$ $\qquad$ $(\mathbf{A}\bot)$ $\Gamma, x : \bot \vdash \Delta$

$$\frac{\Gamma, x : A \Rightarrow B \vdash x \xrightarrow{A} y, \Delta \qquad \Gamma, x : A \Rightarrow B, y : B \vdash \Delta}{\Gamma, x : A \Rightarrow B \vdash \Delta}\ (\Rightarrow \mathbf{L}) \qquad \frac{\Gamma, x \xrightarrow{A} y \vdash \Delta, y : B}{\Gamma \vdash \Delta, x : A \Rightarrow B}\ (\Rightarrow \mathbf{R})\ (y \notin \Gamma, \Delta)$$

$$\frac{u : A \vdash u : B \qquad u : B \vdash u : A}{\Gamma, x \xrightarrow{A} y \vdash x \xrightarrow{B} y, \Delta}\ (\mathbf{EQ}) \qquad \frac{\Gamma, x \xrightarrow{A} y, y : A \vdash \Delta}{\Gamma, x \xrightarrow{A} y \vdash \Delta}\ (\mathbf{ID}) \qquad \frac{\Gamma \vdash x \xrightarrow{A} x, x : A, \Delta}{\Gamma \vdash x \xrightarrow{A} x, \Delta}\ (\mathbf{MP})$$

$$\frac{\Gamma, x \xrightarrow{A} y \vdash \Delta, x : A \qquad \Gamma[x/u, y/u], u \xrightarrow{A} u \vdash \Delta[x/u, y/u]}{\Gamma, x \xrightarrow{A} y \vdash \Delta}\ (\mathbf{CS})\ (x \neq y, u \notin \Gamma, \Delta)$$

$$\frac{\Gamma x \xrightarrow{A} y \vdash \Delta, x \xrightarrow{A} z \qquad (\Gamma x \xrightarrow{A} y \vdash \Delta)[y/u, z/u]}{\Gamma x \xrightarrow{A} y \vdash \Delta}\ (\mathbf{CEM})\ (y \neq z, u \notin \Gamma, \Delta)$$

Given a sequent Γ and labels x and u, $\Gamma[x/u]$ is the sequent obtained by replacing in Γ all occurrences of x with u.

Clarifications: Conditional logics extend classical logic with formulas of the form $A \Rightarrow B$. SeqS uses *selection function* semantics: $A \Rightarrow B$ is true in a world w if B is true in the set of worlds selected by the selection function f for A and w (that are most similar to w). SeqS manipulates *labelled* formulas, where labels represent worlds, of the form $x : A$ (A is true in x) and $x \xrightarrow{A} y$ (y belongs to $f(x, A)$). SeqS considers *normal* conditional logics, such that if A and B are true in the same worlds, then $f(w, A) = f(w, B)$. The rule $(\mathbf{EQ})$ takes care of normality. Besides the rules shown, SeqS includes standard rules for propositional connectives.

History: The calculi SeqS have been introduced in [3]. The theorem prover CondLean, implementing SeqS calculi in Prolog, has been presented in [1, 2].

Remarks: Completeness is a consequence of the admissibility of cut. The calculi SeqS can be used to obtain a PSPACE decision procedure for the respective conditional logics and to develop goal-directed proof procedures.

[1] Nicola Olivetti and Gian Luca Pozzato. "CondLean: A Theorem Prover for Conditional Logics". In: *Automated Reasoning with Analytic Tableaux and Related Methods - Proceedings of TABLEAUX 2003 (12th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods)*. Ed. by M. C. Mayer and F. Pirri. Vol. 2796. LNAI. Roma, Italy: Springer, Sept. 2003, pp. 264–270. DOI: 10.1007/978-3-540-45206-5_23.

[2] Nicola Olivetti and Gian Luca Pozzato. "CondLean 3.0: Improving Condlean for Stronger Conditional Logics". In: *Automated Reasoning with Analytic Tableaux and Related Methods - Proceedings of TABLEAUX 2005 (14th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods)*. Ed. by B. Beckert. Vol. 3702. LNAI. Koblenz, Germany: Springer, Sept. 2005, pp. 328–332. DOI: 10.1007/11554554_27.

[3] Nicola Olivetti, Gian Luca Pozzato, and Camilla Schwind. "A sequent calculus and a theorem prover for standard conditional logics". In: *ACM Transactions on Computational Logic* 8 (4 2007), pp. 1–51. DOI: 10.1145/1276920.1276924.

Entry 63 by: Nicola Olivetti, Gian Luca Pozzato, Camilla Schwind

Preferential Tableau Calculi $\mathcal{T}P^{\mathbf{T}}$ (2005-2009)

$$\Gamma, P, \neg P \ (\mathbf{AX}) \quad \text{with } P \text{ atomic} \qquad \frac{\Gamma, \neg\Box\neg A; \Sigma}{A, \Box\neg A, \Gamma^{\Box}, \Gamma^{\Box^{\downarrow}}, \Gamma^{\vdash\pm}, \Sigma; \emptyset}\ (\Box^{-})$$

$$\frac{\Gamma, \neg(A \vdash B); \Sigma}{A, \Box\neg A, \neg B, \Gamma^{\vdash\pm}; \emptyset}\ (\vdash^{-}) \qquad \frac{\Gamma, A \vdash B; \Sigma}{\Gamma, \neg A; \Sigma, A \vdash B \qquad \Gamma, \neg\Box\neg A; \Sigma, A \vdash B \qquad \Gamma, B; \Sigma, A \vdash B}\ (\vdash^{+})$$

Clarifications: According to Kraus, Lehmann and Magidor (KLM) [1], defeasible knowledge is represented by a (finite) set of nonmonotonic conditionals $A \vdash B$ (normally the A's are B's). Models are possible-world structures equipped with a preference relation (irreflexive and transitive for **P**) among worlds or states. The meaning of $A \vdash B$ is that B holds in the worlds/states where A holds and that are *minimal* with respect to the preference relation.

The calculus $\mathcal{T}P^{\mathbf{T}}$ is based on the idea of interpreting the preference relation as an accessibility relation: a conditional $A \vdash B$ holds in a model if B is true in all minimal A-worlds, where a world w is an A-world if it satisfies A, and it is a minimal A-world if there is no A-world w' preferred to w.

Nodes are pairs $\Gamma; \Sigma$, where Γ is a set of formulas and Σ is a set of conditional formulas $A \vdash B$. Σ is used to keep track of positive conditionals $A \vdash B$ to which the rule $(\vdash^{+})$ has already been applied: the idea is that one does not need to apply $(\vdash^{+})$ on the same conditional formula $A \vdash B$ *more than once in the same world*. When $(\vdash^{+})$ is applied to a formula $A \vdash B \in \Gamma$, then $A \vdash B$ is moved from Γ to Σ in the conclusions of the rule, so that it is no longer available for further applications in the current world. The dynamic rules re-introduce formulas from Σ to Γ in order to allow further applications of $(\vdash^{+})$ in new worlds.

Given Γ, we define:

- $\Gamma^{\Box} = \{\Box\neg A \mid \Box\neg A \in \Gamma\}$
- $\Gamma^{\Box^{\downarrow}} = \{\neg A \mid \Box\neg A \in \Gamma\}$
- $\Gamma^{\vdash^{+}} = \{A \vdash B \mid A \vdash B \in \Gamma\}$
- $\Gamma^{\vdash^{-}} = \{\neg(A \vdash B) \mid \neg(A \vdash B) \in \Gamma\}$
- $\Gamma^{\vdash\pm} = \Gamma^{\vdash^{+}} \cup \Gamma^{\vdash^{-}}$

Besides the rules shown above, the calculus $\mathcal{T}P^{\mathbf{T}}$ also includes standard rules for propositional connectives.

History: In [1] Kraus, Lehmann and Magidor proposed a formalization of nonmonotonic reasoning that was early recognized as a landmark. According to their framework, defeasible knowledge is represented by a (finite) set of nonmonotonic conditionals or assertions of the form $A \vdash B$, whose reading is *normally (or typically) the A's are B's*. The operator "$\vdash$" is nonmonotonic, in the sense that $A \vdash B$ does not imply $A \wedge C \vdash B$. The calculus $\mathcal{T}P^{\mathbf{T}}$ and extensions for all the logics of the KLM family are proposed in [4]. The theorem provers KLMLean and FreeP implementing the tableau calculi have been presented at [3, 2].

Remarks: The calculus $\mathcal{T}P^{\mathbf{T}}$ can be used to define a decision procedure and obtain a complexity bound for the preferential logic **P**, namely that it is **coNP**-complete.

---

[1] S. Kraus, D. Lehmann, and M. Magidor. "Nonmonotonic Reasoning, Preferential Models and Cumulative Logics". In: *Artificial Intelligence* 44.1-2 (1990), pp. 167–207.

Entry 64 by: Laura Giordano, Valentina Gliozzi, Nicola Olivetti, Gian Luca Pozzato

[2] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. "An Implementation of a Free-variable Tableaux for KLM Preferential Logic P of Nonmonotonic Reasoning:" in: *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing - Proceedings of AI*IA 2007 (10th Congress of Italian Association for Artificial Intelligence)*. Ed. by Roberto Basili and Maria Teresa Pazienza. Vol. 4733. Lecture Notes in Artificial Intelligence LNAI. Roma, Italy: Springer, Sept. 2007, pp. 84–96. DOI: 10.1007/978-3-540-74782-6_9.

[3] Laura Giordano, Valentina Gliozzi, and Gian Luca Pozzato. "KLMLean 2.0: A Theorem Prover for KLM Logics of Nonmonotonic Reasoning". In: *Automated Reasoning with Analytic Tableaux and Related Methods - Proceedings of TABLEAUX 2007 (16th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods)*. Ed. by Nicola Olivetti. Vol. 4548. Lecture Notes in Artificial Intelligence LNAI. Aix en Provence, France: Springer, July 2007, pp. 238–244. DOI: 10.1007/978-3-540-73099-6_19.

[4] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. "Analytic Tableaux Calculi for KLM Logics of Nonmonotonic Reasoning". In: *ACM Transactions on Computational Logic (ToCL)* 10 (3 2009), pp. 1–47. DOI: 10.1145/1507244.1507248.

HO Sequent Calculi $\mathcal{G}_\beta$ and $\mathcal{G}_{\beta\mathfrak{fb}}$ (2003-2009)

Basic Rules

$$\frac{\Delta, s}{\Delta, \neg\neg s}\,\mathcal{G}(\neg) \qquad \frac{\Delta, \neg s \quad \Delta, \neg t}{\Delta, \neg(s \vee t)}\,\mathcal{G}(\vee_-) \qquad \frac{\Delta, s, t}{\Delta, (s \vee t)}\,\mathcal{G}(\vee_+)$$

$$\frac{\Delta, \neg(sl)\downarrow_\beta \quad l_\alpha \text{ closed term}}{\Delta, \neg\Pi^\alpha s}\,\mathcal{G}(\Pi^l_-) \qquad \frac{\Delta, (sc)\downarrow_\beta \quad c_\delta \text{ new symbol}}{\Delta, \Pi^\alpha s}\,\mathcal{G}(\Pi^c_+)$$

Initialization

$$\frac{s \text{ atomic (and } \beta\text{-normal)}}{\Delta, s, \neg s}\,\mathcal{G}(init) \qquad \frac{\Delta, (s \doteq^o t) \quad s,t \text{ atomic}}{\Delta, \neg s, t}\,\mathcal{G}(Init^{\doteq})$$

Extensionality

$$\frac{\Delta, (\forall X_\alpha\, sX \doteq^\beta tX)\downarrow_\beta}{\Delta, (s \doteq^{\alpha\to\beta} t)}\,\mathcal{G}(\mathfrak{f}) \qquad \frac{\Delta, \neg s, t \quad \Delta, \neg t, s}{\Delta, (s \doteq^o t)}\,\mathcal{G}(\mathfrak{b})$$

Decomposition

$$\frac{\Delta, (s^1 \doteq^{\alpha_1} t^1) \cdots \Delta, (s^n \doteq^{\alpha_n} t^n) \quad n \geq 1, \beta \in \{o, \iota\}, h_{\overline{\alpha^n}\to\beta} \in \Sigma}{\Delta, (h\overline{s^n} \doteq^\beta h\overline{t^n})}\,\mathcal{G}(d)$$

One-sided sequent calculus $\mathcal{G}_\beta$ is defined by the rules $\mathcal{G}(init)$, $\mathcal{G}(\neg)$, $\mathcal{G}(\vee_-)$, $\mathcal{G}(\vee_+)$, $\mathcal{G}(\Pi^l_-)$ and $\mathcal{G}(\Pi^c_+)$. Calculus $\mathcal{G}_{\beta\mathfrak{fb}}$ extends $\mathcal{G}_\beta$ by the additional rules $\mathcal{G}(\mathfrak{b})$, $\mathcal{G}(\mathfrak{f})$, $\mathcal{G}(d)$, and $\mathcal{G}(Init^{\doteq})$.

Clarifications: Δ and Δ' are finite sets of β-normal closed formulas of classical higher-order logic (HOL; Church's Type Theory) [4]. Δ, s denotes the set $\Delta \cup \{s\}$. Let $\alpha,\beta,o \in T$. HOL *terms* are defined by the grammar (c_α denotes typed constants and X_α typed variables distinct from c_α): $s,t ::= c_\alpha \mid X_\alpha \mid (\lambda X_\alpha s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\, t_\alpha)_\beta \mid (\neg_{o\to o}\, s_o)_o \mid (s_o \vee_{o\to o\to o} t_o)_o \mid (\Pi_{(\alpha\to o)\to o}\, s_{\alpha\to o})_o$. *Leibniz equality* $\doteq^\alpha$ at type α is defined as $s_\alpha \doteq^\alpha t_\alpha := \forall P_{\alpha\to o}(\neg Ps \vee Pt)$. For each simply typed λ-term s there is a unique *β-normal form* (denoted $s\downarrow_\beta$). HOL formulas are defined as terms of type o. A *non-atomic formula* is any formula whose β-normal form is of the form $[c\overline{\mathbf{A}^n}]$ where c is a logical constant. An *atomic formula* is any other formula. In order to prove that a (closed) conjecture c logically follows from a (possibly empty) set of (closed) axioms $\{a^1,\ldots,a^n\}$, we start from the initial sequent $\Delta := \{c, \neg a^1, \ldots, \neg a^n\}$ and reason backwards by applying the inference rules.

History: These calculi were presented in [3], and earlier (two-sided) related versions in [1] and [2].

Remarks: $\mathcal{G}_\beta$ is sound and complete for elementary type theory ($\mathcal{G}_\beta$ is thus also sound for HOL). $\mathcal{G}_{\beta\mathfrak{fb}}$ is sound and complete for HOL. Moreover, both calculi are cut-free and they do not admit cut-simulation [3].

[1] Christoph Benzmüller, Chad Brown, and Michael Kohlhase. *Semantic Techniques for Cut-Elimination in Higher Order Logic.* Tech. rep. Saarland University, Saarbrücken, Germany and Carnegie Mellon University, Pittsburgh, USA, 2003.

[2] Chad E. Brown. "Set Comprehension in Church's Type Theory". See also Chad E. Brown, *Automated Reasoning in Higher-Order Logic*, College Publications, 2007. PhD thesis. Department of Mathematical Sciences, Carnegie Mellon University, 2004.

[3] Christoph Benzmüller, Chad Brown, and Michael Kohlhase. "Cut-Simulation and Impredicativity". In: *Logical Methods in Computer Science* 5.1:6 (2009), pp. 1–21.

[4] Peter Andrews. "Church's Type Theory". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2014. 2014.

Entry 65 by: Christoph Benzmüller

Extensional HO RUE-Resolution (1999-2013)

Normalisation Rules

$$\frac{\mathbf{C} \vee [\mathbf{A} \vee \mathbf{B}]^{\mathsf{tt}}}{\mathbf{C} \vee [\mathbf{A}]^{\mathsf{tt}} \vee [\mathbf{B}]^{\mathsf{tt}}} \vee^{\mathsf{tt}} \qquad \frac{\mathbf{C} \vee [\mathbf{A} \vee \mathbf{B}]^{\mathsf{ff}}}{\begin{array}{l}\mathbf{C} \vee [\mathbf{A}]^{\mathsf{ff}} \\ \mathbf{C} \vee [\mathbf{B}]^{\mathsf{ff}}\end{array}} \vee^{\mathsf{ff}} \qquad \frac{\mathbf{C} \vee [\neg \mathbf{A}]^{\mathsf{tt}}}{\mathbf{C} \vee [\mathbf{A}]^{\mathsf{ff}}} \neg^{\mathsf{tt}} \qquad \frac{\mathbf{C} \vee [\neg \mathbf{A}]^{\mathsf{ff}}}{\mathbf{C} \vee [\mathbf{A}]^{\mathsf{tt}}} \neg^{\mathsf{ff}}$$

$$\frac{\mathbf{C} \vee [\Pi^\tau \mathbf{A}]^{\mathsf{tt}} \quad X^\tau \text{ fresh variable}}{\mathbf{C} \vee [\mathbf{A}\, X]^{\mathsf{tt}}} \Pi^{\mathsf{tt}} \qquad \frac{\mathbf{C} \vee [\Pi^\tau \mathbf{A}]^{\mathsf{ff}} \quad \mathtt{sk}^\tau \text{ Skolem term}}{\mathbf{C} \vee [\mathbf{A}\, \mathtt{sk}^\tau]^{\mathsf{ff}}} \Pi^{\mathsf{ff}}$$

Resolution, Factorisation and Primitive Substitution

$$\frac{[\mathbf{A}]^{p_1} \vee \mathbf{C} \quad [\mathbf{B}]^{p_2} \vee \mathbf{D} \quad p_1 \neq p_2}{\mathbf{C} \vee \mathbf{D} \vee [\mathbf{A} = \mathbf{B}]^{\mathsf{ff}}} \mathsf{res} \qquad \frac{\mathbf{C} \vee [\mathbf{A}]^p \vee [\mathbf{B}]^p}{\mathbf{C} \vee [\mathbf{A}]^p \vee [\mathbf{A} = \mathbf{B}]^{\mathsf{ff}}} \mathsf{fac}$$

$$\frac{[Q_\tau \overline{\mathbf{A}}^n]^p \vee \mathbf{C} \quad \mathbf{P} \in \mathcal{AB}_\tau^{(k)} \text{ for logic connective } k}{([Q_\tau \overline{\mathbf{A}}^n]^p \vee \mathbf{C})[\mathbf{P}/Q]} \mathsf{prim\_subst}$$

Extensionality and Pre-unification

$$\frac{\mathbf{C} \vee [\mathbf{A}^{\sigma\tau} = \mathbf{B}^{\sigma\tau}]^{\mathsf{tt}} \quad X^\tau \text{ fresh variable}}{\mathbf{C} \vee [\mathbf{A}\, X = \mathbf{B}\, X]^{\mathsf{tt}}} \textsc{FuncPos} \qquad \frac{\mathbf{C} \vee [\mathbf{A}^o = \mathbf{B}^o]^{\mathsf{tt}}}{\mathbf{C} \vee [\mathbf{A}^o \longleftrightarrow \mathbf{B}^o]^{\mathsf{tt}}} \textsc{BoolPos}$$

$$\frac{\mathbf{C} \vee [\mathbf{A}^{\sigma\tau} = \mathbf{B}^{\sigma\tau}]^{\mathsf{ff}} \quad \mathtt{sk}^\tau \text{ Skol. term}}{\mathbf{C} \vee [\mathbf{A}\, \mathtt{sk} = \mathbf{B}\, \mathtt{sk}]^{\mathsf{ff}}} \textsc{FuncNeg} \qquad \frac{\mathbf{C} \vee [\mathbf{A}^o = \mathbf{B}^o]^{\mathsf{ff}}}{\mathbf{C} \vee [\mathbf{A}^o \longleftrightarrow \mathbf{B}^o]^{\mathsf{ff}}} \textsc{BoolNeg}$$

$$\frac{\mathbf{C} \vee [h^{\sigma\tau} \overline{\mathbf{A}^\alpha}^k = h^{\sigma\tau} \overline{\mathbf{B}^\alpha}^k]^{\mathsf{ff}}}{\mathbf{C} \vee \overline{[\mathbf{A}_i = \mathbf{B}_i]^{\mathsf{ff}}}^{i \leq k}} \textsc{Dec} \qquad \frac{\mathbf{C} \vee [X = \mathbf{A}]^{\mathsf{ff}} \quad X \notin \mathrm{FV}(\mathbf{A})}{\mathbf{C}[\mathbf{A}/X]} \textsc{Subst}$$

$$\frac{\mathbf{C} \vee [\mathbf{A} = \mathbf{A}]^{\mathsf{ff}}}{\mathbf{C}} \textsc{Triv} \qquad \frac{\mathbf{C} \vee [F^\tau \overline{\mathbf{A}}^n = h \overline{\mathbf{B}}^m]^{\mathsf{ff}} \quad \mathbf{G} \in \mathcal{AB}_\tau^{(h)}}{\mathbf{C} \vee [F = \mathbf{G}]^{\mathsf{ff}} \vee [F \overline{\mathbf{A}}^n = h \overline{\mathbf{B}}^m]^{\mathsf{ff}}} \textsc{FlexRigid}$$

Choice

$$\frac{C := \mathbf{C}' \vee [\mathbf{A}[E_{(\alpha\to o)\to\alpha} \mathbf{B}]]^p \quad \begin{array}{l}\epsilon \in \mathsf{CFs},\ E = \epsilon \text{ or } E \in \mathit{freeVars}(C), \\ \mathit{freeVars}(\mathbf{B}) \subseteq \mathit{freeVars}(C), Y \text{ fresh}\end{array}}{[\mathbf{B}\, Y]^{\mathsf{ff}} \vee [\mathbf{B}\, (\epsilon_{\alpha(\alpha o)} \mathbf{B})]^{\mathsf{tt}}} \mathsf{choice}$$

$$\frac{[PX]^{\mathsf{ff}} \vee [P(f_{(\alpha\to o)\to\alpha} P)]^{\mathsf{tt}}}{\mathsf{CFs} \longleftarrow \mathsf{CFs} \cup \{f_{(\alpha\to o)\to\alpha}\}} \mathsf{detectChoiceFn}$$

Optional additional rules include (a) exhaustive universal instantion rule for (selective) finite domains, (b) detection and removal of Leibniz equations and Andrews equations, and (c) splitting. Like detectChoiceFn these rules are admissible.

Clarifications: $\mathbf{A}$ and $\mathbf{B}$ are metavariables ranging over terms of HOL [8]; see also {65}). The logical connectives are $\neg$, $\vee$, Π^τ (universal quantification over variables of type τ), and $=^\tau$ (equality on terms of type τ). Types are shown only if unclear in context. For example, in rule choice the variable $E^{\alpha(\alpha o)}$ is of function type, also written as $(\alpha \to o) \to \alpha$. Variables like F are presented as upper case symbols and constant symbols like h are lower case. α equality and $\beta\eta$-normalisation are treated implicit, meaning that all clauses are implicitly normalised. $\mathbf{C}$ and $\mathbf{D}$ are metavariables ranging over clauses, which are disjunctions of literals.

Entry 66 by: Christoph Benzmüller

These disjunctions are implicitly assumed associative and commutative; the latter also applies to all equations. Literals are formulas shown in square brackets and labelled with a *polarity* (either tt or ff), e.g. $[\neg X]^{\mathrm{ff}}$ denotes the negation of $\neg X$. $\mathrm{FV}(\mathbf{A})$ denotes the free variables of term $\mathbf{A}$. $\mathcal{AB}^{(h)}_{\tau}$ is the set of approximating bindings for head h and type τ. $\epsilon_{\alpha(\alpha o)}$ is a choice operator and CFs is a set of dynamically collected choice functions symbols; CFs is initialised with a single choice function.

History: The original calculus (without choice) has been presented in [4] and [5]. Recent modifications and extensions (e.g. choice) are discussed in [7] and [6]. The calculus is inspired by and extends Huet's constrained resolution [2, 1] and the extensional resolution calculus in [3].

Remarks: The calculus works for classical higher-order logic with Henkin semantics and choice. Soundness and completeness has been discussed in [4] and [5]. In the prover LEO-II, the factorisation rule is for performance reasons restricted to binary clauses and a (parametrisable) depth limit is employed for pre-unification. Such restrictions are a (deliberate) source for incompleteness.

---

[1] Gérard P. Huet. "Constrained Resolution: A Complete Method for Higher Order Logic". PhD thesis. Case Western Reserve University, 1972.

[2] Gérard P. Huet. "A Mechanization of Type Theory". In: *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*. 1973, pp. 139–146.

[3] Christoph Benzmüller and Michael Kohlhase. "Extensional Higher-Order Resolution". In: *Automated Deduction - CADE-15*. LNAI 1421. Springer, 1998, pp. 56–71.

[4] Christoph Benzmüller. "Extensional Higher-Order Paramodulation and RUE-Resolution". In: *Automated Deduction - CADE-16*. LNCS 1632. Springer, 1999, pp. 399–413.

[5] Christoph Benzmüller. "Comparing Approaches to Resolution based Higher-Order Theorem Proving". In: *Synthese* 133.1-2 (2002), pp. 203–235.

[6] Christoph Benzmüller and Nik Sultana. "LEO-II Version 1.5". In: *PxTP 2013*. Vol. 14. EPiC Series. EasyChair, 2013, pp. 2–10.

[7] Nik Sultana and Christoph Benzmüller. "Understanding LEO-II's Proofs". In: *IWIL 2012*. Vol. 22. EPiC Series. EasyChair, 2013, pp. 33–52.

[8] Peter Andrews. "Church's Type Theory". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2014. 2014.

Focused LK (2007)

Asynchronous introduction rules

$$\frac{}{\vdash \Gamma \Uparrow t^-, \Theta} \qquad \frac{\vdash \Gamma \Uparrow B_1, \Theta \quad \vdash \Gamma \Uparrow B_2, \Theta}{\vdash \Gamma \Uparrow B_1 \wedge^- B_2, \Theta} \qquad \frac{\vdash \Gamma \Uparrow \Theta}{\vdash \Gamma \Uparrow f^-, \Theta} \qquad \frac{\vdash \Gamma \Uparrow B_1, B_2, \Theta}{\vdash \Gamma \Uparrow B_1 \vee^- B_1, \Theta}$$

$$\frac{\vdash \Gamma \Uparrow [y/x]B, \Theta}{\vdash \Gamma \Uparrow \forall x.B, \Theta}$$

Synchronous introduction rules

$$\frac{}{\vdash \Gamma \Downarrow t^+} \qquad \frac{\vdash \Gamma \Downarrow B_1 \quad \vdash \Gamma \Downarrow B_2}{\vdash \Gamma \Downarrow B_1 \wedge^+ B_2} \qquad \frac{\vdash \Gamma \Downarrow B_i}{\vdash \Gamma \Downarrow B_1 \vee^+ B_2}\ i \in \{1,2\} \qquad \frac{\vdash \Gamma \Downarrow [t/x]B}{\vdash \Gamma \Downarrow \exists x.B}$$

Identity rules

$$\frac{P \text{ atomic}}{\vdash \neg P, \Gamma \Downarrow P}\ \mathit{init} \qquad \frac{\vdash \Gamma \Uparrow B \quad \vdash \Gamma \Uparrow \neg B}{\vdash \Gamma \Uparrow \cdot}\ \mathit{cut}$$

Structural rules

$$\frac{\vdash \Gamma, C \Uparrow \Theta}{\vdash \Gamma \Uparrow C, \Theta}\ \mathit{store} \qquad \frac{\vdash \Gamma \Uparrow N}{\vdash \Gamma \Downarrow N}\ \mathit{release} \qquad \frac{\vdash P, \Gamma \Downarrow P}{\vdash P, \Gamma \Uparrow \cdot}\ \mathit{decide}$$

Here, Γ ranges over multisets of polarized formulas; Θ ranges over lists of polarized formulas; P denotes a positive formula; N denotes a negative formula; C denotes either a negative formula or a positive atom; and B denotes an unrestricted polarized formula. The negation in $\neg B$ denotes the negation normal form of the de Morgan dual of B. The right introduction rule for $\forall$ has the the usual eigenvariable restriction that y is not free in any formula in the conclusion sequent.

Clarifications: This proof system involves *polarized* (negative normal) formulas of first-order classical logic: in order to polarize a formula B, one must assign the status of "positive" or "negative" bias to all atomic formulas and replace all occurrences of truth with either t^+ or t^- and replace all occurrences of conjunctions with either $\wedge^+$ or $\wedge^-$; similarly, all occurrences of false and disjunctions must be polarized into f^+, f^-, $\vee^+$, and $\vee^-$. If there are n occurrences of propositional connectives in B, there are 2^n ways to polarize B. The *positive connectives* are f^+, $\vee^+$, t^+, $\wedge^+$, and $\exists$ while the *negative connectives* are t^-, $\wedge^-$, f^-, $\vee^-$, and $\forall$. A formula is *positive* it is a positive atom or has a top-level positive connective; similarly a formula is *negative* if it is a negative atom or has a top-level negative connective.

There are two kinds of sequents in this proof system, namely, $\vdash \Gamma \Uparrow \Theta$ and $\vdash \Gamma \Downarrow B$, where Γ is a multiset of polarized formulas, B is a polarized formula, and Θ is a list of polarized formulas. The list structure of Θ can be replaced by a multiset.

History: This focused proof system is a slight variation of the proof systems in [5, 4]. A multifocus variant of **LKF** has been described in [6]. The design of **LKF** borrows strongly from Andreoli's focused proof system for linear logic [2] and Girard's LC proof system [1]. The first-order versions of the LKT and LKQ proof systems of [3] can be seen subsystems of **LKF**.

[1] Jean-Yves Girard. "A new constructive logic: classical logic". In: 1 (1991), pp. 255–296.

Entry 67 by: Dale Miller

[2] Jean-Marc Andreoli. “Logic Programming with Focusing Proofs in Linear Logic”. In: 2.3 (1992), pp. 297–347.

[3] V. Danos, J.-B. Joinet, and H. Schellinx. “LKT and LKQ: sequent calculi for second order logic based upon dual linear decompositions of classical implication”. In: *Advances in Linear Logic*. Ed. by J.-Y. Girard, Y. Lafont, and L. Regnier. London Mathematical Society Lecture Note Series 222. Cambridge University Press, 1995, pp. 211–224.

[4] Chuck Liang and Dale Miller. “Focusing and Polarization in Intuitionistic Logic”. In: *CSL 2007: Computer Science Logic*. Ed. by J. Duparc and T. A. Henzinger. Vol. 4646. LNCS. Springer, 2007, pp. 451–465.

[5] Chuck Liang and Dale Miller. “Focusing and Polarization in Linear, Intuitionistic, and Classical Logics”. In: *Theoretical Computer Science* 410.46 (2009), pp. 4747–4768.

[6] Kaustuv Chaudhuri, Stefan Hetzl, and Dale Miller. “A Multi-Focused Proof System Isomorphic to Expansion Proofs”. In: (2014).

Focused LJ (2007)

Asynchronous Introduction Rules

$$\frac{\Gamma \Uparrow B_1 \vdash B_2 \Uparrow}{\Gamma \Uparrow \cdot \vdash B_1 \supset B_2 \Uparrow} \qquad \frac{\Gamma \Uparrow \cdot \vdash B_1 \Uparrow \quad \Gamma \Uparrow \cdot \vdash B_2 \Uparrow}{\Gamma \Uparrow \cdot \vdash B_1 \wedge^- B_2 \Uparrow} \qquad \frac{}{\Gamma \Uparrow \cdot \vdash t^- \Uparrow}$$

$$\frac{\Gamma \Uparrow \cdot \vdash [y/x]B \Uparrow}{\Gamma \Uparrow \cdot \vdash \forall x.B \Uparrow} \qquad \frac{\Gamma \Uparrow [y/x]B, \Theta \vdash \mathcal{R}}{\Gamma \Uparrow \exists x.B, \Theta \vdash \mathcal{R}} \qquad \frac{}{\Gamma \Uparrow f^+, \Theta \vdash \mathcal{R}}$$

$$\frac{\Gamma \Uparrow B_1, B_2, \Theta \vdash \mathcal{R}}{\Gamma \Uparrow B_1 \wedge^+ B_2, \Theta \vdash \mathcal{R}} \qquad \frac{\Gamma \Uparrow \Theta \vdash \mathcal{R}}{\Gamma \Uparrow t^+, \Theta \vdash \mathcal{R}} \qquad \frac{\Gamma \Uparrow B_1, \Theta \vdash \mathcal{R} \quad \Gamma \Uparrow B_2, \Theta \vdash \mathcal{R}}{\Gamma \Uparrow B_1 \vee^+ B_2, \Theta \vdash \mathcal{R}}$$

Synchronous Introduction Rules

$$\frac{\Gamma \vdash B_1 \Downarrow \quad \Gamma \Downarrow B_2 \vdash E}{\Gamma \Downarrow B_1 \supset B_2 \vdash E} \qquad \frac{\Gamma \Downarrow [t/x]B \vdash E}{\Gamma \Downarrow \forall x.B \vdash E} \qquad \frac{\Gamma \Downarrow B_i \vdash E}{\Gamma \Downarrow B_1 \wedge^- B_2 \vdash E}\ i \in \{1,2\}$$

$$\frac{\Gamma \vdash B_i \Downarrow}{\Gamma \vdash B_1 \vee^+ B_2 \Downarrow} \qquad \frac{}{\Gamma \vdash t^+ \Downarrow} \qquad \frac{\Gamma \vdash B_1 \Downarrow \quad \Gamma \vdash B_2 \Downarrow}{\Gamma \vdash B_1 \wedge^+ B_2 \Downarrow} \qquad \frac{\Gamma \vdash [t/x]B \Downarrow}{\Gamma \vdash \exists x.B \Downarrow}$$

Identity rules

$$\frac{N \text{ atomic}}{\Gamma \Downarrow N \vdash N}\ I_l \qquad \frac{P \text{ atomic}}{\Gamma, P \vdash P \Downarrow}\ I_r \qquad \frac{\Gamma \Uparrow \cdot \vdash B \Uparrow \cdot \qquad \Gamma \Uparrow B \vdash \cdot \Uparrow E}{\Gamma \Uparrow \cdot \vdash \cdot \Uparrow E}\ Cut$$

Structural rules

$$\frac{\Gamma, N \Downarrow N \vdash E}{\Gamma, N \Uparrow \cdot \vdash \cdot \Uparrow E}\ D_l \qquad \frac{\Gamma \vdash P \Downarrow}{\Gamma \Uparrow \cdot \vdash \cdot \Uparrow P}\ D_r \qquad \frac{\Gamma \Uparrow P \vdash \cdot \Uparrow E}{\Gamma \Downarrow P \vdash E}\ R_l \qquad \frac{\Gamma \Uparrow \cdot \vdash N \Uparrow \cdot}{\Gamma \vdash N \Downarrow}\ R_r$$

$$\frac{C, \Gamma \Uparrow \Theta \vdash \mathcal{R}}{\Gamma \Uparrow C, \Theta \vdash \mathcal{R}}\ S_l \qquad \frac{\Gamma \Uparrow \cdot \vdash \cdot \Uparrow E}{\Gamma \Uparrow \cdot \vdash E \Uparrow \cdot}\ S_r$$

Here, Θ ranges over multisets of polarized formulas; Γ ranges over lists of polarized formulas; P denotes a positive formula; N denotes a negative formula; C denotes either a negative formula or a positive atom; and E denotes either a positive formula or a negative atom; and B denotes an unrestricted polarized formula. The introduction rule for $\forall$ has the usual eigenvariable restriction that y is not free in any formula in the conclusion sequent.

Clarifications: This proof system involves *polarized* formulas of first-order intuitionistic logic: in order to polarize a formula B, one must assign the status of "positive" or "negative" bias to all atomic formulas and replace all occurrences of truth with either t^+ or t^- and all occurrences of conjunction with either $\wedge^+$ or $\wedge^-$. If there are n occurrences of truth and conjunction in B, there are 2^n ways to do this replacement. Similarly, we replace the false and disjunction with f^+ and $\vee^+$ since only the positive polarization for these connectives are available in **LJF**. (Assigning polarization in classical logic is different: see the **LKF** proof system {67}.) The *positive connectives* are f^+, $\vee^+$, t^+, $\wedge^+$, and $\exists$ while the *negative connectives* are t^-, $\wedge^-$, $\supset$, and $\forall$. A formula is *positive* if it is a positive atom or has a top-level positive connective; similarly a formula is *negative* if it is a negative atom or has a top-level negative connective.

Entry 68 by: Dale Miller

There are two kinds of sequents in this proof system. One kind contains a single $\Downarrow$ on either the right or the left of the turnstyle ($\vdash$) and are of the form $\Gamma \Downarrow B \vdash E$ or $\Gamma \vdash B \Downarrow$: in both of these cases, the formula B is the *focus* of the sequent. The other kind of sequent has an occurrence of $\Uparrow$ on each side of the turnstyle, eg., $\Gamma \Uparrow \Theta \vdash \Delta_1 \Uparrow \Delta_2$, and is such that the union of the two multisets Δ_1 and Δ_2 contains exactly one formula: that is, one of these multisets is empty and the other is a singleton. When writing asynchronous rules that introduce a connective on the left-hand side, we write $\mathcal{R}$ to denote $\Delta_1 \Downarrow \Delta_2$.

Note that in the asynchronous phase, a right introduction rule is applied only when the left asynchronous zone Γ is empty. Similarly, a left-introduction rule in the async phase introduces the connective at the top-level of the first formula in that context. The scheduling of introduction rules during this phase can be assigned arbitrarily and the zone Γ can be interpreted as a multiset instead of a list.

The choice of how to polarize an unpolarized formula does not affect provability in LJF but can make a big impact on the structure of LJF proofs that can be built.

History: This focused proof system is a slight variation of the proof system in [6, 5]. **LJF** can be seen as a generalization to the MJ sequent system of Howe [2]. Other focused proof systems, such as LJT [1], LJQ/LJQ' [4], and λRCC [3] can be directly emulated within **LJF** by making the appropriate choice of polarization.

---

[1] Hugo Herbelin. "Séquents qu'on calcule: de l'interprétation du calcul des séquents comme calcul de lambda-termes et comme calcul de stratégies gagnantes". PhD thesis. Université Paris 7, 1995.

[2] J. M. Howe. "Proof Search Issues in Some Non-Classical Logics". Available as University of St Andrews Research Report CS/99/1. PhD thesis. University of St Andrews, 1998.

[3] Radha Jagadeesan, Gopalan Nadathur, and Vijay Saraswat. "Testing concurrent systems: An interpretation of intuitionistic logic". In: *FSTTCS05*. Vol. 3821. LNCS. Hyderabad, India: Springer, 2005, pp. 517–528.

[4] R. Dyckhoff and S. Lengrand. "LJQ: a strongly focused calculus for intuitionistic logic". In: *Computability in Europe 2006*. Ed. by A. Beckmann *et al.* Vol. 3988. Springer, 2006, pp. 173–185.

[5] Chuck Liang and Dale Miller. "Focusing and Polarization in Intuitionistic Logic". In: *CSL 2007: Computer Science Logic*. Ed. by J. Duparc and T. A. Henzinger. Vol. 4646. LNCS. Springer, 2007, pp. 451–465.

[6] Chuck Liang and Dale Miller. "Focusing and Polarization in Linear, Intuitionistic, and Classical Logics". In: 410.46 (2009), pp. 4747–4768.

$\lambda\Pi$-Calculus Modulo (2007)

Typing Rules for Terms

$$\frac{\Gamma \vdash^{ctx} \Delta}{\Gamma;\Delta \vdash \mathbf{Type} : \mathbf{Kind}} \textbf{ (Sort)}$$

$$\frac{\Gamma \vdash^{ctx} \Delta \qquad (c : A) \in \Gamma}{\Gamma;\Delta \vdash c : A} \textbf{ (Constant)} \qquad \frac{\Gamma \vdash^{ctx} \Delta \qquad (x : A) \in \Delta}{\Gamma;\Delta \vdash x : A} \textbf{ (Variable)}$$

$$\frac{\Gamma;\Delta \vdash t : \Pi x : A.B \qquad \Gamma;\Delta \vdash u : A}{\Gamma;\Delta \vdash tu : B[x/u]} \textbf{ (Application)}$$

$$\frac{\Gamma;\Delta(x : A) \vdash t : B \qquad \Gamma;\Delta \vdash \Pi x : A.B : s}{\Gamma;\Delta \vdash \lambda x : A.t : \Pi x : A.B} \textbf{ (Abstraction)}$$

$$\frac{\Gamma;\Delta \vdash A : \mathbf{Type} \qquad \Gamma;\Delta(x : A) \vdash B : s}{\Gamma;\Delta \vdash \Pi x : A.B : s} \textbf{ (Product)}$$

$$\frac{\Gamma;\Delta \vdash t : A \qquad \Gamma;\Delta \vdash B : s \qquad A \equiv_{\beta\Gamma} B}{\Gamma;\Delta \vdash t : B} \textbf{ (Conversion)}$$

Well-Formedness for Local Contexts

$$\frac{\Gamma \ \mathbf{wf}}{\Gamma \vdash^{ctx} \emptyset} \qquad \frac{\Gamma \vdash^{ctx} \Delta \qquad \Gamma;\Delta \vdash U : \mathbf{Type} \qquad x \notin dom(\Delta)}{\Gamma \vdash^{ctx} \Delta(x : U)}$$

Well-Formedness Rules for Global Contexts

$$\frac{}{\emptyset \ \mathbf{wf}} \qquad \frac{\Gamma \ \mathbf{wf} \qquad \Gamma;\emptyset \vdash U : \mathbf{Type}}{\Gamma(c : U) \ \mathbf{wf}} \qquad \frac{\Gamma \ \mathbf{wf} \qquad \Gamma;\emptyset \vdash K : \mathbf{Kind}}{\Gamma(C : K) \ \mathbf{wf}}$$

$$\frac{\Gamma \ \mathbf{wf} \qquad \rightarrow_\beta \cup \rightarrow_{\Gamma\Xi} \text{ is confluent} \qquad \begin{array}{c}(\forall i)\Gamma \vdash u_i \hookrightarrow v_i \\ \Xi = (u_1 \hookrightarrow v_1)\ldots(u_n \hookrightarrow v_n)\end{array}}{\Gamma\Xi \ \mathbf{wf}}$$

Clarifications: The $\lambda\Pi$-Calculus Modulo is an extension of the λ-Calculus with dependent types and rewrite rules. Computational equivalence is extended from β-equivalence to $\beta\Gamma$-equivalence ($\equiv_{\beta\Gamma}$), the congruence generated by β-reduction and the rewrite rules ($u \hookrightarrow v$) in the global context Γ.

History: The $\lambda\Pi$-Calculus Modulo has been introduced by Cousineau and Dowek [1] as an expressive logical framework. It has been used to design *shallow encodings* of many logics and calculus such as functional Pure Type Systems [1], Higher-Order Logic [4], the Calculus of Inductive Constructions [2], resolution and superposition [3], or the ς-calculus [5]. The well-formedness rules for global contexts were not part of the original type system and have been introduced by Saillard [6]. The $\lambda\Pi$-Calculus Modulo is implemented in the proof checker Dedukti [7].

Remarks: Confluence of the rewriting relation $\rightarrow_{\beta\Gamma}$ is required to guarantee subject reduction. This requirement can be weakened to confluence for a notion of rewriting modulo β [6]. Decidability of type inference depends on strong normalization.

Entry 69 by: Ronan Saillard

[1] Denis Cousineau and Gilles Dowek. “Embedding Pure Type Systems in the Lambda-Pi-Calculus Modulo”. In: *Typed Lambda Calculi and Applications, 8th International Conference (TLCA)*. 2007, pp. 102–117.

[2] M. Boespflug and G. Burel. “CoqInE : Translating the calculus of inductive constructions into the $\lambda\Pi$-calculus modulo.” In: *The Second International Workshop on Proof Exchange for Theorem Proving (PxTP)*. 2012.

[3] G. Burel. “A Shallow Embedding of Resolution and Superposition Proofs into the $\lambda\Pi$-Calculus Modulo”. In: *The Third International Workshop on Proof Exchange for Theorem Proving (PxTP ’13)*. 2013.

[4] A. Assaf and G. Burel. “Translating HOL to Dedukti”. 2014. URL: `https://hal.archives-ouvertes.fr/hal-01097412`.

[5] R. Cauderlier and C. Dubois. “Objects and Subtyping in the $\lambda\Pi$-Calculus Modulo”. In: *Post-proceedings of Types for Proofs and Programs (TYPES ’14)*. 2015.

[6] R. Saillard. “Rewriting Modulo β in the $\lambda\Pi$-Calculus Modulo”. In: *11th International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP)*. 2015.

[7] M. Boespflug, Q. Carbonneaux, O. Hermant, and R. Saillard. *Dedukti*. URL: `http://dedukti.gforge.inria.fr`.

Ordered Fine-Grained Resolution with Selection for Monodic First-Order Temporal Logic (2009)

The calculus consists of six *inference rules* operating on individual clauses, and *derivation rules* operating on sets of clauses:

$$\frac{D_1 \vee L_1 \quad \neg L_2 \vee D_2}{(D_1 \vee D_2)\sigma} \qquad \frac{D_3 \vee L_1 \vee L_2}{(D_3 \vee L_1)\sigma} \qquad \frac{C_1 \to \bigcirc\bot}{\neg C_1}$$

$$\frac{C_1 \to \bigcirc(D_1 \vee L_1) \quad C_2 \to \bigcirc(\neg L_2 \vee D_2)}{(C_1 \wedge C_2)\sigma \to \bigcirc(D_1 \vee D_2)\sigma} \qquad \frac{C_1 \to \bigcirc(D_1 \vee L_1) \quad \neg L_2 \vee D_2}{C_1\sigma \to \bigcirc(D_1 \vee D_2)\sigma} \qquad \frac{C_3 \to \bigcirc(D_3 \vee L_1 \vee L_2)}{C_3\sigma \to \bigcirc(D_3 \vee L_1)\sigma}$$

where σ is a most general unifier of the literals L_1 and L_2 such that σ does not map variables from C_1 or C_2 into a constant or a functional term, L_1 is eligible in $D_1 \vee L_1$ for σ, $\neg L_2$ is eligible in $D_2 \vee \neg L_2$ for σ, and L_1 is eligible in $D_3 \vee L_1 \vee L_2$ for σ. By $\mathrm{Res}(C,O)$ we denote the set of all clauses derivable from clause C with clauses in O using the inference rules above. Derivations are constructed according to the *derivation rules* below:

$$\langle \mathcal{U},\mathcal{I},\mathcal{S},\mathcal{E}\rangle \Longrightarrow \mathcal{N}|\emptyset|\emptyset \text{ where } \mathcal{N} = \mathcal{U} \cup \mathcal{I} \cup \mathcal{S} \cup \{P(c^L) \supset \bigcirc Q(c^L) \mid P(x) \to \bigcirc Q(x) \in \mathcal{S}, \Diamond L(x) \in \mathcal{E}\} \cup \{s_0^{L,1} \supset \bigcirc L(c^L \mid \Diamond L(x) \in \mathcal{E}\} \cup \{s_0^{L,0} \supset \bigcirc L \mid \Diamond L \in \mathcal{E}\}$$

$$\emptyset|\mathcal{P} \cup \{s_i^{L,k} \to \bigcirc\bot\}|O \Longrightarrow \{\bot\}|\mathcal{P}|O \cup \{s_i^{L,k} \to \bigcirc\bot\} \qquad \text{for some } i,k,L$$

$$\emptyset|\mathcal{P} \cup \{s_i^{L,1} \wedge C \to \bigcirc\bot\}|O \Longrightarrow \{s_{i+1}^{L,1} \to \bigcirc\neg C \vee L(c^l)\}|\mathcal{P}|O \cup \{s_i^{L,1} \wedge C \to \bigcirc\bot\} \qquad \text{for some } i,L \text{ and } C \neq \emptyset$$

$$\emptyset|\mathcal{P} \cup \{s_i^{L,0} \wedge C \to \bigcirc\bot\}|O \Longrightarrow \{s_{i+1}^{L,0} \to \bigcirc\neg C \vee L\}|\mathcal{P}|O \cup \{s_i^{L,0} \wedge C \to \bigcirc\bot\} \qquad \text{for some } i,L \text{ and } C \neq \emptyset$$

$$\emptyset|\mathcal{P} \cup \{C\}|O \Longrightarrow \mathcal{N}|\mathcal{P}|O \cup \{C\} \qquad \text{if none of the previous rule applies and } \mathcal{N} = \mathrm{Res}(C,O)$$

$$\emptyset|\mathcal{P}|O \Longrightarrow \{\Box\forall x\neg H_{i+1}^L(x) \mid \text{for all } i, L \text{ with } \models \forall x(H_i^L(x) \Leftrightarrow H_{i+1}^L(x))\}|\mathcal{P}|O \quad \text{where } H_i^L(x) := \bigvee\{(\bar{\exists} C_j)\{c^l/x\} \mid s_i^{L,1} \wedge C_j \to \bigcirc\bot \in \mathcal{P} \cup O\} \text{ for all } i, L$$

$$\emptyset|\mathcal{P}|O \Longrightarrow \{\Box\neg H_{i+1}^L \mid \text{for all } i, L \text{ with } \models (H_i^L \Leftrightarrow H_{i+1}^L)\}|\mathcal{P}|O \quad \text{where } H_i^L := \bigvee\{C_j \mid s_i^{L,0} \wedge C_j \to \bigcirc\bot \in \mathcal{P} \cup O\} \text{ for all } i, L$$

$$\mathcal{N} \cup \{C\}|\mathcal{P}|O \Longrightarrow \mathcal{N}|\mathcal{P} \cup \{C\}|O \qquad \text{if none of the previous rules applies}$$

Clarifications: To determine the satisfiability of a formula φ of monodic first-order discrete linear time temporal logic over expanding domains, the formula φ is transformed into a *Clausal Monodic Temporal Problem* $\mathcal{P} = \langle \mathcal{U},\mathcal{I},\mathcal{S},\mathcal{E}\rangle$ where $\mathcal{U}$ and $\mathcal{I}$ are sets of first-order clauses; $\mathcal{S}$ is a set of *step clauses* of the form $p \to \bigcirc q$, where p and q are propositions, or $P(t) \to \bigcirc Q(t)$, where P and Q are unary predicates and t is either a variable or a constant; $\mathcal{E}$ is a set of *eventuality clauses* of the form $\Diamond L$, where L is a propositional literal, $\Diamond(\neg)P(t)$, where P is a unary predicate and t is either a constant or a variable. Inferences are restricted by an *ordering* $\succ$ on literals and a subsumption-compatible *selection function* S. S maps any first-order clause C to a possibly empty subset of its negative literals. A literal L is *eligible* in a clause $L \vee C$ for a substitution σ if either $L \in S(C)$, or $S(C) = \emptyset$ and $L\sigma$ is $\succ$-maximal w.r.t. $C\sigma$.

History: Introduced in [1] and related to {57}. It has been implemented in the prover TSPASS [3]. Satisfiability equivalence of the transformation, soundness and refutational completeness are shown in [2].

---

[1] M. Ludwig and U. Hustadt. "Fair Derivations in Monodic Temporal Reasoning". In: *CADE-22*. Vol. 5663. LNCS. Springer, 2009, pp. 261–276. DOI: 10.1007/978-3-642-02959-2_21.

[2] M. Ludwig. "Advancing Formal Verification: Resolution-Based Methods for Linear-Time Temporal Logics". PhD thesis. University of Liverpool, UK, 2010.

[3] M. Ludwig and U. Hustadt. "Implementing a fair monodic temporal logic prover". In: *AI Communications* 23.2–3 (2010), pp. 69–96. DOI: 10.3233/AIC-2010-0457.

Entry 70 by: Ullrich Hustadt

Resolution for Computation Tree Logic (CTL) (2009)

$$\frac{\mathbf{A}\,\Box(\mathbf{start} \to (C \vee l)) \quad \mathbf{A}\,\Box(\mathbf{start} \to (D \vee \neg l))}{\mathbf{A}\,\Box(\mathbf{start} \to (C \vee D))} \qquad \frac{\mathbf{A}\,\Box(P \to \mathbf{E}_{\langle ind\rangle}\bigcirc(C \vee l)) \quad \mathbf{A}\,\Box(Q \to \mathbf{E}_{\langle ind\rangle}\bigcirc(D \vee \neg l))}{\mathbf{A}\,\Box(P \wedge Q \to \mathbf{E}_{\langle ind\rangle}\bigcirc(C \vee D))}$$

$$\frac{\mathbf{A}\,\Box(\mathbf{true} \to (C \vee l)) \quad \mathbf{A}\,\Box(\mathbf{start} \to (D \vee \neg l))}{\mathbf{A}\,\Box(\mathbf{start} \to (C \vee D))} \qquad \frac{\mathbf{A}\,\Box(P \to \mathbf{A}\bigcirc(C \vee l)) \quad \mathbf{A}\,\Box(Q \to \mathbf{A}\bigcirc(D \vee \neg l))}{\mathbf{A}\,\Box(P \wedge Q \to \mathbf{A}\bigcirc(C \vee D))}$$

$$\frac{\mathbf{A}\,\Box(\mathbf{true} \to (C \vee l)) \quad \mathbf{A}\,\Box(\mathbf{true} \to (D \vee \neg l))}{\mathbf{A}\,\Box(\mathbf{true} \to (C \vee D))} \qquad \frac{\mathbf{A}\,\Box(P \to \mathbf{A}\bigcirc(C \vee l)) \quad \mathbf{A}\,\Box(Q \to \mathbf{E}_{\langle ind\rangle}\bigcirc(D \vee \neg l))}{\mathbf{A}\,\Box(P \wedge Q \to \mathbf{E}_{\langle ind\rangle}\bigcirc(C \vee D))}$$

$$\frac{\mathbf{A}\,\Box(\mathbf{true} \to (C \vee l)) \quad \mathbf{A}\,\Box(P \to \mathbf{A}\bigcirc(D \vee \neg l))}{\mathbf{A}\,\Box(P \to \mathbf{A}\bigcirc(C \vee D))} \qquad \frac{\mathbf{A}\,\Box(\mathbf{true} \to (C \vee l)) \quad \mathbf{A}\,\Box(P \to \mathbf{E}_{\langle ind\rangle}\bigcirc(D \vee \neg l))}{\mathbf{A}\,\Box(P \to \mathbf{E}_{\langle ind\rangle}\bigcirc(C \vee D))}$$

$$\frac{\mathbf{A}\,\Box(P^{\dagger} \to \mathbf{E}\bigcirc\mathbf{E}\,\Box\, l) \quad \mathbf{A}\,\Box(D \to \mathbf{A}\Diamond\neg l)}{\mathbf{A}\,\Box(Q \to \mathbf{A}(\neg P^{\dagger}\,\mathcal{W}\, l))}\ \textit{(ERES1)} \qquad \frac{\mathbf{A}\,\Box(P^{\dagger} \to \mathbf{E}_{\langle ind\rangle}\bigcirc\mathbf{E}_{\langle ind\rangle}\,\Box\, l) \quad \mathbf{A}\,\Box(Q \to \mathbf{E}_{\langle ind\rangle}\Diamond\neg l)}{\mathbf{A}\,\Box(Q \to \mathbf{E}_{\langle ind\rangle}(\neg P^{\dagger}\,\mathcal{W}\,\neg l))}\ \textit{(ERES2)}$$

$$\mathbf{A}\,\Box(P \to \mathbf{A}\bigcirc\mathbf{false}) \Rightarrow \mathbf{A}\,\Box(\mathbf{true} \to \neg P) \qquad \mathbf{A}\,\Box(P \to \mathbf{E}_{\langle ind\rangle}\bigcirc\mathbf{false}) \Rightarrow \mathbf{A}\,\Box(\mathbf{true} \to \neg P)$$

where l is a literal, C and D are (possibly empty) disjunctions of literals, P and Q are (possibly empty) conjunctions of literals, $P^{\dagger} \to \varphi$, $\varphi \in \{\mathbf{E}\bigcirc\mathbf{E}\,\Box\, l, \mathbf{E}_{\langle ind\rangle}\bigcirc\mathbf{E}_{\langle ind\rangle}\,\Box\, l\}$, represents a set of clauses that together imply φ.

Derivations terminate if either $\mathbf{A}\,\Box(\mathbf{start} \to \mathbf{false})$ or $\mathbf{A}\,\Box(\mathbf{true} \to \mathbf{false})$ is derived (unsatisfiable) or no new clause can be derived (satisfiable).

Clarifications: This calculus is for propositional computation tree logic [1]. Formulae are first translated, in a satisfiability preserving way, into the following normal form: $\mathbf{A}\,\Box(\mathbf{start} \to A)$, an *initial clause*, $\mathbf{A}\,\Box(\mathbf{true} \to A)$, a *global clause*, $\mathbf{A}\,\Box(P \to \mathbf{P}\bigcirc C)$, a *P-step clause*, and $\mathbf{A}\,\Box(P \to \mathbf{P}\Diamond l)$, a $\mathbf{P}$*-eventuality clause* where $\mathbf{P}$ represents either $\mathbf{A}$ or $\mathbf{E}_{\langle ind\rangle}$. Indices ind attached to existential path operators $\mathbf{E}_{\langle ind\rangle}$ are elements of an arbitrary enumerable set Ind; they are introduced during the transformation to normal form to represent a particular path and are used to preserve satisfiability. The logical constant $\mathbf{start}$ only holds in the first moment in time. In the rules, $\mathbf{true}$ stands for an empty conjunction of literals, $\mathbf{false}$ stands for an empty disjunction of literals. The resolvents of the ERES rules need transformation into the normal form.

History: The calculus was first presented in [3]. It removes two redundant rules from an earlier resolution calculus for CTL [2]. Full details, including a normal form transformation and algorithms for the application of ERES1 and ERES2, are provided in [5]. The calculus has been implemented in the prover CTL-RP [4].

Remarks: Soundness, completeness and termination are shown in [5].

[1] E. M. Clarke and E. A. Emerson. "Design and synthesis of synchronisation skeletons using branching time temporal logic". In: *Proc. Logic of Programs*. Vol. 131. LNCS. Springer, 1981, pp. 52–71.

[2] A. Bolotov and M. Fisher. "A Clausal Resolution Method for CTL Branching Time Temporal Logic". In: *Journal of Experimental and Theoretical Artificial Intelligence* 11 (1999), pp. 77–93.

[3] L. Zhang, U. Hustadt, and C. Dixon. "A Refined Resolution Calculus for CTL". In: *CADE-22*. Ed. by Renate A. Schmidt. Vol. 5663. LNCS. Springer, 2009, pp. 245–260.

[4] L. Zhang, U. Hustadt, and C. Dixon. "CTL-RP: A computation tree logic resolution prover". In: *AI Communications* 23.2–3 (2010), pp. 111–136. DOI: 10.3233/AIC-2010-0463.

[5] L. Zhang, U. Hustadt, and C. Dixon. "A Resolution Calculus for Branching-Time Temporal Logic CTL". In: *ACM Transactions on Computational Logic* 15.1 (2014), 10:1–38. DOI: 10.1145/2529993.

Entry 71 by: Clare Dixon, Ullrich Hustadt

Untyped λ Reduction (2011)

Term-Sequent Rules

$$\frac{}{\Gamma \vdash t :\!\!- t, \Psi, \Delta}\ (Ax_{:-}) \qquad \frac{\Gamma \vdash \Theta :\!\!- s, \Psi, \Delta \quad \Gamma \vdash \langle \ldots s \ldots \rangle :\!\!- t, \Psi, \Delta}{\Gamma \vdash \langle \ldots \Theta \ldots \rangle :\!\!- t, \Psi, \Delta}\ (Cut_\lambda)$$

$$\frac{\Gamma \vdash \langle \ldots \langle t_1, t_2 \rangle \ldots \rangle :\!\!- t, \Psi, \Delta}{\Gamma \vdash \langle \ldots t_1 \cdot t_2 \ldots \rangle :\!\!- t, \Psi, \Delta}\ (\cdot L) \qquad \frac{\Gamma \vdash \Theta_1 :\!\!- t_1, \Psi, \Delta \quad \Gamma \vdash \Theta_2 :\!\!- t_2, \Psi, \Delta}{\Gamma \vdash \langle \Theta_1, \Theta_2 \rangle :\!\!- t_1 \cdot t_2, \Psi, \Delta}\ (\cdot R)$$

$$\frac{\Gamma \vdash \Theta :\!\!- t_1, \Psi, \Delta \quad \Gamma \vdash \langle \ldots t_2 \ldots \rangle :\!\!- t, \Psi, \Delta}{\Gamma, t_1 \rightsquigarrow t_2 \vdash \langle \ldots \Theta \ldots \rangle :\!\!- t, \Psi, \Delta}\ (\rightsquigarrow L) \qquad \frac{\Gamma \vdash t_1 :\!\!- t_2, \Psi, \Delta}{\Gamma \vdash t_1 \rightsquigarrow t_2, \Psi, \Delta}\ (\rightsquigarrow R)$$

$$\frac{\Gamma \vdash \Theta :\!\!- t_1, \Psi, \Delta \quad \Gamma \vdash \langle \ldots t_2[x/t_1] \ldots \rangle :\!\!- t, \Psi, \Delta}{\Gamma \vdash \langle \ldots \langle \lambda x.t_2, \Theta \rangle \ldots \rangle :\!\!- t, \Psi, \Delta}\ (\lambda L) \qquad \frac{\Gamma \vdash \langle \Theta, y \rangle :\!\!- t[x/y], \Psi, \Delta}{\Gamma \vdash \Theta :\!\!- \lambda x.t, \Psi, \Delta}\ (\lambda R)$$

(Classical) Sequent Rules

$$\frac{\Gamma \vdash \Psi, A, \Delta \quad \Gamma, A \vdash \Psi, \Delta}{\Gamma \vdash \Psi, \Delta}\ (Cut)$$

$$\frac{\Gamma, A[x/t] \vdash \Psi, \Delta}{\Gamma, \forall x.A \vdash \Psi, \Delta}\ (\forall L) \qquad \frac{\Gamma \vdash \Psi, A[x/y], \Delta}{\Gamma \vdash \Psi, \forall x.A, \Delta}\ (\forall R)$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Psi, \Delta}{\Gamma, A \to B \vdash \Psi, \Delta}\ (\to L) \qquad \frac{\Gamma, A \vdash \Psi, B, \Delta}{\Gamma \vdash \Psi, A \to B, \Delta}\ (\to R) \qquad \frac{}{\Gamma, \bot \vdash \Psi, \Delta}\ (\bot)$$

Clarifications: A *term-sequent* is a pair $\Theta :\!\!- t$ where Θ is a tree and t is a term. Define *trees* by: $\Theta ::= t \mid \langle \Theta_1, \Theta_2 \rangle$. If Θ' occurs as a *subtree* of Θ then we write Θ as $\langle \ldots \Theta' \ldots \rangle$. A *sequent* has the form $\Gamma \vdash \Psi, \Delta$ where Γ and Δ are sets of formulae and Ψ is a set of term-sequents. y must not be free in the lower term-sequent of (λR) nor the lower sequent of $(\forall R)$.

Remarks: Cut elimination — for both (Cut) and (Cut_λ) — is proved in [2], as is soundness and completeness of term-sequents with respect to the calculus of untyped lambda reduction with β-reduction and η-expansion. Soundness and completeness of the full calculus is shown for an axiomatic presentation of a stronger system in [1] (using a model theory of lambda reduction similar to *graph models* which is expanded further in [3, 4]).

[1] Michael J. Gabbay and Murdoch J. Gabbay. "A simple class of Kripke-style models in which logic and computation have equal standing". In: *International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR)*. Vol. 6355. Lecture Notes in Computer Science. Springer, 2010, pp. 231–254. DOI: http://dx.doi.org/10.1007/978-3-642-17511-4_14.

[2] Michael Gabbay. "A proof-theoretic treatment of λ-reduction with cut-elimination: λ-calculus as a logic programming language". In: *Journal of Symbolic Logic* 76.2 (2011), pp. 673–699.

[3] Michael J. Gabbay and Murdoch J. Gabbay. "A Simple and Complete Model Theory for Intensional and Extensional Untyped Lambda-Equality". In: *IFCoLog Journal of Logic and its Applications* 1.2 (2014).

[4] Murdoch J. Gabbay and Michael Gabbay. "Representation and duality of the untyped λ-calculus in nominal lattice and topological semantics, with a proof of topological completeness". In: *Annals of Pure and Applied Logic* (2016). DOI: http://dx.doi.org/10.1016/j.apal.2016.10.001.

Entry 72 by: Michael Gabbay

Sequent Calculi for Paraconsistent Logics (2012)

$$(\mathbf{c})\ \frac{\Gamma,\varphi\Rightarrow\Delta}{\Gamma,\neg\neg\varphi\Rightarrow\Delta} \qquad (\mathbf{e})\ \frac{\Gamma\Rightarrow\Delta,\varphi}{\Gamma\Rightarrow\Delta,\neg\neg\varphi} \qquad (\mathbf{i})\ \frac{\Gamma,\varphi,\neg\varphi\Rightarrow\Delta}{\Gamma,\neg\circ\varphi\Rightarrow\Delta}$$

$$(\mathbf{n}^{\mathbf{r}}_{\wedge})\ \frac{\Gamma\Rightarrow\Delta,\neg\psi,\neg\varphi}{\Gamma\Rightarrow\Delta,\neg(\varphi\wedge\psi)} \qquad (\mathbf{n}^{\mathbf{r}}_{\vee})\ \frac{\Gamma\Rightarrow\Delta,\neg\varphi \quad \Gamma\Rightarrow\Delta,\neg\psi}{\Gamma\Rightarrow\Delta,\neg(\varphi\vee\psi)} \qquad (\mathbf{n}^{\mathbf{r}}_{\supset})\ \frac{\Gamma\Rightarrow\Delta,\varphi \quad \Gamma\Rightarrow\Delta,\neg\psi}{\Gamma\Rightarrow\Delta,\neg(\varphi\supset\psi)}$$

$$(\mathbf{n}^{\mathbf{l}}_{\wedge})\ \frac{\Gamma,\neg\varphi\Rightarrow\Delta \quad \Gamma,\neg\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\wedge\psi)\Rightarrow\Delta} \qquad (\mathbf{n}^{\mathbf{l}}_{\vee})\ \frac{\Gamma,\neg\varphi,\neg\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad (\mathbf{n}^{\mathbf{l}}_{\supset})\ \frac{\Gamma,\varphi,\neg\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}$$

$$(\mathbf{a}_{\vee})\ \frac{\dfrac{\Gamma,\neg\varphi\Rightarrow\Delta \quad \Gamma,\neg\psi,\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad \dfrac{\Gamma,\neg\psi\Rightarrow\Delta \quad \Gamma,\neg\varphi,\varphi\Rightarrow\Delta}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta}}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad (\mathbf{a}_{\supset})\ \frac{\dfrac{\Gamma,\varphi\Rightarrow\Delta \quad \Gamma,\neg\psi,\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta} \qquad \dfrac{\Gamma,\neg\varphi,\varphi\Rightarrow\Delta \quad \Gamma,\neg\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}$$

$$(\mathbf{o}^{\mathbf{1}}_{\wedge})\ \frac{\Gamma,\neg\varphi\Rightarrow\Delta \quad \Gamma\Rightarrow\Delta,\psi}{\Gamma,\neg(\varphi\wedge\psi)\Rightarrow\Delta} \qquad (\mathbf{o}^{\mathbf{1}}_{\vee})\ \frac{\dfrac{\Gamma,\neg\varphi\Rightarrow\Delta}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad \dfrac{\Gamma,\varphi\Rightarrow\Delta \quad \Gamma\Rightarrow\Delta,\psi}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta}}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad (\mathbf{o}^{\mathbf{1}}_{\supset})\ \frac{\dfrac{\Gamma,\neg\varphi\Rightarrow\Delta \quad \Gamma\Rightarrow\Delta,\psi}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta} \qquad \dfrac{\Gamma,\varphi\Rightarrow\Delta}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}$$

$$(\mathbf{o}^{\mathbf{2}}_{\wedge})\ \frac{\Gamma,\neg\psi\Rightarrow\Delta \quad \Gamma\Rightarrow\Delta,\varphi}{\Gamma,\neg(\varphi\wedge\psi)\Rightarrow\Delta} \qquad (\mathbf{o}^{\mathbf{2}}_{\vee})\ \frac{\dfrac{\Gamma,\neg\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad \dfrac{\Gamma,\psi\Rightarrow\Delta \quad \Gamma\Rightarrow\Delta,\varphi}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta}}{\Gamma,\neg(\varphi\vee\psi)\Rightarrow\Delta} \qquad (\mathbf{o}^{\mathbf{2}}_{\supset})\ \frac{\dfrac{\Gamma,\neg\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta} \qquad \dfrac{\Gamma,\varphi\Rightarrow\Delta \quad \Gamma,\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}$$

Clarifications: Let B be a subset of the above rules, that does not contain any of the pairs: $(\mathbf{o}^{\mathbf{1}}_{\wedge})$, $(\mathbf{n}^{\mathbf{r}}_{\wedge})$; $(\mathbf{o}^{\mathbf{2}}_{\wedge})$, $(\mathbf{n}^{\mathbf{r}}_{\wedge})$; $(\mathbf{o}^{\mathbf{1}}_{\vee})$, $(\mathbf{n}^{\mathbf{r}}_{\vee})$; $(\mathbf{o}^{\mathbf{2}}_{\vee})$,$(\mathbf{n}^{\mathbf{r}}_{\vee})$; and $(\mathbf{o}^{\mathbf{1}}_{\supset})$, $(\mathbf{n}^{\mathbf{r}}_{\supset})$. $\mathbf{G_{BK}}[B]$ is obtained from propositional $\mathbf{LK}$ $\{6\}$ by deleting $\neg l$, and adding the rules $R(x)$ for every $x\in B$, as well as the rules: $(\circ\Rightarrow)$ $\dfrac{\Gamma\Rightarrow\varphi,\Delta \quad \Gamma\Rightarrow\neg\varphi,\Delta}{\Gamma,\circ\varphi\Rightarrow\Delta}$ and $(\Rightarrow\circ)$ $\dfrac{\Gamma,\varphi,\neg\varphi\Rightarrow\Delta}{\Gamma\Rightarrow\circ\varphi,\Delta}$.

History: One of the most important families of da Costa's Brazilian School of paraconsistency is that of C-systems [1]. The family of logics induced by the calculi above includes every C-system ever studied in the literature. All calculi were given in [2], where well-known axioms for paraconsistent logics were translated to sequent rules.

Remarks: All calculi above enjoy cut-admissibility.

[1] Walter A. Carnielli and João Marcos. "A taxonomy of **C**-systems". In: *Paraconsistency: The logical way to the inconsistent*. Ed. by W. A. Carnielli, M. E. Coniglio, and I. M. L. D'Ottaviano. Vol. 228. Lecture Notes in Pure and Applied Mathematics. Marcel Dekker, 2002, pp. 1–94.

[2] Arnon Avron, Beata Konikowska, and Anna Zamansky. "Efficient reasoning with inconsistent information using C-systems". In: *Information Sciences* 296 (2015), pp. 219–236.

Entry 73 by: Anna Zamansky, Yoni Zohar

Counterfactual Sequent Calculi I (1983,1992,2012,2013)

$$\frac{\begin{array}{c}\{\, B_k \vdash A_1,\ldots,A_n,D_1,\ldots,D_m \mid k \le n \,\} \\ \cup\, \{\, C_k \vdash A_1,\ldots,A_n,D_1,\ldots,D_{k-1} \mid k \le m \,\}\end{array}}{\Gamma,(C_1 \preccurlyeq D_1),\ldots,(C_m \preccurlyeq D_m) \vdash \Delta,(A_1 \preccurlyeq B_1),\ldots,(A_n \preccurlyeq B_n)}\ R_{n,m}$$

$$\frac{\{\, C_k \vdash D_1,\ldots,D_{k-1} \mid k \le m \,\} \qquad \Gamma \vdash \Delta, D_1,\ldots,D_m}{\Gamma,(C_1 \preccurlyeq D_1),\ldots,(C_m \preccurlyeq D_m) \vdash \Delta}\ T_m$$

$$\frac{\{\, C_k \vdash A_1,\ldots,A_n,D_1,\ldots,D_{k-1} \mid k \le m \,\} \qquad \Gamma \vdash \Delta, A_1,\ldots,A_n,D_1,\ldots,D_m}{\Gamma,(C_1 \preccurlyeq D_1),\ldots,(C_m \preccurlyeq D_m) \vdash \Delta,(A_1 \preccurlyeq B_1),\ldots,(A_n \preccurlyeq B_n)}\ W_{n,m}$$

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta,(A \preccurlyeq B)}\ R_{C1} \qquad \frac{\begin{array}{c}\{\, \Gamma^{\preccurlyeq}, B_k \vdash \Delta^{\preccurlyeq}, A_1,\ldots,A_n,D_1,\ldots,D_m \mid k \le n \,\} \\ \cup\, \{\, \Gamma^{\preccurlyeq}, C_k \vdash \Delta^{\preccurlyeq}, A_1,\ldots,A_n,D_1,\ldots,D_{k-1} \mid k \le m \,\}\end{array}}{\Gamma,(C_1 \preccurlyeq D_1),\ldots,(C_m \preccurlyeq D_m) \vdash \Delta,(A_1,\preccurlyeq B_1),\ldots,(A_n \preccurlyeq B_n)}\ A_{n,m} \qquad \frac{\Gamma, A \vdash \Delta \qquad \Gamma \vdash \Delta, B}{\Gamma,(A \preccurlyeq B) \vdash \Delta}\ R_{C2}$$

$$\mathcal{R}_{\mathbb{V}_\preccurlyeq} = \{R_{n,m} \mid n \ge 1, m \ge 0\}$$

$$\begin{array}{ll} \mathcal{R}_{\mathbb{VN}_\preccurlyeq} = \{R_{n,m} \mid n+m \ge 1\} & \mathcal{R}_{\mathbb{VC}_\preccurlyeq} = \mathcal{R}_{\mathbb{V}} \cup \{R_{C1}, R_{C2}\} \\ \mathcal{R}_{\mathbb{VT}_\preccurlyeq} = \mathcal{R}_{\mathbb{V}_\preccurlyeq} \cup \{T_m \mid m \ge 1\} & \mathcal{R}_{\mathbb{VA}_\preccurlyeq} = \{A_{n,m} \mid n \ge 1, m \ge 0\} \\ \mathcal{R}_{\mathbb{VW}_\preccurlyeq} = \mathcal{R}_{\mathbb{V}_\preccurlyeq} \cup \{W_{n,m} \mid n+m \ge 1\} & \mathcal{R}_{\mathbb{VNA}_\preccurlyeq} = \{A_{n,m} \mid n+m \ge 1\} \end{array}$$

Clarifications: Sequents are based on multisets. The rules $\mathcal{R}_{\mathcal{L}_\preccurlyeq}$ form a calculus for a counterfactual logic $\mathcal{L}$ described in [1], where $\preccurlyeq$ is the *comparative plausibility* operator. Besides the rules shown above, these calculi also include the propositional rules of **G3c** {45} and contraction rules. The contexts $\Gamma^{\preccurlyeq}$ and $\Delta^{\preccurlyeq}$ contain all formulae of resp. Γ and Δ of the form $A \preccurlyeq B$.

History: The calculus for $\mathbb{VC}$ was introduced in the tableaux setting [2, 3]. The remaining calculi were introduced in [4, 6] and corrected in [5].

Remarks: Soundness and completeness are shown by proving equivalence to Hilbert-style calculi and (syntactical) cut elimination. These calculi yield PSPACE decision procedures (EXPTIME for $\mathbb{VA}_\preccurlyeq$ and $\mathbb{VNA}_\preccurlyeq$) and, in most cases, enjoy Craig Interpolation. Contraction can be made admissible.

[1] David Lewis. *Counterfactuals*. Blackwell, 1973.

[2] Harrie C.M. de Swart. "A Gentzen- or Beth-Type System, a Practical Decision Procedure and a Constructive Completeness Proof for the Counterfactual Logics VC and VCS". In: *J. Symb. Log.* 48.1 (1983), pp. 1–20.

[3] Ian P. Gent. "A Sequent- or Tableau-style System for Lewis's Counterfactual Logic VC". In: *Notre Dame J. Form. Log.* 33.3 (1992), pp. 369–382.

[4] Björn Lellmann and Dirk Pattinson. "Sequent Systems for Lewis' Conditional Logics". In: *JELIA 2012*. Ed. by Luis Fariñas del Cerro, Andreas Herzig, and Jerome Mengin. Vol. 7519. LNCS. Springer-Verlag Berlin Heidelberg, 2012, pp. 320–332.

[5] Björn Lellmann. "Sequent Calculi with Context Restrictions and Applications to Conditional Logic". PhD thesis. Imperial College London, 2013. URL: `http://hdl.handle.net/10044/1/18059`.

[6] Björn Lellmann and Dirk Pattinson. "Constructing Cut Free Sequent Systems With Context Restrictions Based on Classical or Intuitionistic Logic". In: *ICLA 2013*. Ed. by Kamal Lodaya. Vol. 7750. LNAI. Springer-Verlag Berlin Heidelberg, 2013, pp. 148–160.

Entry 74 by: Björn Lellmann

Counterfactual Sequent Calculi II (2012, 2013)

$$\frac{\{\, C_k, \mathbf{B}^I \vdash \mathbf{A}^{[n]\setminus I}, \mathbf{C}^J, \mathbf{D}^{[k-1]\setminus J} \mid 1 \le k \le m, I \subseteq [n], J \subseteq [k-1] \,\} \\ \cup \{\, A_k, B_k, \mathbf{B}^I \vdash \mathbf{A}^{[n]\setminus I}, \mathbf{C}^J, \mathbf{D}^{[m]\setminus J} \mid k \le n, I \subseteq [n], J \subseteq [m] \,\}}{\Gamma, (A_1 \boxdot\!\!\Rightarrow B_1), \ldots, (A_n \boxdot\!\!\Rightarrow B_n) \vdash \Delta, (C_1 \boxdot\!\!\Rightarrow D_1), \ldots, (C_m \boxdot\!\!\Rightarrow D_m)}\ R_{n,m}$$

$$\frac{\{\, \Gamma \vdash \Delta, \mathbf{C}^J, \mathbf{D}^{[m]\setminus J} \mid J \subseteq [m] \,\} \cup \{\, C_k \vdash D_k, \mathbf{C}^J, \mathbf{D}^{[k-1]\setminus J} \mid 1 \le k \le m, J \subseteq [k-1] \,\}}{\Gamma \vdash \Delta, (C_1 \boxdot\!\!\Rightarrow D_1), \ldots, (C_m \boxdot\!\!\Rightarrow D_m)}\ T_m$$

$$\frac{\{\, C_k, \mathbf{B}^I \vdash \mathbf{A}^{[n]\setminus I}, \mathbf{C}^J, \mathbf{D}^{[k-1]\setminus J} \mid 1 \le k \le m, I \subseteq [n], J \subseteq [k-1] \,\} \\ \cup \{\, \Gamma, \mathbf{B}^I \vdash \mathbf{A}^{[n]\setminus I}, \mathbf{C}^J, \mathbf{D}^{[m]\setminus J} \mid I \subseteq [n], J \subseteq [m] \,\}}{\Gamma, (A_1 \boxdot\!\!\Rightarrow B_1), \ldots, (A_n \boxdot\!\!\Rightarrow B_n) \vdash \Delta, (C_1 \boxdot\!\!\Rightarrow D_1), \ldots, (C_m \boxdot\!\!\Rightarrow D_m)}\ W_{n,m}$$

$$\frac{\Gamma \vdash \Delta, A \qquad \Gamma, B \vdash \Delta}{\Gamma, (A \boxdot\!\!\Rightarrow B) \vdash \Delta}\ R_{C1} \qquad \frac{\Gamma \vdash \Delta, A \qquad \Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, (A \boxdot\!\!\Rightarrow B)}\ R_{C2}$$

For $n > 0$ the set $[n]$ is $\{1, \ldots, n\}$ and $[0]$ is $\emptyset$. For a set I of indices, $\mathbf{A}^I$ contains all A_i with $i \in I$.

$$\mathcal{R}_{\mathbb{V}_{\boxdot\!\!\Rightarrow}} = \{R_{n,m} \mid n \ge 1, m \ge 0\}$$

$$\mathcal{R}_{\mathbb{VN}_{\boxdot\!\!\Rightarrow}} = \{R_{n,m} \mid n+m \ge 1\} \qquad \mathcal{R}_{\mathbb{VW}_{\boxdot\!\!\Rightarrow}} = \mathcal{R}_{\mathbb{VT}_{\boxdot\!\!\Rightarrow}} \cup \{W_{n,m} \mid n+m \ge 1\}$$

$$\mathcal{R}_{\mathbb{VT}_{\boxdot\!\!\Rightarrow}} = \mathcal{R}_{\mathbb{V}_{\boxdot\!\!\Rightarrow}} \cup \{T_m \mid m \ge 1\} \qquad \mathcal{R}_{\mathbb{VC}_{\boxdot\!\!\Rightarrow}} = \mathcal{R}_{\mathbb{V}_{\boxdot\!\!\Rightarrow}} \cup \{R_{C1}, R_{C2}\}$$

Clarifications: Sequents are based on multisets. The rules $\mathcal{R}_{\mathcal{L}_{\boxdot\!\!\Rightarrow}}$ form a calculus for a counterfactual logic $\mathcal{L}$ described in [1], where $\boxdot\!\!\Rightarrow$ is the *strong counterfactual implication* operator. Besides the rules shown above, these calculi also include the propositional rules of **G3c** {45} and contraction rules.

History: These calculi were introduced in [2] and corrected in [3].

Remarks: The calculi are translations of the calculi in {74} to the language with $\boxdot\!\!\Rightarrow$. They inherit cut elimination and yield PSPACE decision procedures. Contraction can be made admissible.

[1] David Lewis. *Counterfactuals*. Blackwell, 1973.

[2] Björn Lellmann and Dirk Pattinson. "Sequent Systems for Lewis' Conditional Logics". In: *JELIA 2012*. Ed. by Luis Fariñas del Cerro, Andreas Herzig, and Jerome Mengin. Vol. 7519. LNCS. Springer-Verlag Berlin Heidelberg, 2012, pp. 320–332.

[3] Björn Lellmann. *Sequent Calculi with Context Restrictions and Applications to Conditional Logic*. 2013. URL: http://hdl.handle.net/10044/1/18059.

Entry 75 by: Björn Lellmann

Conditional Nested Sequents $\mathcal{NS}$ (2012-2014)

$$\frac{}{\Gamma(P,\neg P)}\ (AX)_{P \text{ atomic}} \qquad \frac{}{\Gamma(\top)}\ (AX_{\top}) \qquad \frac{}{\Gamma(\neg\bot)}\ (AX_{\bot})$$

$$\frac{\Gamma(A)}{\Gamma(\neg\neg A)}\ (\neg) \qquad \frac{\Gamma(\neg(A \Rightarrow B),[A' : \Delta,\neg B]) \quad A,\neg A' \quad A',\neg A}{\Gamma(\neg(A \Rightarrow B),[A' : \Delta])}\ (\Rightarrow^{-}) \qquad \frac{\Gamma([A : B])}{\Gamma(A \Rightarrow B)}\ (\Rightarrow^{+})$$

$$\frac{\Gamma([A : \Delta,\neg A])}{\Gamma([A : \Delta])}\ (ID) \qquad \frac{\Gamma([A : \Delta,\Sigma],[B : \Sigma]) \qquad A,\neg B \qquad B,\neg A}{\Gamma([A : \Delta],[B : \Sigma])}\ (CEM)$$

$$\frac{\Gamma(\neg(A \Rightarrow B),A) \qquad \Gamma(\neg(A \Rightarrow B),\neg B)}{\Gamma(\neg(A \Rightarrow B))}\ (MP)$$

$$\frac{\Gamma,\neg(C \Rightarrow D),[A : \Delta,\neg D] \quad \Gamma,\neg(C \Rightarrow D),[A : C] \quad \Gamma,\neg(C \Rightarrow D),[C : A]}{\Gamma,\neg(C \Rightarrow D),[A : \Delta]}\ (CSO)$$

Clarifications: Conditional logics extend classical logic with formulas of the form $A \Rightarrow B$: intuitively, $A \Rightarrow B$ is true in a world x if B is true in the set of worlds where A is true and that are most similar to x. The calculi $\mathcal{NS}$ manipulate *nested* sequents, a generalization of ordinary sequent calculi where sequents are allowed to occur within sequents. A nested sequent $\Gamma = A_1,\ldots,A_m,[B_1 : \Gamma_1],\ldots,[B_n : \Gamma_n]$ is inductively defined by the formula $\mathcal{F}(\Gamma) = A_1 \vee \cdots \vee A_m \vee (B_1 \Rightarrow \mathcal{F}(\Gamma_1)) \vee \cdots \vee (B_n \Rightarrow \mathcal{F}(\Gamma_n))$. $\Gamma(\Delta)$ represents a sequent Γ containing a *context* (a unique empty position) filled by the (nested) sequent Δ. Besides the rules shown above, the calculi $\mathcal{NS}$ also include standard rules for propositional connectives.

History: The calculi $\mathcal{NS}$ have been introduced in [1] and extended in [2]. The theorem prover NESCOND, implementing $\mathcal{NS}$ in Prolog, has been presented in [3].

Remarks: Completeness is a consequence of cut admissibility. $\mathcal{NS}$ calculi can be used to obtain a PSPACE decision procedure for the respective conditional logics (optimal for CK and extensions with ID and MP).

[1] Régis Alenda, Nicola Olivetti, and Gian Luca Pozzato. "Nested Sequent Calculi for Conditional Logics". In: *Logics in Artificial Intelligence - 13th European Conference, JELIA 2012*. Ed. by Luis Farinas del Cerro, Andreas Herzig, and Jérôme Mengin. Vol. 7519. Lecture Notes in Artificial Intelligence LNAI. Toulouse, France: Springer, Sept. 2012, pp. 14–27. DOI: 10.1007/978-3-642-33353-8_2.

[2] Régis Alenda, Nicola Olivetti, and Gian Luca Pozzato. "Nested Sequent Calculi for Normal Conditional Logics". In: *Journal of Logic and Computation* (2013). DOI: doi:10.1093/logcom/ext034.

[3] Nicola Olivetti and Gian Luca Pozzato. "NESCOND: an Implementation of Nested Sequent Calculi for Conditional Logics". In: *Proceedings of IJCAR 2014 (7th International Joint Conference on Automated Reasoning)*. Ed. by Stephane Demri, Deepak Kapur, and Christoph Weidenbach. Vol. 8562. Lecture Notes in Artificial Intelligence LNAI. Vienna (Austria): Springer, July 2014, pp. 511–518.

Entry 76 by: Nicola Olivetti, Gian Luca Pozzato

FILL Deep Nested Sequent Calculus (2013)

Propagation rules:

$$\frac{X[\mathcal{S} \Rightarrow (A, \mathcal{S}' \Rightarrow \mathcal{T}'), \mathcal{T}]}{X[\mathcal{S}, A \Rightarrow (\mathcal{S}' \Rightarrow \mathcal{T}'), \mathcal{T}]}\ pl_1 \qquad \frac{X[(\mathcal{S} \Rightarrow \mathcal{T}, A), \mathcal{S}' \Rightarrow \mathcal{T}']}{X[(\mathcal{S} \Rightarrow \mathcal{T}), \mathcal{S}' \Rightarrow A, \mathcal{T}']}\ pr_1$$

$$\frac{X[\mathcal{S}, A, (\mathcal{S}' \Rightarrow \mathcal{T}') \Rightarrow \mathcal{T}]}{X[\mathcal{S}, (\mathcal{S}', A \Rightarrow \mathcal{T}') \Rightarrow \mathcal{T}]}\ pl_2 \qquad \frac{X[\mathcal{S} \Rightarrow \mathcal{T}, A, (\mathcal{S}' \Rightarrow \mathcal{T}')]}{X[\mathcal{S} \Rightarrow \mathcal{T}, (\mathcal{S}' \Rightarrow \mathcal{T}', A)]}\ pr_2$$

Identity and logical rules: In branching rules, $X[\ ] \in X_1[\ ] \bullet X_2[\ ]$, $\mathcal{S} \in \mathcal{S}_1 \bullet \mathcal{S}_2$ and $\mathcal{T} \in \mathcal{T}_1 \bullet \mathcal{T}_2$.

$$\frac{X[\ ], \mathcal{U} \text{ and } \mathcal{V} \text{ are hollow.}}{X[\mathcal{U}, p \Rightarrow p, \mathcal{V}]}\ id^d \qquad \frac{X[\ ], \mathcal{U} \text{ and } \mathcal{V} \text{ are hollow.}}{X[\bot, \mathcal{U} \Rightarrow \mathcal{V}]}\ \bot_l^d \qquad \frac{X[\mathcal{S} \Rightarrow \mathcal{T}]}{X[\mathcal{S} \Rightarrow \mathcal{T}, \bot]}\ \bot_r^d$$

$$\frac{X[\mathcal{S} \Rightarrow \mathcal{T}]}{X[\mathcal{S}, \mathrm{I} \Rightarrow \mathcal{T}]}\ \mathrm{I}_l^d \qquad \frac{X[\ ], \mathcal{U} \text{ and } \mathcal{V} \text{ are hollow.}}{X[\mathcal{U} \Rightarrow \mathrm{I}, \mathcal{V}]}\ \mathrm{I}_r^d$$

$$\frac{X[\mathcal{S}, A, B \Rightarrow \mathcal{T}]}{X[\mathcal{S}, A \otimes B \Rightarrow \mathcal{T}]}\ \otimes_l^d \qquad \frac{X_1[\mathcal{S}_1 \Rightarrow A, \mathcal{T}_1] \quad X_2[\mathcal{S}_2 \Rightarrow B, \mathcal{T}_2]}{X[\mathcal{S} \Rightarrow A \otimes B, \mathcal{T}]}\ \otimes_r^d$$

$$\frac{X_1[\mathcal{S}_1 \Rightarrow A, \mathcal{T}_1] \quad X_2[\mathcal{S}_2, B \Rightarrow \mathcal{T}_2]}{X[\mathcal{S}, A \multimap B \Rightarrow \mathcal{T}]}\ \multimap_l^d \qquad \frac{X[\mathcal{S} \Rightarrow \mathcal{T}, (A \Rightarrow B)]}{X[\mathcal{S} \Rightarrow \mathcal{T}, A \multimap B]}\ \multimap_r^d$$

$$\frac{X_1[\mathcal{S}_1, A \Rightarrow \mathcal{T}_1] \quad X_2[\mathcal{S}_2, B \Rightarrow \mathcal{T}_2]}{X[\mathcal{S}, A \parr B \Rightarrow \mathcal{T}]}\ \parr_l^d \qquad \frac{X[\mathcal{S} \Rightarrow A, B, \mathcal{T}]}{X[\mathcal{S} \Rightarrow A \parr B, \mathcal{T}]}\ \parr_r^d$$

$$\frac{X[\mathcal{S}, (A \Rightarrow B) \Rightarrow \mathcal{T}]}{X[\mathcal{S}, A -\!\!< B \Rightarrow \mathcal{T}]}\ -\!\!<_l^d \qquad \frac{X_1[\mathcal{S}_1 \Rightarrow A, \mathcal{T}_1] \quad X_2[\mathcal{S}_2, B \Rightarrow \mathcal{T}_2]}{X[\mathcal{S} \Rightarrow A -\!\!< B, \mathcal{T}]}\ -\!\!<_r^d$$

Clarifications: Following Kashima [1], nested sequents are defined as below where A_i and B_j are formulae [2]:

$$S \quad T ::= S_1, \ldots, S_k, A_1, \ldots, A_m \Rightarrow B_1, \ldots, B_n, T_1, \ldots, T_l$$

Γ and Δ are multisets of formulae and P, Q, S, T, X, Y, etc., are nested sequents, and $\mathcal{S}$, $\mathcal{X}$, etc., are multisets of nested sequents and formulae.

Inference rules in BiILL$_{dn}$ are applied in a *context*, i.e., a nested sequent with a hole [ ]. Notice that BiILL$_{dn}$ contain no structural rules. The branching rules require operations to merge contexts and nested sequents, which are explained below. The zero-premise rules require that certain sequents or contexts are *hollow*, i.e., containing no occurrences of formulae.

The *merge set* $X_1 \bullet X_2$ of two sequents X_1 and X_2 is defined as:

$$\begin{aligned} X_1 \bullet X_2 = \{\ & (\Gamma_1, \Gamma_2, Y_1, \ldots, Y_m \Rightarrow \Delta_1, \Delta_2, Z_1, \ldots, Z_n) \mid \\ & X_1 = (\Gamma_1, P_1, \ldots, P_m \Rightarrow \Delta_1, Q_1, \ldots, Q_n) \text{ and} \\ & X_2 = (\Gamma_2, S_1, \ldots, S_m \Rightarrow \Delta_2, T_1, \ldots, T_n) \text{ and} \\ & Y_i \in P_i \bullet S_i \text{ for } 1 \le i \le m \text{ and } Z_j \in Q_j \bullet T_j \text{ for } 1 \le j \le n \ \} \end{aligned}$$

When $X \in X_1 \bullet X_2$, we say that X_1 and X_2 are a *partition* of X.

The merge set $X_1[\ ] \bullet X_2[\ ]$ of two contexts $X_1[\ ]$ and $X_2[\ ]$ is defined in [3]. If $X[\ ] = X_1[\ ] \bullet X_2[\ ]$ we say $X_1[\ ]$ and $X_2[\ ]$ are a *partition* of $X[\ ]$. The notion of a merge set between multisets of formulae and sequents is as follows. Given $\mathcal{X} = \Gamma \cup \{X_1, \ldots, X_n\}$ and $\mathcal{Y} = \Delta \cup \{Y_1, \ldots, Y_n\}$ their merge set contains all multisets of the form: $\Gamma \cup \Delta \cup \{Z_1, \ldots, Z_n\}$ where $Z_i \in X_i \bullet Y_i$.

Entry 77 by: Rajeev Goré

History: The sequent calculus arose from an attempt to give a display calculus for full intuitionistic linear logic (FILL). As usual for display calculi, a detour is necessary through an extension of FILL with an "exclusion" connective $-\!\!<$ which forms an adjunction with $\parr$. The resulting logic is called Bi-intuitionistic Linear Logic (BiILL). Although sound and complete for BiILL, the resulting display calculus is bad for backward proof search. Following Kashima [1], Alwen Tiu first obtained a shallow nested sequent calculus for BiILL, and then refined that into a deep nested sequent calculus for BiILL. The proof of cut-elimination for the shallow calculus, and the equivalence of the shallow and deep calculi requires over 615 different cases!

Remarks: The calculus shown is for Bi-Intuitionistic Linear Logic [2]. It is sound and complete. The soundness w.r.t. the categorical semantics is via the shallow nested sequent calculus for BiILL. A nested sequent is a (nested) *FILL-sequent* if it has no nesting of sequents on the left of $\Rightarrow$, and no occurrences of $-\!\!<$ at all. Only in the deep nested sequent calculus is it obvious that a derivation of a FILL-sequent encounters FILL-sequents only. The deep sequent calculus enjoys the subformula property and terminating backward-proof search. The validity problem for FILL is co-NP complete and BiILL is conservative over FILL [2]. All of these proofs were eventually formalised in Isabelle [3] by Jeremy Dawson. As far as is known, it is the only sequent calculus for FILL ({29}) that does not require (type) annotations.

---

[1] Ryo Kashima. "Cut-free sequent calculi for some tense logics". In: *Studia Logica* 53.1 (1994), pp. 119–136.

[2] Ranald Clouston, Jeremy E. Dawson, Rajeev Goré, and Alwen Tiu. "Annotation-Free Sequent Calculi for Full Intuitionistic Linear Logic". In: *Computer Science Logic (CSL)*. 2013, pp. 197–214.

[3] Jeremy E. Dawson, Ranald Clouston, Rajeev Goré, and Alwen Tiu. "From Display Calculi to Deep Nested Sequent Calculi: Formalised for Full Intuitionistic Linear Logic". In: *Proc. Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014*. 2014, pp. 250–264.

Contextual Natural Deduction (2013)

$$\frac{}{\Gamma, a : A \vdash a : A}$$

$$\frac{\Gamma, a : A \vdash b : C_\pi[B]}{\Gamma \vdash \lambda_\pi a^A.b : C_\pi[A \to B]} \to_I (\pi)$$

$$\frac{\Gamma \vdash f : C^1_{\pi_1}[A \to B] \quad \Gamma \vdash x : C^2_{\pi_2}[A]}{\Gamma \vdash (f\ x)^{\rightarrow}_{(\pi_1;\pi_2)} : C^1_{\pi_1}[C^2_{\pi_2}[B]]} \to^{\rightarrow}_E (\pi_1;\pi_2)$$

$$\frac{\Gamma \vdash f : C^1_{\pi_1}[A \to B] \quad \Gamma \vdash x : C^2_{\pi_2}[A]}{\Gamma \vdash (f\ x)^{\leftarrow}_{(\pi_1;\pi_2)} : C^2_{\pi_1}[C^1_{\pi_2}[B]]} \to^{\leftarrow}_E (\pi_1;\pi_2)$$

π, π_1 and π_2 must be positive positions. a is allowed to occur in b only if π is strongly positive.

Clarifications: $C_\pi[F]$ denotes a formula with F occurring in the hole of a *context* $C_\pi[]$. π is the position of the hole. It is: *positive* iff it is in the left side of an even number of implications; *strongly positive* iff this number is zero.

History: Contextual Natural Deduction [1] combines the idea of deep inference with Gentzen's natural deduction {5}.

Remarks: Soundness and completeness w.r.t. minimal logic are proven [1] by providing translations between $\mathbf{ND^c}$ and the minimal fragment of **NJ** {5}. $\mathbf{ND^c}$ proofs can be quadratically shorter than proofs in the minimal fragment of **NJ**.

---

[1] Bruno Woltzenlogel Paleo. "Contextual Natural Deduction". In: *Logical Foundations of Computer Science, International Symposium, LFCS 2013, San Diego, CA, USA, January 6-8, 2013. Proceedings*. Ed. by Sergei N. Artëmov and Anil Nerode. Vol. 7734. Lecture Notes in Computer Science. Springer, 2013, pp. 372–386. ISBN: 978-3-642-35721-3. DOI: 10.1007/978-3-642-35722-0_27. URL: http://dx.doi.org/10.1007/978-3-642-35722-0_27.

Entry 78 by: Bruno Woltzenlogel Paleo

IR (2014)

$$\frac{}{\left\{x^{\mathsf{restr}(\tau,x)} \mid x \in C, x \text{ is existential}\right\}} \text{ (Ax)}$$

C is a non-tautological clause from the matrix.
$\tau = \{0/u \mid u \text{ is universal in } C\}$, where the notation $0/u$ for literals u is shorthand for $0/y$ if $u = y$ and $1/y$ if $u = \neg y$. We define $\mathsf{restr}(\tau, x)$ as $\{c/u \mid c/u \in \tau, \mathsf{lv}(u) < \mathsf{lv}(x)\}$.

$$\frac{x^\tau \vee C_1 \qquad \neg x^\tau \vee C_2}{C_1 \cup C_2} \text{ (Resolution)}$$

$$\frac{C}{\left\{x^\xi \mid x^\sigma \in C, x \text{ is existential}\right\}} \text{ (Instantiation)}$$

τ is a partial assignment to universal variables with $\mathsf{rng}(\tau) \subseteq \{0,1\}$. $\xi = \sigma \cup \{c/u \mid c/u \in \mathsf{restr}(\tau, x), u \notin \mathsf{dom}(\sigma)\}$

The rules of IR [1]

Clarifications: The calculus aims to refute a quantified Boolean formula (QBF) of the form $Q_1 x_1 \dots Q_n x_n . \varphi$ where $Q_i \in \{\forall, \exists\}$ and φ is a Boolean formula in conjunctive normal form (CNF). The formula φ is referred to as the *matrix*. We write $\mathsf{lv}(x)$ for the *quantification level* of x, i.e. $\mathsf{lv}(x_i) = i$. A variable x_i is *existential* (resp. *universal*) if $Q_i = \exists$ (resp. $Q_i = \forall$).

The calculus works by introducing clauses as *annotated clauses*, which are sets of annotated literals. Annotated literals consist of an existential literal and an annotation – a partial assignment to universal variables in $\{0,1\}$. Two literals are identical if and only if both the existential literal and annotation are equal. The calculus enables deriving the empty clause if and only if the given formula is false.

Remarks: Soundness was shown by extracting valid Herbrand functions. Completeness is shown by p-simulation of another known QBF system Q-Resolution.

History: The name of the calculus comes from the two pivotal operations *instantiation* and *resolution*. The calculus naturally generalizes an older calculus ∀Exp+Res [2], which requires all clauses to be introduced into the proof by using a complete assignment.

[1] Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. "On Unification of QBF Resolution-Based Calculi". In: *Mathematical Foundations of Computer Science (MFCS)*. 2014.

[2] Mikoláš Janota and Joao Marques-Silva. "Expansion-based QBF solving versus Q-resolution". In: *Theoretical Computer Science* 577 (2015), pp. 25–42. DOI: http://dx.doi.org/10.1016/j.tcs.2015.01.048.

Entry 79 by: Leroy Chew, Mikoláš Janota

Sequent Calculus for Superintuitionistic Modal Logic (2014)

$$\top\mathrm{R}\ \frac{}{\Gamma \vdash \top, \Delta} \qquad \mathrm{id}\ \frac{}{\Gamma, \varphi \vdash \varphi, \Delta} \qquad \bot\mathrm{L}\ \frac{}{\Gamma, \bot \vdash \Delta}$$

$$\vee\mathrm{L}\ \frac{\Gamma, \varphi \vdash \Delta \qquad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta} \qquad \vee\mathrm{R}\ \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \vee \psi, \Delta}$$

$$\wedge\mathrm{L}\ \frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta} \qquad \wedge\mathrm{R}\ \frac{\Gamma \vdash \varphi, \Delta \qquad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \varphi \wedge \psi, \Delta}$$

$$\to\mathrm{L}\ \frac{\Gamma, \varphi \twoheadrightarrow \psi \vdash \varphi, \Delta \qquad \Gamma, \varphi \twoheadrightarrow \psi, \psi \vdash \Delta}{\Gamma, \varphi \to \psi \vdash \Delta} \qquad \to\mathrm{R}\ \frac{\Gamma, \varphi \vdash \psi, \Delta \qquad \Gamma \vdash \varphi \twoheadrightarrow \psi, \Delta}{\Gamma \vdash \varphi \to \psi, \Delta}$$

$$\textsc{step}\ \frac{\mathrm{Prem}_1 \quad \cdots \quad \mathrm{Prem}_k \quad \mathrm{Prem}_{k+1} \quad \cdots \quad \mathrm{Prem}_{k+n}}{\Sigma_l, \Theta^{\triangleright}, \Gamma^{\twoheadrightarrow} \vdash \Delta^{\twoheadrightarrow}, \Phi^{\triangleright}, \Sigma_r}\ \dagger$$

$$\mathrm{Prem}_{1 \le i \le k} = \Sigma_l, \Theta, \Theta^{\triangleright}, \Gamma^{\to}, \varphi_i \twoheadrightarrow \psi_i, \varphi_i \vdash \psi_i, \Delta^{\to}_{-i}, \Phi$$
$$\mathrm{Prem}_{k+1 \le i \le k+n} = \Sigma_l, \Theta, \Theta^{\triangleright}, \Gamma^{\to}, \triangleright\varphi_{i-k} \vdash \Delta^{\to}, \Phi$$

$\Theta^{\triangleright} = \triangleright\theta_1, \cdots, \triangleright\theta_j$ $\qquad$ $\Theta = \theta_1, \cdots, \theta_j$

$\Gamma^{\twoheadrightarrow} = \{\alpha_1 \twoheadrightarrow \beta_1, \cdots, \alpha_l \twoheadrightarrow \beta_l\}$ $\qquad$ $\Gamma^{\to} = \{\alpha_1 \to \beta_1, \cdots, \alpha_l \to \beta_l\}$

$\Delta^{\twoheadrightarrow} = \{\varphi_1 \twoheadrightarrow \psi_1, \cdots, \varphi_k \twoheadrightarrow \psi_k\}$ $\qquad$ $\Delta^{\to} = \{\varphi_1 \to \psi_1, \cdots, \varphi_k \to \psi_k\}$

$\Delta^{\to}_{-i} = \Delta^{\to} \setminus \{\varphi_i \to \psi_i\}$

$\Phi^{\triangleright} = \triangleright\varphi_1, \cdots, \triangleright\varphi_n$ $\qquad$ $\Phi = \varphi_1, \cdots, \varphi_n$

where † means that the conditions C0, C1 and C2 below must hold

(C0) $\Delta^{\twoheadrightarrow} \cup \Phi^{\triangleright} \neq \emptyset$

(C1) $\bot \notin \Sigma_l$ and $\top \notin \Sigma_r$ and $(\Sigma_l \cup \Theta^{\triangleright} \cup \Gamma^{\twoheadrightarrow}) \cap (\Delta^{\twoheadrightarrow} \cup \Phi^{\triangleright} \cup \Sigma_r) = \emptyset$

(C2) Σ_l and Σ_r each contain atomic formulae only

Explanations for the conditions:

(C0) there is at least one $\triangleright$- or $\twoheadrightarrow$-formula in the succedent of the conclusion

(C1) none of the rules $\bot$L, $\top$R, *id* are applicable to the conclusion

(C2) none of the rules $\vee$L, $\vee$R, $\wedge$L, $\wedge$R, $\to$L, $\to$R are applicable to the conclusion

Clarifications: There is a unary modal connective $\triangleright$ to be read as "later". Its semantics are box-like in terms of the underlying intuitionistic Kripke relation, which is irreflexive! There is also a new connective $\twoheadrightarrow$ corresponding to an irreflexive version of intuitionistic implication, which can be defined as $\triangleright(\varphi \to \psi)$. The $\to$L rule is **LK**-like {6} in that it is multi-conclusioned and has one branch for each subformula of $\varphi \wedge \psi$, but it also converts $\varphi \to \psi$ to $\varphi \twoheadrightarrow \psi$, read upwards, building in a form of contraction on such formulae. The $\to$R rule is unusual in that it has two premises: the left one is LK-like in that it does not delete Δ, read upwards, while the right one converts $\varphi \to \psi$ to $\varphi \twoheadrightarrow \psi$ read upwards building in a form of contraction on such formulae. The STEP rule has an indeterminate number of premises, one for each $\varphi_i \twoheadrightarrow \psi_i \in \Delta^{\twoheadrightarrow}$, and one for each $\varphi_i \in \Phi^{\triangleright}$. For each such "eventuality", the rule creates a premise that contains the subformula on an appropriate side,

Entry 80 by: Rajeev Goré

but also creates a copy of the principal formula in the antecedent of that premise, thus building in aspects of the standard sequent calculus for Göedel-Löb logic.

History: The superintuitionistic modal logic KMlin is obtained from Kuznetsov-Muravitsky logic KM [5] by demanding that the underlying Kripke frames be linear. The semantics of the unary modality $\triangleright$ becomes "true in all strict successors".

Clouston and Goré [4] defined this sequent calculus. Their rules are inspired by those of: Mauro Ferrarri, Camillo Fiorentini and Guido Fiorino for a sequent calculus with compartments for intuitionistic logic [3]; Giovanna Corsi for her sequent calculus for (quantified) Gödel-Dummett logic LC [1]; and George Boolos for his sequent calculus for Göedel-Löb logic GL [2].

Remarks: Clouston and Goré gave semantic proofs of soundness, cut-free completeness and the finite model property, thus giving decidability. They showed that the validity problem for this logic is coNP-complete. They also showed that all rules are invertible, so the sequent calculus can be used for backtrack-free and terminating backward proof search via the following strategy for rule applications: apply any applicable rule backwards, always preferring zero-premise rules if possible!

---

[1] Giovanna Corsi. "A cut-free calculus for Dummett's LC quantified". In: *Zeitschr. f. math. Logik und Grundlagen d. Math.* 35 (1989), pp. 289–301.

[2] George Boolos. *The logic of provability*. CUP, 1995.

[3] Mauro Ferrari, Camillo Fiorentini, and Guido Fiorino. "Contraction-Free Linear Depth Sequent Calculi for Intuitionistic Propositional Logic with the Subformula Property and Minimal Depth Counter-Models". In: *J. Autom. Reason.* 51.2 (2013), pp. 129–149.

[4] Ranald Clouston and Rajeev Goré. "Sequent Calculus in the Topos of Trees". In: *Proc. Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015*. 2015, pp. 133–147.

[5] Tadeusz Litak. "Constructive modalities with provability smack". Trends in Logic, to appear.

Erotetic Dual Resolution for Classical Propositional Logic (2014)

$$\frac{?(\Phi\,;\dashv S\,'\beta\,'T\,;\,\Psi)}{?(\Phi\,;\dashv S\,'\beta_1\,'T\,;\dashv S\,'\beta_2\,'T\,;\,\Psi)}\ \mathbf{R}_{\beta}$$

$$\frac{?(\Phi\,;\dashv S\,'\alpha\,'T\,;\,\Psi)}{?(\Phi\,;\dashv S\,'\alpha_1\,'\alpha_2\,'T\,;\,\Psi)}\ \mathbf{R}_{\alpha} \qquad \frac{?(\Phi\,;\dashv S\,'\neg\neg A\,'T\,;\,\Psi)}{?(\Phi\,;\dashv S\,'A\,'T\,;\,\Psi)}\ \mathbf{R}_{\neg\neg}$$

$$\frac{?(\Phi\,;\dashv S\,'A\,'T\,;\,\Psi\,;\dashv U\,'\overline{A}\,'V\,;\,\Omega)}{?(\dashv \underline{S}\,'\underline{T}\,'\underline{U}\,'\underline{V}\,;\,\Phi\,;\,\Psi\,;\,\Omega\,;\dashv S\,'A\,'T\,;\dashv U\,'\overline{A}\,'V)}\ \mathbf{R}_{res}$$

In $\mathbf{R}_{res}$: A and $\overline{A}$ must be complementary, that is either $A = \neg\overline{A}$ or $\overline{A} = \neg A$. $\underline{S}$, $\underline{T}$ are obtained from S, T by deleting all occurrences of A and $\underline{U}$, $\underline{V}$ are obtained from U, V by deleting all occurrences of $\neg A$. For the α, β notation see entry {58}.

Clarifications: The calculus is built in the framework of Inferential Erotetic Logic ([1]) and is designed to transform questions concerning refutability (falsifiability) of a formula. However, the rules act upon *reversed sequents*. The sequents are right-sided with sequences of formulas in the succedent. Φ, Ψ are finite (possibly empty) sequences of sequents. S, T are finite (possibly empty) sequences of formulas. The semicolon ';' is the concatenation sign for sequences of sequents, whereas ''' is the concatenation sign for sequences of formulas. The resolution rule is non-clausal (all the formulas in the premise, A included, may be compound) and dual with respect to the standard resolution (instead of CNF of $\neg A$, DNF of A is derived). A Socratic refutation of a sequent of the form '$\dashv A$' is a kind of a resolution refutation of $\neg A$ and it ends when the empty sequent (a counterpart of the empty clause) is arrived at.

History: The calculus $\mathbf{E}^{\mathsf{CPL}}_{res}$ has been presented in [2] together with extensions to some paraconsistent logics (see {82}). Compare also {58}.

Remarks: A formula A is CPL-valid iff $\dashv A$ has a Socratic refutation in $\mathbf{E}^{\mathsf{CPL}}_{res}$. Similar results are obtained with respect to CLuN, CLuNs and mbC.

---

[1] Andrzej Wiśniewski. *Questions, Inferences, and Scenarios*. College Publications, 2013.

[2] Szymon Chlebowski and Dorota Leszczyńska-Jasion. "Dual Erotetic Calculi and the Minimal LFI". In: *Studia Logica* (2015). doi: 10.1007/s11225-015-9617-0.

Entry 81 by: Dorota Leszczyńska-Jasion

Erotetic Dual Resolution for mbC (2014)

The rules of $\mathbf{E}^{\mathsf{CPL}}_{res}$ (see {81}) and the following rules ('$\neg$' is used for the classical negation and '$\sim$' for the paraconsistent one):

$$\frac{?(\Phi;\dashv S'\sim A'T;\Psi)}{?(\Phi;\dashv S'\neg A'T;\dashv S'\chi\sim A'T;\Psi)}\ \mathbf{R}_{\sim} \qquad \frac{?(\Phi;\dashv S'\neg\sim A'T;\Psi)}{?(\Phi;\dashv S'A'\neg\chi\sim A'T;\Psi)}\ \mathbf{R}_{\neg\sim}$$

$$\frac{?(\Phi;\dashv S'\circ A'T;\Psi)}{?(\Phi;\dashv S'\neg A'\chi\circ A'T;\dashv S'\neg\sim A'\chi\circ A'T;\Psi)}\ \mathbf{R}_{\circ}$$

$$\frac{?(\Phi;\dashv S'\neg\circ A'T;\Psi)}{?(\Phi;\dashv S'A'\sim A'T;\dashv S'\neg\chi\circ A'T;\Psi)}\ \mathbf{R}_{\neg\circ}$$

Clarifications: See {81} for notational conventions.

The calculus is worded in a language being an extension of the language of mbC, where the role of the additional χ operator is to syntactically express the fact that certain formulas (e.g. of the form '$\sim A$', '$\circ A$') may have a logical value independent of the value of A.

History: The calculus $\mathbf{E}^{\mathsf{mbC}}_{res}$ has been presented in [2] together with similar calculi for CLuN, CLuNs and for Classical Propositional Logic. The idea to use χ operator was taken from [1], where the authors presented erotetic calculi for logics CLuN and CLuNs in a non-resolution account.

Remarks: A formula A is mbC-valid iff $\dashv A$ has a Socratic refutation in $\mathbf{E}^{\mathsf{mbC}}_{res}$. A formula A is CLuN-valid iff $\dashv A$ has a Socratic refutation constructed without the use of rules $\mathbf{R}_{\circ}$, $\mathbf{R}_{\neg\circ}$. Similar results are obtained with respect to CLuNs and the classical case.

---

[1] Andrzej Wiśniewski, Guido Vanackere, and Dorota Leszczyńska. "Socratic Proofs and Paraconsistency: A Case Study". In: *Studia Logica* 80.2-3 (2004), pp. 433–468.

[2] Szymon Chlebowski and Dorota Leszczyńska-Jasion. "Dual Erotetic Calculi and the Minimal LFI". In: (2015). doi: 10.1007/s11225-015-9617-0.

Entry 82 by: Dorota Leszczyńska-Jasion

Multi-type Sequent Calculi Mt.SC (2014, 2016)

Identity and Cut rules:

$$\text{LId}\ \ p^{\mathtt{i}} \vdash C^{\mathtt{i}}[p^{\mathtt{i}}]^{suc} \qquad \text{RId}\ \ C^{\mathtt{i}}[p^{\mathtt{i}}]^{pre} \vdash p^{\mathtt{i}} \qquad \text{LCut}\ \frac{Z^{\mathtt{i}} \vdash A^{\mathtt{i}} \qquad (X \vdash Y)[A^{\mathtt{i}}]^{pre}}{(X \vdash Y)[Z/A]^{pre}} \qquad \frac{(X \vdash Y)[A^{\mathtt{i}}]^{suc} \qquad A^{\mathtt{i}} \vdash Z^{\mathtt{i}}}{(X \vdash Y)[Z/A]^{suc}}\ \text{RCut}$$

Introduction rules: for any $f \in \mathcal{F}$ and $g \in \mathcal{G}$,

$$f_L\ \frac{H(A_1,\dots,A_{n_f}) \vdash X}{f(A_1,\dots,A_{n_f}) \vdash X} \qquad \frac{\left(X_i \vdash A_i \quad A_j \vdash X_j \mid 1 \le i,j \le n_f, \varepsilon_f(i) = 1 \text{ and } \varepsilon_f(j) = \partial\right)}{H(X_1,\dots,X_{n_f}) \vdash f(A_1,\dots,A_n)}\ f_R$$

$$g_L\ \frac{\left(A_i \vdash X_i \quad X_j \vdash A_j \mid 1 \le i,j \le n_g, \varepsilon_g(i) = 1 \text{ and } \varepsilon_g(j) = \partial\right)}{g(A_1,\dots,A_{n_g}) \vdash K(X_1,\dots,X_n)} \qquad \frac{X \vdash K(A_1,\dots,A_{n_g})}{X \vdash g(A_1,\dots,A_{n_g})}\ g_R$$

Display postulates: for any $f \in \mathcal{F}$ and $g \in \mathcal{G}$, for some/any $1 \le i \le n_f$ and $1 \le h \le n_g$,

$$(\varepsilon_f(i) = 1)\ \frac{H(X_1,\dots,X_i,\dots,X_{n_f}) \vdash Y}{X_i \vdash H_i(X_1,\dots,Y,\dots,X_{n_f})} \qquad \frac{H(X_1,\dots,X_i,\dots,X_{n_f}) \vdash Y}{H_i(X_1,\dots,Y,\dots,X_{n_f}) \vdash X_i}\ (\varepsilon_f(i) = \partial)$$

$$(\varepsilon_g(h) = 1)\ \frac{Y \vdash K(X_1\dots,X_h,\dots X_{n_g})}{K_h(X_1,\dots,Y,\dots,X_{n_g}) \vdash X_h} \qquad \frac{Y \vdash K(X_1,\dots,X_h,\dots,X_{n_g})}{X_h \vdash K_h(X_1,\dots,Y,\dots,X_{n_g})}\ (\varepsilon_g(h) = \partial)$$

Clarifications: The language $\mathcal{L}_{\mathrm{MT}}(\mathcal{F},\mathcal{G})$ of any MtSC consists of *logical* (or *operational*) and *structural terms* in each (pairwise disjoint) type $\mathsf{T}_1,\dots,\mathsf{T}_n$.

The set of logical terms takes as parameters: 1) denumerable (possibly empty) sets of atomic terms $\mathsf{At}(\mathsf{T}_i)$ for *some* $1 \le i \le m$, elements of which are denoted $p^{\mathtt{i}}$, possibly with indexes; 2) disjoint sets of connectives $\mathcal{F}$ and $\mathcal{G}$. Each $f \in \mathcal{F}$ and $g \in \mathcal{G}$ has arity $n_f \in \mathbb{N}$ (resp. $n_g \in \mathbb{N}$), and is associated with some *functional type* $f : \mathsf{T}_{i_1} \times \cdots \times \mathsf{T}_{i_{n_f}} \longrightarrow \mathsf{T}_f$ (resp. $g : \mathsf{T}_{i_1} \times \cdots \times \mathsf{T}_{i_{n_g}} \longrightarrow \mathsf{T}_g$) and with some *order-type* ε_f over n_f (resp. ε_g over n_g), where an *order-type* over $m \in \mathbb{N}$ is an m-tuple $\varepsilon \in \{1,\partial\}^m$. The functional type of each connective uniquely determines which type T_i is taken as argument in each coordinate and the type of the output of the connective. The order-type of each connective f or g determines which of its coordinates is monotone ($\varepsilon_f(i) = 1$ and $\varepsilon_g(i) = 1$ respectively) or antitone ($\varepsilon_f(i) = \partial$ and $\varepsilon_g(i) = \partial$ respectively). If $\mathsf{T}_{i_1} = \cdots = \mathsf{T}_{i_{n_f}} = \mathsf{T}_f = \mathsf{T}$ (resp. $\mathsf{T}_{i_1} = \cdots = \mathsf{T}_{i_{n_g}} = \mathsf{T}_g = \mathsf{T}$) then f (resp. g) is *homogeneous* of type T ($f \in \mathcal{F}_{\mathsf{T}}$ and $g \in \mathcal{G}_{\mathsf{T}}$); otherwise, f (resp. g) is *heterogeneous* ($f \in \mathcal{F}_{\mathrm{MT}}$ and $g \in \mathcal{G}_{\mathrm{MT}}$). The structural terms are built by means of structural connectives, taking logical terms as atomic structures. The set of structural connectives includes the structural counterpart H (resp. K) of each $f \in \mathcal{F}$ ($g \in \mathcal{G}$), and possibly connectives H_i (resp. K_j) corresponding to the residual of H (resp. K) in the ith (resp. jth) coordinate, for $1 \le i \le n_f$ (resp. $1 \le j \le n_g$). Summing up, for each $1 \le i \le n$, the logical and structural terms of type T_i are generated by simultaneous induction as follows (where $\overline{A}$ and $\overline{X}$ are vectors of formulas and structures, respectively of suitable length):

$$A^{\mathtt{i}} ::= (p^{\mathtt{i}} \mid) f(\overline{A}) \mid g(\overline{A}) \qquad \text{and} \qquad X^{\mathtt{i}} ::= (A^{\mathtt{i}} \mid) H(\overline{X}) \mid K(\overline{X}) (\mid H'_i(\overline{X}) \mid K'_j(\overline{X}).$$

In the Identity rules, the notation $C[p^{\mathtt{i}}]^{pre}$ (resp. $C[p^{\mathtt{i}}]^{suc}$) indicates that $p^{\mathtt{i}}$ occurs in precedent (resp. succedent) position in the structure C. Notice that in the identity rules the structural context C is specific to each logic and might be empty. In the Cut rules, the notation $(X \vdash Y)[A^{\mathtt{i}}]^{pre}$ (resp. $(X \vdash Y)[A^{\mathtt{i}}]^{suc}$) indicates that $A^{\mathtt{i}}$ occurs in precedent (resp. succedent) position in the sequent $X \vdash Y$. For each type T_i, the calculus defines a deduction relation $\vdash^{\mathtt{i}}$ (the superscript is usually dropped unambiguously).

Entry 83 by: Giuseppe Greco

Introduction rules for $f \in \mathcal{F}$ and $g \in \mathcal{G}$ encode the information relative to their arity, functional type and order-type, and cover also zero-ary connectives (e.g. $\top \in \mathcal{F}$ and $\bot \in \mathcal{G}$) and classical conjunction and disjunction ($\wedge \in \mathcal{F}$ and $\vee \in \mathcal{G}$). Notice that the auxiliary formulas in the premises and the principal formula in the conclusion of each rule occur in isolation (this feature is sometimes referred to as the *visibility property*).

If each operator has an adjoint or a residual in each coordinate, then the calculus includes all display postulates, and hence enjoys the so-called *display property*, which makes it a multi-type *display* calculus. Notice that the display property implies the visibility property but not conversely. If each rule is *closed under uniform substitution within each type*, the multi-type (display) calculus is called *proper*.

History: Belnap-style cut elimination for multi-type sequent calculi was proved in [4], where the principal atomic formulas in the Identity rules are required to be displayable. The restricted meta-theorem for proper multi-type display calculi is stated in [8, Sec. A] (cf. Thm. A2). Display calculi are introduced by Belnap [1], and proper display calculi by Wansing [2]. The visibility property is identified by Sambin et al. (cf. [5]). Properly displayable logics are characterized in a purely proof-theoretic way in [6] and in an algebraic way in [7], building on the theory of unified correspondence [3].

Remarks: (Proper) Multi-type calculi are endowed by design with a natural algebraic semantics, thanks to which soundness, completeness and conservativity can be proved in a uniform way for the basic calculi reported on above, and are preserved when extending the calculi with *analytic* structural rules (cf. [7, Definition 4]). Cut elimination and subformula property for MtSC can be inferred from a meta-theorem, following the strategy introduced by Belnap for display calculi. These properties are preserved when adding or removing analytic structural rules. Logics which are not properly displayable have been presented as proper multi-type display calculi, see for instance semi DeMorgan logic {92}, inquisitive logic {89}, lattice logic {91}, dynamic epistemic logic {90}, linear logic [8], and propositional dynamic logic. The logic of resources and capabilities is the first logic originally introduced with a multi-type calculus.

---

[1] Nuel Belnap. "Display Logic". In: *Journal of Philosophical Logic* 11 (1982), pp. 375–417.

[2] Heinrich Wansing. *Displaying modal Logic*. Kluwer, 1998.

[3] Willem Conradie, Silvio Ghilardi, and Alessandra Palmigiano. "Unified Correspondence". In: *Johan van Benthem on Logic and Information Dynamics*. Ed. by Alexandru Baltag and Sonja Smets. Vol. 5. Outstanding Contributions to Logic. Springer International Publishing, 2014, pp. 933–975.

[4] Sabine Frittella, Giuseppe Greco, Alexander Kurz, Alessandra Palmigiano, and Vlasta Sikimić. "Multi-type sequent calculi". In: *Proceedings Trends in Logic XIII, A. Indrzejczak, J. Kaczmarek, M. Zawidski eds* 13 (2014), pp. 81–93.

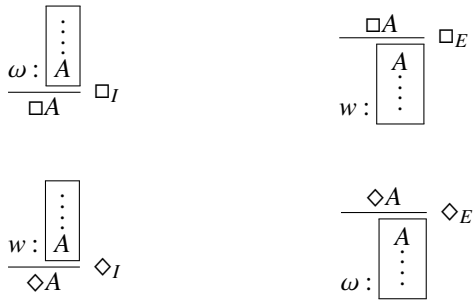
[5] Giovanni Sambin, Giulia Battilotti, and Claudia Faggian. "Basic logic: reflection, symmetry, visibility". In: *Journal of Symbolic Logic* 65.3 (2014), pp. 979–1013. DOI: 10.2307/2586685.

[6] Agata Ciabattoni and Revantha Ramanayake. "Power and limits of structural display rules". In: *ACM Transactions on Computational Logic (TOCL)* 17.3 (2016), p. 17.

[7] Giuseppe Greco, Minghui Ma, Alessandra Palmigiano, Apostolos Tzimoulis, and Zhiguang Zhao. "Unified Correspondence as a Proof-Theoretic Tool". In: *Journal of Logic and Computation* (2016).

[8] Giuseppe Greco and Alessandra Palmigiano. "Linear Logic properly displayed". ArXiv preprint 1611.04181. Nov. 2016.

Modal Natural Deduction (2015)



eigen-box condition:
a modal inference is said to *access*
the box immediately above or below it.
$\Box_I$ and $\Diamond_E$ are *strong* modal rules:
ω must be a fresh name for the box they access
(in analogy to the eigen-variable condition for strong quantifier rules).
Every box must be accessed by *exactly one* strong modal inference.

boxed assumption condition:
assumptions should be discharged within the box where they are created.

Clarifications: This calculus extends any natural deduction calculus (e.g. {5}) by surrounding derivations by boxes and by adding the four modal rules shown above, which introduce and eliminate modal operators, moving formulas in and out of boxes.

History: An embedding of (a higher-order version of) this calculus in the Coq proof assistant was mentioned in [1] and the calculus was studied in more depth in [2], where it was used to formalize Gödel's ontological argument.

Remarks: In [2], this calculus was shown to be sound and complete relative to a natural deduction calculus extended with a necessitation rule and the modal axiom *K*. Consequently, the calculus is sound and complete for rigid modal logic **K**.

---

[1] Christoph Benzmüller and Bruno Woltzenlogel Paleo. "Interacting with Modal Logics in the Coq Proof Assistant". In: *Computer Science - Theory and Applications - 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings*. Ed. by Lev D. Beklemishev and Daniil V. Musatov. Vol. 9139. Lecture Notes in Computer Science. Springer, 2015, pp. 398–411. ISBN: 978-3-319-20296-9. DOI: 10.1007/978-3-319-20297-6_25.

[2] Annika Kanckos and Bruno Woltzenlogel Paleo. "Variants of Gödel's Ontological Proof in a Natural Deduction Calculus". In: *Studia Logica* 105.3 (2017), pp. 553–586. DOI: 10.1007/s11225-016-9700-1.

Entry 84 by: Bruno Woltzenlogel Paleo

Polynomial Ring Calculus with Operators (2015)

As basis for modal logics, formulas of the Classical Propositional Logic CPL are translated into Boolean rings BR via the translation T_{BR}:$Form_{CPL} \longrightarrow Term_{BR}$, recursively defined by:

$$\begin{aligned} T_{BR}(p_i) &= x_i, \\ T_{BR}(\neg\alpha) &= T_{BR}(\alpha) + 1, \\ T_{BR}(\alpha \vee \beta) &= T_{BR}(\alpha) \cdot T_{BR}(\beta) + T_{BR}(\alpha) + T_{BR}(\beta), \\ T_{BR}(\alpha \wedge \beta) &= T_{BR}(\alpha) \cdot T_{BR}(\beta), \\ T_{BR}(\alpha \longrightarrow \beta) &= T_{BR}(\alpha) \cdot T_{BR}(\beta) + T_{BR}(\alpha) + 1, \\ T_{BR}(\alpha \leftrightarrow \beta) &= T_{BR}(\alpha) + T_{BR}(\beta) + 1. \end{aligned}$$

This obtains an immediate proof procedure and decision method for CPL, in the sense that $\vdash_{CPL} \alpha$ iff $BR \vDash T_{BR}(\alpha) \approx 1$, i.e, iff the translated expression reduces to 1 by cancellation rules of polynomials with coefficients in $\mathbb{Z}_2$.
The structure $\mathcal{R} = \langle R, \mathbf{n} \rangle$ is a *boolean ring with operators* (BRO, or *modal ring*) if BR is a BR and $\mathbf{n}$:$A \longrightarrow A$ satisfies:

$$\begin{aligned} \mathbf{n}(1) &= 1, \\ \mathbf{n}(x \cdot y) &= \mathbf{n}(x) \cdot \mathbf{n}(y). \end{aligned}$$

BRO defines a *Polynomial Ring Calculus with Operators* (PRCO) for the normal modal logic K. Subsequent proof systems for normal modal logics are obtained by adding new conditions to this basic PRCO, according to the characteristics of the necessitation operator. For each of the the following axioms that define the systems KD, $KT = T$, $KT4 = S4$ and $KT5 = S5$, the specific conditions are:
1. For D, $\Box\alpha \longrightarrow \Diamond\alpha$ defines PRCO_D, the extension of PRCO by adding:

$$\mathbf{n}(0) \vdash_{\approx} 0.$$

2. For T, $\Box\alpha \longrightarrow \alpha$ defines PRCO_T, the extension of PRCO by adding:

$$\mathbf{n}(P) \vdash_{\approx} P\mathbf{n}(P).$$

3. For S4, $\Box\alpha \longrightarrow \Box\Box\alpha$ defines PRCO_{S4}, the extension of PRCO_T by adding:

$$\begin{aligned} &\mathbf{n}(P) \vdash_{\approx} \mathbf{n}(QR) \text{ (if Fact}(P) = Q\mathbf{n}(R) \text{ and } D(Q) \leq D(R)), \\ &\mathbf{n}(P) \vdash_{\approx} \mathbf{n}(\mathbf{n}(Q)R) \text{ (if Fact}(P) = QR \text{ and } D(Q) < D(R)). \end{aligned}$$

   where $Fact(Q)$ denotes the factorization of a modal polynomial Q.
4. For S5, $\Diamond\alpha \longrightarrow \Box\Diamond\alpha$ defines PRCO_{S5}, the extension of PRCO_T by adding:

$$\mathbf{n}(P\mathbf{n}(Q) + R) \vdash_{\approx} \mathbf{n}(QR + QP) + \mathbf{n}(QR) + \mathbf{n}(R).$$

Clarifications: The Polynomial Ring Calculus with Operators (PRCO's) are algebraic proof methods, here devoted to the modal logics K, KD, T, S4, S5 and to intuitionistic logic. They consist of translating formulas

Entry 85 by: Walter Carnielli

of a logical system into polynomials over an adequate field, in such a way that algebraic reductions on polynomials allow us to determine whether a formula is or is not a theorem of the respective system. The decision procedure works as a proof in algebraic terms. As shown in [1], PRC are particularly applicable to any truth-functional finite-valued propositional logic, and can also be applied to non-truth-functional propositional logics, see [4] and [6].

History: Polynomial Ring Calculi (PRC) were introduced in [1] and extended in [2], [3] and [5]. Conceptual discussions and extensions to non-deterministic semantics appear in [4] and [6].

Remarks: Full completeness, several examples and a comparison with other methods are provided in [5].

---

[1] Walter Carnielli. "Polynomial ring calculus for many-valued logics". In: *Proceedings of the 35th International Symposium on Multiple-Valued Logic*. Ed. by B. Werner. Vol. 7750. IEEE Computer Society, 2005, pp. 20–25.

[2] Walter Carnielli. "Polynomizing: Logic inference in polynomial format and the legacy of Boole". In: *Model-Based Reasoning in Science, Technology, and Medicine*. Ed. by L. Magnani and P. Li. Springer Berlin Heidelberg, 2007, pp. 349–364. DOI: 10.1007/978-3-540-71986-1_20.

[3] Juan C. Agudelo-Agudelo and Walter Carnielli. "Polynomial ring calculus for modal logics: a new semantics and proof method for modalities". In: *Review of Symbolic Logic* 4.1 (2011), pp. 150–170.

[4] Walter Carnielli and Mariana Matulovic. "Non-deterministic semantics in polynomial format". In: *Electronic Notes in Theoretical Computer Science* 305 (2014), pp. 19–34.

[5] Juan C. Agudelo-Agudelo and Walter Carnielli. "Polynomial ring calculus for modalities". In: *Journal of Logic and Computation* (2015). Advanced publication. URL: https://doi.org/10.1093/logcom/exv069.

[6] Walter Carnielli and Mariana Matulovic. "The method of polynomial ring calculus and its potentialities". In: *Theoretical Computer Science* 606 (2015), pp. 42–56.

Conflict Resolution (2016)

Unit-Propagating Resolution:

$$\frac{\ell_1 \quad \ldots \quad \ell_n \quad \overline{\ell'_1} \vee \ldots \vee \overline{\ell'_n} \vee \ell}{\ell\,\sigma}\ \mathbf{u}(\sigma)$$

σ is a unifier of ℓ_k and ℓ'_k, for all $k \in \{1, \ldots, n\}$

Conflict:

$$\frac{\ell \quad \overline{\ell'}}{\bot}\ \mathbf{c}(\sigma)$$

σ is a unifier of ℓ and ℓ'.

Conflict-Driven Clause Learning:

$$\frac{\begin{array}{cc}[\ell_1]^i_1 & [\ell_n]^i_n \\ \vdots\ (\sigma^1_1, \ldots, \sigma^1_{m_1}) & \vdots\ (\sigma^n_1, \ldots, \sigma^n_{m_n}) \\ & \vdots \\ & \bot\end{array}}{(\overline{\ell_1}\sigma^1_1 \vee \ldots \vee \overline{\ell_1}\sigma^1_{m_1}) \vee \ldots \vee (\overline{\ell_n}\sigma^n_1 \vee \ldots \vee \overline{\ell_n}\sigma^n_{m_n})}\ \mathbf{cl}^i$$

σ^k_j is the composition of all substitutions used on the j-th path from ℓ_k to $\bot$.

Clarifications: Proofs are directed acyclic graphs and not necessarily tree-like. Therefore, there may be several paths connecting ℓ_k to $\bot$. The conflict-driven clause learning rule takes all paths into account. When restricted to propositional logic, **CR**-derivations ending in $\bot$ are isomorphic to *conflict graphs* of SAT-solvers.

History: Unit-propagating resolution is a restriction of resolution {13} also known as unit-resulting resolution. The conflict-driven clause learning rule is, at the proof-theoretical level, a first-order lifting of a procedure implemented by SAT-solvers. This rule generalizes natural deduction's implication introduction rule to the case with unification and multiple assumptions, as Robinson's resolution rule generalizes implication elimination (modus ponens) [1]. `Scavenger` [4, 2] was the first theorem prover based on Conflict Resolution.

Remarks: Soundness was proven by simulation by a clausal variant of natural deduction {5} (i.e. every **CR**-proof can be translated into a natural deduction proof) and refutational completeness was proven by simulating resolution {13} (i.e. every resolution refutation can be translated into a **CR**-refutation) [3].

[1] Bruno Woltzenlogel Paleo. "First-Order Conflict-Driven Clause Learning from a Proof-Theoretical Perspective". In: *Dagstuhl seminar on the Universality of Proofs*. 2016.

[2] Daniyar Itegulov, John Slaney, and Bruno Woltzenlogel Paleo. "Scavenger 0.1: A Theorem Prover Based on Conflict Resolution". In: *Automated Deduction - CADE-26 - 26th International Conference on Automated Deduction, Gothenburg, Sweden, August 6-11, 2017, Proceedings*. 2017.

[3] John Slaney and Bruno Woltzenlogel Paleo. "Conflict Resolution: a First-Order Resolution Calculus with Decision Literals and Conflict-Driven Clause Learning". In: *Journal of Automated Reasoning* (2017), pp. 1–24. ISSN: 1573-0670. DOI: 10.1007/s10817-017-9408-6.

[4] John Slaney and Bruno Woltzenlogel Paleo. "Proof Search in the Conflict Resolution Calculus". In: *Second Conf. on Artificial Intelligence and Theorem Proving, March 26–30, Obergurgl, Austria*. 2017.

Entry 86 by: Bruno Woltzenlogel Paleo

Sequent Systems for Negative Modalities (2016)

$$[\sim\Rightarrow]\ \frac{\Gamma\Rightarrow\varphi,\Delta}{\sim\Delta,\sim\varphi\Rightarrow\sim\Gamma} \qquad [\Rightarrow\sim]\ \frac{\Gamma,\varphi\Rightarrow\Delta}{\sim\Delta\Rightarrow\sim\varphi,\sim\Gamma}$$

$$[\Rightarrow\ominus]\ \frac{\Gamma,\varphi,\sim\varphi\Rightarrow\Delta}{\Gamma\Rightarrow\ominus\varphi,\Delta} \qquad [\ominus\Rightarrow]\ \frac{\Gamma\Rightarrow\varphi,\Delta\quad\Gamma\Rightarrow\sim\varphi,\Delta}{\Gamma,\ominus\varphi\Rightarrow\Delta} \qquad [\Rightarrow\ominus]\ \frac{\Gamma,\varphi\Rightarrow\Delta\quad\Gamma,\sim\varphi\Rightarrow\Delta}{\Gamma\Rightarrow\ominus\varphi,\Delta} \qquad [\ominus\Rightarrow]\ \frac{\Gamma\Rightarrow\varphi,\sim\varphi,\Delta}{\Gamma,\ominus\varphi\Rightarrow\Delta}$$

$$[\mathbf{D}]\ \frac{\Gamma\Rightarrow\Delta}{\sim\Delta\Rightarrow\sim\Gamma} \qquad [\mathbf{Fun}]\ \frac{\Gamma\Rightarrow\Delta}{\sim\Delta\Rightarrow\sim\Gamma} \qquad [\mathbf{T}_1]\ \frac{\Gamma,\varphi\Rightarrow\Delta}{\Gamma\Rightarrow\sim\varphi,\Delta} \qquad [\mathbf{T}_2]\ \frac{\Gamma\Rightarrow\varphi,\Delta}{\Gamma,\sim\varphi\Rightarrow\Delta}$$

$$[\mathbf{B}_1]\ \frac{\Gamma,\sim\Gamma',\varphi\Rightarrow\Delta,\sim\Delta'}{\sim\Delta,\Delta'\Rightarrow\sim\varphi,\sim\Gamma,\Gamma'} \qquad [\mathbf{B}_2]\ \frac{\Gamma,\sim\Gamma'\Rightarrow\varphi,\Delta,\sim\Delta'}{\sim\Delta,\Delta',\sim\varphi\Rightarrow\sim\Gamma,\Gamma'}$$

$$[\mathbf{4}_1]\ \frac{\sim\Gamma,\Gamma',\varphi\Rightarrow\sim\Delta,\Delta'}{\sim\Gamma,\sim\Delta'\Rightarrow\sim\varphi,\sim\Delta,\sim\Gamma'} \qquad [\mathbf{4}_2]\ \frac{\sim\Gamma,\Gamma'\Rightarrow\varphi,\sim\Delta,\Delta'}{\sim\Gamma,\sim\Delta',\sim\varphi\Rightarrow\sim\Delta,\sim\Gamma'}$$

$$[\mathbf{D}_B]\ \frac{\sim\Gamma',\Gamma\Rightarrow\Delta,\sim\Delta'}{\Delta',\sim\Delta\Rightarrow\sim\Gamma,\Gamma'} \qquad [\mathbf{D}_4]\ \frac{\sim\Gamma',\Gamma\Rightarrow\Delta,\sim\Delta'}{\sim\Gamma',\sim\Delta\Rightarrow\sim\Gamma,\sim\Delta'}$$

Clarifications: P is obtained from propositional positive **LK** {6} by the addition of the rules for $\ominus$ and $\ominus$. PK = P + $[\sim\Rightarrow]$ + $[\sim\Rightarrow]$, PKD = PK + $[\mathbf{D}]$, PKT = PK + $[\mathbf{T}_1]$ + $[\mathbf{T}_2]$, PKF = P + $[\mathbf{Fun}]$ + $[\sim=\sim]$, PKB = P + $[\mathbf{B}_1]$ + $[\mathbf{B}_2]$, PK4 = P + $[\mathbf{4}_1]$ + $[\mathbf{4}_2]$, PKDB = PKB + $[\mathbf{D_B}]$, and PKD4 = PK4 + $[\mathbf{D_4}]$.

History: All systems enjoy the replacement property, which is missing in various logics that employ a non-classical negation, such as Kleene's three-valued logic, and da Costa's C_1. PK was introduced in [2]. All other systems were introduced in [3, 4].

Remarks: All systems enjoy a generalized sub-formula property, in which $\sim\varphi$ is considered a sub-formula of $\ominus\varphi$ (and $\sim\varphi$ is a sub-formula of $\ominus\varphi$). All systems that do not contain **B** also enjoy cut-admissibility. These facts were proven semantically, using the general mechanism of [1].

---

[1] Ori Lahav and Arnon Avron. "A Unified Semantic Framework for Fully Structural Propositional Sequent Systems". In: *ACM Trans. Comput. Logic* 14.4 (2013), 27:1–27:33. DOI: 10.1145/2528930.

[2] Adriano Dodó and João Marcos. "Negative modalities, consistency and determinedness". In: *Electronic Notes in Theoretical Computer Science* 300 (Jan. 2014), pp. 21–45.

[3] Ori Lahav, João Marcos, and Yoni Zohar. "It ain't necessarily so: Basic sequent systems for negative modalities". In: *Advances in Modal Logic*. Vol. 11. College Publications, 2016, pp. 449–468.

[4] Ori Lahav, João Marcos, and Yoni Zohar. "Sequent Systems for Negative Modalities". In: *Logica Universalis* (July 2017). DOI: 10.1007/s11787-017-0175-2.

Entry 87 by: Yoni Zohar

Update Logic UL (2016)

$$\frac{\dfrac{X \,,_{\mathrm{i}}\, Y \vdash Z}{X \vdash *Y \,,_{\mathrm{j}}\, Z}}{Y \vdash Z \,,_{\mathrm{k}}\, *X} \qquad \frac{\dfrac{Z \vdash X \,,_{\mathrm{i}}\, Y}{*Y \,,_{\mathrm{j}}\, Z \vdash X}}{Z \,,_{\mathrm{k}}\, *X \vdash Y} \qquad \frac{X \vdash \bullet Y}{\bullet X \vdash Y} \qquad \frac{U \vdash \varphi \quad \varphi \vdash V}{U \vdash V}\ Cut$$

$$\frac{U \vdash V}{U, X \vdash V}\ \mathsf{K} \qquad \frac{X, X \vdash U}{X \vdash U}\ \mathsf{WI} \qquad \frac{Y, X \vdash U}{X, Y \vdash U}\ \mathsf{CI} \qquad \frac{(X, Y), Z \vdash U}{X, (Y, Z) \vdash U}\ \mathsf{B}^c$$

$$\frac{}{p \vdash p}\ Id \qquad \frac{}{\bot \vdash}\ \bot_A \qquad \frac{}{\vdash \top}\ \top_K$$

$$\frac{U \vdash \varphi}{U \vdash \varphi \vee \psi}\ \vee^1_K \qquad \frac{U \vdash \psi}{U \vdash \varphi \vee \psi}\ \vee^2_K \qquad \frac{\varphi \vdash U \quad \psi \vdash U}{\varphi \vee \psi \vdash U}\ \vee_A \qquad \frac{U \vdash *\varphi}{U \vdash \neg\varphi}\ \neg_K$$

$$\frac{U \vdash \varphi \quad U \vdash \psi}{U \vdash \varphi \wedge \psi}\ \wedge_K \qquad \frac{\varphi \vdash U}{\varphi \wedge \psi \vdash U}\ \wedge^1_A \qquad \frac{\psi \vdash U}{\varphi \wedge \psi \vdash U}\ \wedge^2_A \qquad \frac{*\varphi \vdash U}{\neg\varphi \vdash U}\ \neg_A$$

$$\frac{X \vdash \varphi \quad Y \vdash \psi}{X \,,_{\mathrm{i}}\, Y \vdash \varphi \otimes_{\mathrm{i}} \psi}\ \otimes^{\mathrm{i}}_K \qquad \frac{\varphi \,,_{\mathrm{i}}\, \psi \vdash U}{\varphi \otimes_{\mathrm{i}} \psi \vdash U}\ \otimes^{\mathrm{i}}_A \qquad \frac{U \vdash \varphi \,,_{\mathrm{i}}\, \psi}{U \vdash \varphi \oplus_{\mathrm{i}} \psi}\ \oplus^{\mathrm{i}}_K \qquad \frac{\varphi \vdash X \quad \psi \vdash Y}{\varphi \oplus_{\mathrm{i}} \psi \vdash X \,,_{\mathrm{i}}\, Y}\ \oplus^{\mathrm{i}}_A$$

$$\frac{X \,,_{\mathrm{i}}\, \varphi \vdash \psi}{X \vdash \varphi \supset_{\mathrm{j}} \psi}\ \supset^{\mathrm{j}}_K \qquad \frac{X \vdash \varphi \quad \psi \vdash Y}{\varphi \supset_{\mathrm{j}} \psi \vdash *X \,,_{\mathrm{j}}\, Y}\ \supset^{\mathrm{j}}_A \qquad \frac{Y \vdash \psi \quad \varphi \vdash X}{*X \,,_{\mathrm{j}}\, Y \vdash \psi \prec_{\mathrm{j}} \varphi}\ \prec^{\mathrm{j}}_K \qquad \frac{\psi \vdash \varphi \,,_{\mathrm{k}}\, X}{\psi \prec_{\mathrm{j}} \varphi \vdash X}\ \prec^{\mathrm{j}}_A$$

$$\frac{\varphi \,,_{\mathrm{i}}\, X \vdash \psi}{X \vdash \psi \subset_{\mathrm{k}} \varphi}\ \subset^{\mathrm{k}}_K \qquad \frac{\psi \vdash Y \quad X \vdash \varphi}{\psi \subset_{\mathrm{k}} \varphi \vdash Y \,,_{\mathrm{k}}\, *X}\ \subset^{\mathrm{k}}_A \qquad \frac{\varphi \vdash X \quad Y \vdash \psi}{Y \,,_{\mathrm{k}}\, *X \vdash \varphi \succ_{\mathrm{k}} \psi}\ \succ^{\mathrm{k}}_K \qquad \frac{\psi \vdash \varphi \,,_{\mathrm{i}}\, X}{\varphi \succ_{\mathrm{k}} \psi \vdash X}\ \succ^{\mathrm{k}}_A$$

$(i, j, k) \in \{(1,2,3), (2,3,1), (3,1,2)\}$

$$\frac{\bullet X \vdash \varphi}{X \vdash \Box\varphi}\ \Box_K \qquad \frac{\varphi \vdash X}{\Box\varphi \vdash \bullet X}\ \Box_A \qquad \frac{U \vdash *\bullet *\varphi}{U \vdash \Box^-\varphi}\ \Box^-_K \qquad \frac{\varphi \vdash X}{\Box^-\varphi \vdash *\bullet *X}\ \Box^-_A$$

$$\frac{X \vdash \varphi}{\bullet X \vdash \Diamond^-\varphi}\ \Diamond^-_K \qquad \frac{\varphi \vdash \bullet X}{\Diamond^-\varphi \vdash X}\ \Diamond^-_A \qquad \frac{X \vdash \varphi}{*\bullet *X \vdash \Diamond\varphi}\ \Diamond_K \qquad \frac{*\bullet *\varphi \vdash U}{\Diamond\varphi \vdash U}\ \Diamond_A$$

Clarifications: Update logic [1] generalizes the non–associative Lambek calculus {12} and includes modal, propositional and dual substructural connectives. Correspondences between frame properties and structural display calculus rules are used to define novel display sequent calculi for a broad range of substructural logics (bi–intuitionistic logic is a case study of [1]).

Remarks: Update logic is a (proper) display calculus which enjoys cut elimination.

[1] Guillaume Aucher. "Displaying updates in logic". In: *J. Log. Comput.* 26.6 (2016), pp. 1865–1912. DOI: 10.1093/logcom/exw001. URL: https://doi.org/10.1093/logcom/exw001.

Entry 88 by: Guillaume Aucher

Proper Multi-type Display Calculus for Inquisitive Logic MtD.InqL (2016)

Identity and Cut rules:

$$\frac{}{p \vdash p}\ \text{Id} \qquad \frac{X \vdash A \qquad A \vdash Y}{X \vdash Y}\ \text{Cut} \qquad \frac{\Gamma \vdash \alpha \qquad \alpha \vdash \Delta}{\Gamma \vdash \Delta}\ \text{Cut}$$

Structural rules governing the interaction between the types Flat and General:

$$\frac{\mathrm{F}^*\Gamma \vdash \Delta}{\Gamma \vdash \mathrm{F}\Delta}\ \text{f adj} \qquad \frac{\mathrm{F}X \vdash \Gamma}{X \vdash \Downarrow\Gamma}\ \text{d adj} \qquad \frac{X \vdash \Downarrow \mathrm{F}Y}{X \vdash Y}\ \text{d-f elim}$$

$$\frac{\Gamma \vdash \Delta}{\mathrm{F}^*\Gamma \vdash \Downarrow\Delta}\ \text{bal} \qquad \frac{X \vdash Y}{\mathrm{F}X \vdash \mathrm{F}Y}\ \text{f mon} \qquad \frac{X \vdash \Downarrow(\Gamma \sqsupset \Delta)}{X \vdash \mathrm{F}^*\Gamma > \Downarrow\Delta}\ \text{d dis} \qquad \frac{\mathrm{F}X, \mathrm{F}Y \vdash Z}{\mathrm{F}(X;Y) \vdash Z}\ \text{f dis}$$

$$\frac{X \vdash \mathrm{F}^*\Gamma > (Y;Z) \qquad X \vdash \mathrm{F}^*\Gamma > (Y;Z)}{X \vdash (\mathrm{F}^*\Gamma > Y);(\mathrm{F}^*\Gamma > Z)}\ \text{KP}$$

Clarifications: The language $\mathcal{L}_{\mathrm{MT}}(\mathcal{F},\mathcal{G})$ of MtD.InqL consists of *logical* and *structural terms* in the types $\mathsf{T}_1 := \mathsf{Flat}$ and $\mathsf{T}_2 := \mathsf{General}$. Following the notation of {83}, the set of logical terms takes as parameters: 1) a denumerable set of atomic terms $\mathsf{At}(\mathsf{Flat})$, elements of which are denoted p, possibly with indexes; 2) disjoint sets of connectives $\mathcal{F} := \mathcal{F}_{\mathsf{Flat}} \uplus \mathcal{F}_{\mathsf{General}} \uplus \mathcal{F}_{\mathrm{MT}}$ and $\mathcal{G} := \mathcal{G}_{\mathsf{Flat}} \uplus \mathcal{G}_{\mathsf{General}} \uplus \mathcal{G}_{\mathrm{MT}}$ defined as follows: $\mathcal{F}_{\mathsf{Flat}} := \{\sqcap\}$, $\mathcal{F}_{\mathsf{General}} := \{\wedge\}$, $\mathcal{F}_{\mathrm{MT}} := \emptyset$, where $n_{\sqcap} = n_{\wedge} = 2$, and $\varepsilon_{\sqcap}(i) = \varepsilon_{\wedge}(i) = 1$ for every $i \in \{1,2\}$, and $\mathcal{G}_{\mathsf{Flat}} := \{0, \rightarrow\}$, $\mathcal{G}_{\mathsf{General}} := \{\vee, \longrightarrow\}$, $\mathcal{G}_{\mathrm{MT}} := \{\downarrow\}$ where $n_0 = 0$, $n_{\rightarrow} = n_{\vee} = n_{\longrightarrow} = 2$ and $n_{\downarrow} = 1$, and $\varepsilon_{\rightarrow}(1) = \varepsilon_{\longrightarrow}(1) = \partial$, and $\varepsilon_{\rightarrow}(2) = \varepsilon_{\longrightarrow}(2) = \varepsilon_{\sqcup}(i) = \varepsilon_{\vee}(i) = 1$ for every $i \in \{1,2\}$, and $\varepsilon_{\downarrow}(1) = 1$. The functional type of the heterogeneous connective $\downarrow$ is $\mathsf{Flat} \to \mathsf{General}$.

The structural terms are built by means of structural connectives, taking logical terms as atomic structures. The set of structural connectives includes $\sqsupset$, Φ, $\Downarrow$ which are the structural counterparts of $\rightarrow, 0, \downarrow$, respectively. It also includes ; as the structural counterpart of both $\wedge$ (when occurring in antecedent position) and $\vee$ (when occurring in succedent position), and $>$ as structural counterpart of $\longrightarrow$ (when occurring in succedent position) and left residual of ; (when occurring in antecedent position). Finally, it includes F and F^*, where F is the left adjoint of $\Downarrow$ and F^* is the left adjoint of F. Hence, the functional type of F is $\mathsf{General} \to \mathsf{Flat}$ and of F^* is $\mathsf{Flat} \to \mathsf{General}$, and $\mathrm{F}^* \in \mathcal{F}$, while $\mathrm{F} \in \mathcal{F} \cap \mathcal{G}$.

Summing up, the well formed terms of MtD.InqL are generated by simultaneous induction as follows:

Flat	General
$\alpha ::= p \mid 0 \mid \alpha \sqcap \alpha \mid \alpha \rightarrow \alpha$	$A ::= \downarrow\alpha \mid A \wedge A \mid A \vee A \mid A \longrightarrow A$
$\Gamma ::= \alpha \mid \Phi \mid \Gamma, \Gamma \mid \Gamma \sqsupset \Gamma \mid \mathrm{F}X$	$X ::= A \mid \Downarrow\Gamma \mid \mathrm{F}^*\Gamma \mid X;X \mid X > X$

The introduction rules instantiate the general template described in {83}, and hence are omitted. Also, the pure-type structural rules are the standard ones capturing classical logic (for Flat) and intuitionistic logic (for General) and are also omitted.

History: Inquisitive logic is the logic of inquisitive semantics, developed by Ciardelli, Groenendijk and Roelofsen [2, 3] for capturing both assertions and questions in natural language. In [4], systematic connections are developed between inquisitive semantics and the so called team semantics for dependence logic [1]. Building on the algebraic analysis of the team semantics of [1], in [5] the team semantics for inquisitive

Entry 89 by: Giuseppe Greco

logic is recast in a multi-type algebraic framework which provides the guidelines for the design of a multi-type sequent calculus for InqL, of which the proper multi-type display calculus above is a straightforward refinement.

Remarks: Every known axiomatization of inquisitive logic is neither analytic nor closed under uniform substitution. Hence, inquisitive logic cannot be captured by a single-type proper display calculus on the basis of any known axiomatization. The calculus above is sound and complete w.r.t. the team semantics for inquisitive logic, reformulated algebraically as discussed in [5]; it is conservative, and enjoys the cut elimination and subformula property as immediate consequences of the general theory of multi-type calculi.

---

[1] Samsom Abramsky and Jouko Väänänen. "From IF to BI". In: *Synthese* 167.2 (2009), pp. 207–230.

[2] Jeroen Groenendijk and Floris Roelofsen. "Inquisitive Semantics and Pragmatics". In: *Meaning, Content, and Argument: Proceedings of the ILCLI International Workshop on Semantics, Pragmatics, and Rhetoric*. Ed. by Jesus M. Larrazabal and Larraitz Zubeldia. University of the Basque Country Publication Service, May 2009, pp. 41–72.

[3] Ivano Ciardelli and Floris Roelofsen. "Inquisitive Logic". In: *Journal of Philosophical Logic* 40.1 (2011), pp. 55–94.

[4] Fan Yang. "On Extensions and Variants of Dependence Logic". PhD thesis. University of Helsinki, 2014.

[5] Sabine Frittella, Giuseppe Greco, Alessandra Palmigiano, and Fan Yang. "A Multi-type Calculus for Inquisitive Logic". In: *WoLLIC 2016, Proceedings LNCS 9803*. Ed. by Jouko Väänänen, Åsa Hirvonen, and Ruy de Queiroz. Springer, 2016, pp. 215–233.

Multi-type Sequent Calculus for Dynamic Epistemic Logic MtD.DEL (2016)

Atom rule: where $F_1,\dots,F_n,G_1,\dots,G_m \in \mathsf{FNC}$, $\circ \in \{ \triangle_0 , \blacktriangle_0 \}$, $\triangleright \in \{ \triangleright_0 , \blacktriangleright_0 \}$ and $n,m \in \mathbb{N}$,

$$F_1 \circ (F_2 \circ \cdots (F_n \circ p)\cdots) \vdash G_1 \triangleright (G_2 \triangleright \cdots (G_m \triangleright p)\cdots)$$

Balance rule:

$$\frac{X \vdash Y}{F \triangle_0 X \vdash F \triangleright_0 Y}$$

Necessitation, conjugation, Fisher Servi and monotonicity rules: for $0 \le i \le 2$,

$$\frac{W \vdash \mathrm{I}}{W \vdash x \blacktriangleright_i \mathrm{I}}\,(nec_i \to) \qquad (conj_i\,\triangle)\ \frac{x \triangle_i ((x \blacktriangle_i Y) ; Z) \vdash W}{Y ; (x \triangle_i Z) \vdash W} \qquad \frac{W \vdash (x \blacktriangle_i Y) > (x \blacktriangleright_i Z)}{W \vdash x \blacktriangleright_i (Y > Z)}\,(FS_i \to)$$

$$(mon_i\,\triangle)\ \frac{(x \triangle_i Y) ; (x \triangle_i Z) \vdash W}{x \triangle_i (Y ; Z) \vdash W} \qquad (mon_i\,\blacktriangle)\ \frac{(x \blacktriangle_i Y) ; (x \blacktriangle_i Z) \vdash W}{x \blacktriangle_i (Y ; Z) \vdash W} \qquad \frac{W \vdash (x \blacktriangleright_i Y) ; (x \blacktriangleright_i Z)}{W \vdash x \blacktriangleright_i (Y ; Z)}\,(mon_i \to)$$

Interaction rules between dynamic and epistemic modalities:

$$\frac{X \vdash (\mathsf{a} \triangle F) \blacktriangleright (\mathsf{a} \blacktriangleright Y)}{X \vdash \mathsf{a} \blacktriangleright (F \blacktriangleright Y)}\,\text{swap-out}_R \qquad \frac{X \vdash \mathsf{a} \blacktriangleright (F \blacktriangleright Y)}{X \vdash (\mathsf{a} \triangle F) \blacktriangleright (\mathsf{a} \blacktriangleright ((F \triangle \mathrm{I}) > Y))}\,\text{swap-in}_R$$

Clarifications: The language $\mathcal{L}_{\mathrm{MT}}(\mathcal{F},\mathcal{G})$ of MtD.DEL consists of *logical* and *structural terms* in the types $\mathsf{T}_1 := \mathsf{Fm}$, $\mathsf{T}_2 := \mathsf{Fnc}$, $\mathsf{T}_3 := \mathsf{Act}$, $\mathsf{T}_4 := \mathsf{Ag}$. Following the notation of {83}, the set of logical terms takes as parameters: 1) a denumerable set of atomic terms $\mathsf{At}(\mathsf{Fm})$, elements of which are denoted p, possibly with indexes; a denumerable set of atomic terms $\mathsf{At}(\mathsf{Fnc})$, elements of which are denoted α, possibly with indexes; a finite set of atomic terms $\mathsf{At}(\mathsf{Ag})$, elements of which are denoted a, possibly with indices; 2) sets of connectives $\mathcal{F} := \mathcal{F}_{\mathsf{Fm}} \uplus \mathcal{F}_{\mathsf{Fnc}} \uplus \mathcal{F}_{\mathsf{Act}} \uplus \mathcal{F}_{\mathsf{Ag}} \uplus \mathcal{F}_{\mathrm{MT}}$ and $\mathcal{G} := \mathcal{G}_{\mathsf{Fm}} \uplus \mathcal{G}_{\mathsf{Fnc}} \uplus \mathcal{G}_{\mathsf{Act}} \uplus \mathcal{G}_{\mathsf{Ag}} \uplus \mathcal{G}_{\mathrm{MT}}$ defined as follows: $\mathcal{F}_{\mathsf{Fm}} := \{\top, \wedge, \succ\!\!-\}$, $\mathcal{F}_{\mathsf{Fnc}} := \emptyset$, $\mathcal{F}_{\mathsf{Act}} := \emptyset$, $\mathcal{F}_{\mathsf{Ag}} := \emptyset$, $\mathcal{F}_{\mathrm{MT}} := \{\triangle_k | \ 0 \le k \le 3\}$, where $n_\top = 0$, $n_\wedge = n_{\succ\!\!-} = n_{\triangle_k} = 2$ for every $0 \le k \le 3$, and $\varepsilon_{\succ\!\!-}(1) = \partial$ and $\varepsilon_\wedge(i) = \varepsilon_{\succ\!\!-}(2) = \varepsilon_{\triangle_k}(i) = 1$ for every $i \in \{1,2\}$, and every $0 \le k \le 3$, and $\mathcal{G}_{\mathsf{Fm}} := \{\bot, \vee, \to\}$, $\mathcal{G}_{\mathsf{Fnc}} := \emptyset$, $\mathcal{G}_{\mathsf{Act}} := \emptyset$, $\mathcal{G}_{\mathsf{Ag}} := \emptyset$, $\mathcal{G}_{\mathrm{MT}} := \{ \multimap_k | \ 0 \le k \le 2\}$, where $n_\bot = 0$, $n_\vee = n_\to = n_{\multimap_k} = 2$ for every $0 \le k \le 3$, and $\varepsilon_\to(1) = \varepsilon_{\multimap_k}(1) = \partial$ and $\varepsilon_\vee(i) = \varepsilon_\to(2) = \varepsilon_{\multimap_k}(2) = 1$ for every $i \in \{1,2\}$, and every $0 \le k \le 2$. The functional types of the heterogeneous connectives are given as follows: $\triangle_0, \multimap_0 : \mathsf{Fnc} \times \mathsf{Fm} \longrightarrow \mathsf{Fm}$, $\triangle_1, \multimap_1 : \mathsf{Act} \times \mathsf{Fm} \longrightarrow \mathsf{Fm}$, $\triangle_2, \multimap_2 : \mathsf{Ag} \times \mathsf{Fm} \longrightarrow \mathsf{Fm}$, $\triangle_3 : \mathsf{Ag} \times \mathsf{Fnc} \longrightarrow \mathsf{Act}$.

The structural terms are built by means of structural connectives, taking logical terms as atomic structures. The set of structural connectives includes $\triangle_k$ and $\triangleright_j$ for each $0 \le k \le 3$ and $0 \le j \le 2$ which are the structural counterparts of $\triangle_k$ and $\multimap_j$, respectively, for each $0 \le k \le 3$ and $0 \le j \le 2$. It also includes ; as the structural counterpart of both $\wedge$ (when occurring in antecedent position) and $\vee$ (when occurring in succedent position), I as the structural counterpart of both $\top$ (when occurring in antecedent position) and $\bot$ (when occurring in succedent position), and $>$ as the structural counterpart of both $\succ\!\!-$ (when occurring in antecedent position) and $\to$ (when occurring in succedent position). Finally, it includes $\blacktriangleleft_k$ and $\blacktriangleright_k$ for each $0 \le k \le 3$, and $\blacktriangle_j$ and $\triangleleft_j$ for each $0 \le j \le 2$, where $\blacktriangleleft_k$ and $\blacktriangleright_k$ are the residuals of $\triangle_k$ in its first and second coordinate respectively, and $\triangleleft_j$ and $\blacktriangle_j$ are the residuals of $\triangleright_j$ in its first and second coordinate respectively. Hence, the functional types of these structural connectives are given as follows: $\blacktriangle_0, \blacktriangleright_0 : \mathsf{Fnc} \times \mathsf{Fm} \longrightarrow \mathsf{Fm}$, $\blacktriangle_1, \blacktriangleright_1 : \mathsf{Act} \times \mathsf{Fm} \longrightarrow \mathsf{Fm}$, $\blacktriangle_2, \blacktriangleright_2 : \mathsf{Ag} \times \mathsf{Fm} \longrightarrow \mathsf{Fm}$, $\blacktriangleright_3 : \mathsf{Ag} \times \mathsf{Act} \longrightarrow \mathsf{Fnc}$, $\triangleleft_0, \blacktriangleleft_0 : \mathsf{Fm} \times \mathsf{Fm} \longrightarrow$

Entry 90 by: Giuseppe Greco

Fnc, $\lhd_1, \blacktriangleleft_1 : \mathsf{Fm} \times \mathsf{Fm} \longrightarrow \mathsf{Act}$, $\lhd_2, \blacktriangleleft_2 : \mathsf{Fm} \times \mathsf{Fm} \longrightarrow \mathsf{Ag}$, $\blacktriangleleft_3 : \mathsf{Act} \times \mathsf{Fnc} \longrightarrow \mathsf{Ag}$. Summing up, the well formed terms of MtD.SDM are generated by simultaneous induction as follows:

$$
\begin{array}{l}
\mathsf{Fm} \\
A ::= p \mid \bot \mid \top \mid A \wedge A \mid A \vee A \mid A \to A \mid A \mathrel{>\!\!-} A \mid \alpha \vartriangle_0 A \mid \alpha \mathrel{-\!\!\triangleright}_0 A \mid \gamma \vartriangle_1 A \mid \gamma \mathrel{-\!\!\triangleright}_1 A \mid \mathsf{a} \vartriangle_2 A \mid \mathsf{a} \mathrel{-\!\!\triangleright}_2 A \\
X ::= A \mid \mathrm{I} \mid X \,; X \mid X > X \mid F \triangle_0 X \mid F \triangleright_0 X \mid \Gamma \triangle_1 X \mid \Gamma \triangleright_1 X \mid \mathsf{a} \triangle_2 X \mid \mathsf{a} \triangleright_2 X \mid \\
\qquad F \blacktriangle_0 X \mid F \blacktriangleright_0 X \mid \Gamma \blacktriangle_1 X \mid \Gamma \blacktriangleright_1 X \mid \mathsf{a} \blacktriangle_2 X \mid \mathsf{a} \blacktriangleright_2 X
\end{array}
$$

Fnc	Act	Ag
$\alpha ::= \alpha$	$\gamma ::= \mathsf{a} \vartriangle_3 \alpha$	$\mathsf{a} ::= \mathsf{a}$
$F ::= \alpha \mid X \lhd_0 X \mid X \blacktriangleleft_0 X \mid \mathsf{a} \blacktriangleright_3 \Gamma$	$\Gamma ::= \mathsf{a} \triangle_3 F \mid X \lhd_1 X \mid X \blacktriangleleft_1 X$	$\mathsf{a} ::= \mathsf{a} \mid X \lhd_2 X \mid X \blacktriangleleft_2 X \mid \Gamma \blacktriangleleft_3 F$

The Identity, Cut, Display and introduction rules instantiate the general template described in {83}, and hence are omitted. Also, the pure-type structural rules for Fm are the standard ones capturing the usual (classical, intuitionistic, substructural...) propositional base, and are also omitted. The calculus MtD.DEL is a proper fragment of the calculus introduced in [4] and captures the diamond-only presentation of dynamic epistemic logic. The rules swap-out$_R$ and swap-in$_R$ are a notational variant of the corresponding rules introduced in [4].

History: MtD.DEL was introduced in [4] and captures Baltag-Moss-Solecki's dynamic epistemic logic [1] in its classical version, and the logic IEAK [3] in its intuitionistic version. An overview of the literature about the proof theoretic approaches for dynamic epistemic logics can be found in [5].

Remarks: The axiomatization of Baltag-Moss-Solecki's dynamic epistemic logic is not closed under uniform substitution and is given in terms of meta-linguistic labels. The multi-type language of MtD.DEL makes it possible to internalize the labels into the object language of the calculus so that when recast in this language, the axioms are analytic inductive according to the definition of [6]. The 0-ary rule Atom captures the axioms which are not closed under uniform substitution. The calculus above is sound and complete w.r.t. final coalgebra semantics introduced in [2]; it is conservative, and enjoys the cut elimination and subformula property as immediate consequences of the general theory of multi-type calculi.

---

[1] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. *The Logic of Public Announcements, Common Knowledge and Private Suspicious*. Tech. rep. SEN-R9922. CWI, Amsterdam, 1999.

[2] Giuseppe Greco, Alexander Kurz, and Alessandra Palmigiano. "Dynamic Epistemic Logic Displayed". In: *Proceedings of the 4th International Workshop on Logic, Rationality and Interaction (LORI-4)*. Ed. by Huaxin Huang, Davide Grossi, and Olivier Roy. Vol. 8196. LNCS. 2013.

[3] Alexander Kurz and Alessandra Palmigiano. "Epistemic Updates on Algebras". In: *Logical Methods in Computer Science* (2013).

[4] Sabine Frittella, Giuseppe Greco, Alexander Kurz, Alessandra Palmigiano, and Vlasta Sikimić. "A Multi-type Display Calculus for Dynamic Epistemic Logic". In: *Journal of Logic and Computation* 26.6 (2016), pp. 2017–2065.

[5] Sabine Frittella, Giuseppe Greco, Alexander Kurz, Alessandra Palmigiano, and Vlasta Sikimić. "A Proof-Theoretic Semantic Analysis of Dynamic Epistemic Logic". In: *Journal of Logic and Computation* 26.6 (2016), pp. 1961–2015.

[6] Giuseppe Greco, Minghui Ma, Alessandra Palmigiano, Apostolos Tzimoulis, and Zhiguang Zhao. "Unified Correspondence as a Proof-Theoretic Tool". In: (2016).

Proper Multi-type Display Calculus for Lattice Logic MtD.LatL (2017)

Identity and Cut rules:

$$\frac{}{p \vdash p}\ \text{Id} \qquad \frac{X \vdash A \qquad A \vdash Y}{X \vdash Y}\ \text{Cut} \qquad \frac{\Gamma \vdash \alpha \qquad \alpha \vdash \Delta}{\Gamma \vdash \Delta}\ \text{Cut} \qquad \frac{\Pi \vdash \xi \qquad \xi \vdash \Sigma}{\Pi \vdash \Sigma}\ \text{Cut}$$

Multi-type display rules

$$\frac{\Gamma \vdash \circ X}{\bullet \Gamma \vdash X}\ \text{adj} \qquad \frac{\circ^{\text{op}} X \vdash \Pi}{X \vdash \bullet^{\text{op}} \Pi}\ \text{adj}$$

Clarifications: The language $\mathcal{L}_{\text{MT}}(\mathcal{F},\mathcal{G})$ of MtD.LatL consists of *logical* and *structural terms* in the types $\mathsf{T}_1 := \mathsf{Lattice}$, $\mathsf{T}_2 := \mathsf{Left}$ and $\mathsf{T}_3 := \mathsf{Right}$. Following the notation of {83}, the set of logical terms takes as parameters: 1) a denumerable set of atomic terms $\mathsf{At}(\mathsf{Lattice})$, elements of which are denoted p, possibly with indexes; 2) disjoint sets of connectives $\mathcal{F} := \mathcal{F}_{\mathsf{Lattice}} \uplus \mathcal{F}_{\mathsf{Left}} \uplus \mathcal{F}_{\mathsf{Right}} \uplus \mathcal{F}_{\text{MT}}$ and $\mathcal{G} := \mathcal{G}_{\mathsf{Lattice}} \uplus \mathcal{G}_{\mathsf{Left}} \uplus \mathcal{G}_{\mathsf{Right}} \uplus \mathcal{G}_{\text{MT}}$ defined as follows: $\mathcal{F}_{\mathsf{Lattice}} := \{\top\}$, $\mathcal{F}_{\mathsf{Left}} := \{\cap\}$, $\mathcal{F}_{\mathsf{Right}} := \{\cap^{\text{op}}\}$, $\mathcal{F}_{\text{MT}} := \{\blacklozenge, \Diamond^{\text{op}}\}$, where $n_\top = 0$, $n_\blacklozenge = n_{\Diamond^{\text{op}}} = 1$, $n_\cap = n_{\cap^{\text{op}}} = 2$, and $\varepsilon_\blacklozenge(1) = \varepsilon_{\Diamond^{\text{op}}}(1) = \varepsilon_\cap(i) = \varepsilon_{\cap^{\text{op}}}(i) = 1$ for every $i \in \{1,2\}$, and $\mathcal{G}_{\mathsf{Lattice}} := \{\bot\}$, $\mathcal{G}_{\mathsf{Left}} := \{\cup\}$, $\mathcal{G}_{\mathsf{Right}} := \{\cup^{\text{op}}\}$, $\mathcal{G}_{\text{MT}} := \{\blacksquare^{\text{op}}, \Box\}$, where $n_\bot = 0$, $n^{\text{op}}_\blacksquare = n_\Box = 1$, $n_\cup = n^{\text{op}}_\cup = 2$, and $\varepsilon_{\blacksquare^{\text{op}}}(1) = \varepsilon_\Box(1) = \varepsilon_\cup(i) = \varepsilon_{\cup^{\text{op}}}(i) = 1$ for every $i \in \{1,2\}$. The functional types of the heterogeneous connectives $\Box$, $\Diamond^{\text{op}}$, $\blacklozenge$ and $\blacksquare^{\text{op}}$ are $\mathsf{Lattice} \to \mathsf{Left}$, $\mathsf{Lattice} \to \mathsf{Right}$, $\mathsf{Left} \to \mathsf{Lattice}$, and $\mathsf{Right} \to \mathsf{Lattice}$ respectively.

The structural terms are built by means of structural connectives, taking logical terms as atomic structures. The set of structural connectives includes $\circ, \circ^{\text{op}}, \bullet, \bullet^{\text{op}}$ which are the structural counterparts of $\Box, \Diamond^{\text{op}}, \blacklozenge, \blacksquare^{\text{op}}$, respectively. It also includes I which is the structural counterpart of $\top$ (when occurring in precedent position) and $\bot$ (when occurring in succedent position); $\mathbf{.}$ which is the structural counterpart of $\cap$ (when occurring in precedent position) and $\cup$ (when occurring in succedent position); $\mathbf{.}^{\text{op}}$ which is the structural counterpart of $\cap^{\text{op}}$ (when occurring in precedent position) and $\cup^{\text{op}}$ (when occurring in succedent position). Finally, it includes $\supset$ and $\supset^{\text{op}}$ which, when occurring in precedent position, correspond to the left residuals of $\mathbf{.}$ and $\mathbf{.}^{\text{op}}$ respectively, and, when occurring in succedent position, correspond to the right residuals of $\mathbf{.}$ and $\mathbf{.}^{\text{op}}$ respectively, and the structural constants Ⓢ and $Ⓢ^{\text{op}}$, corresponding to the top (when occurring in precedent position) and bottom (when occurring in succedent position) of Left and Right, respectively.

Summing up, the well formed terms of MtD.LatL are generated by simultaneous induction as follows:

Lattice	Left	Right
$A ::= p \mid \blacklozenge\alpha \mid \blacksquare\xi$	$\alpha ::= \Box A \mid \alpha \cap \alpha \mid \alpha \cup \alpha$	$\xi ::= \Diamond^{\text{op}} A \mid \xi \cap^{\text{op}} \xi \mid \xi \cup^{\text{op}} \xi$
$X ::= p \mid \mathrm{I} \mid \bullet\Gamma \mid \bullet^{\text{op}}\Pi$	$\Gamma ::= \circ X \mid Ⓢ \mid \Gamma \mathbf{.} \Gamma \mid \Gamma \supset \Gamma$	$\Pi ::= \circ^{\text{op}} X \mid Ⓢ^{\text{op}} \mid \Pi \mathbf{.}^{\text{op}} \Pi \mid \Pi \supset^{\text{op}} \Pi$

The introduction rules instantiate the general template described in {83}, and hence are omitted. Also, the pure-type structural rules are the standard ones capturing distributive lattices (for the types Left and Right), and are also omitted.

History: Lattice logic is the $\{\wedge, \vee, \top, \bot\}$-fragment of classical propositional logic without distributivity. In [1], a display calculus is introduced for lattice logic, regarded as the 'additive fragment' of linear logic. In this calculus, no structural counterparts are assigned to $\wedge$ and $\vee$, and the introduction rules of these connectives are given in the so called *additive* form, which subsumes the usual weakening, associativity, exchange and contraction rules. In [2], Birkhoff's representation theorem for complete lattices provides the guidelines for the design of the proper multi-type display calculus MtD.LatL.

Entry 91 by: Giuseppe Greco

Remarks: In MtD.LatL, all connectives are introduced by means of standard rules as discussed in {83}, which occur in the so called *multiplicative* form and do not subsume weakening, associativity, exchange and contraction. The calculus above is sound and complete w.r.t. complete lattices, equivalently presented as heterogeneous algebras as discussed in [2]; it is conservative, and enjoys the cut elimination and subformula property as immediate consequences of the general theory of multi-type calculi.

---

[1] Nuel Belnap. "Linear Logic Displayed". In: *Notre Dame J of Formal Logic* 31.1 (1990), pp. 14–25.

[2] Giuseppe Greco and Alessandra Palmigiano. "Lattice logic properly displayed". In: *Logic, Language, Information and Computation, 24th Workshop, WoLLIC. Proceedings.* Ed. by Ruy J.G.B. de Queiroz Juliette Kennedy. Vol. 10388. 2017, pp. 153–169.

Proper Multi-type Display Calculus for semi De Morgan Logic MtD.SDM (2017)

Identity and Cut rules:

$$\frac{}{p \vdash p}\ \text{Id} \qquad \frac{X \vdash A \qquad A \vdash Y}{X \vdash Y}\ \text{Cut} \qquad \frac{\Gamma \vdash \alpha \qquad \alpha \vdash \Delta}{\Gamma \vdash \Delta}\ \text{Cut}$$

Display postulates and structural rule for pure DeMorgan type

$$\text{adj}\ \frac{\tilde{*}\Gamma \vdash \Delta}{\tilde{*}\Delta \vdash \Gamma} \qquad \frac{\Gamma \vdash \tilde{*}\Delta}{\Delta \vdash \tilde{*}\Gamma}\ \text{adj} \qquad \frac{\Gamma \vdash \Delta}{\tilde{*}\Delta \vdash \tilde{*}\Gamma}\ \text{cont}$$

Multi-type display postulates and structural rules

$$\text{adj}\ \frac{X \vdash \breve{\Box}\Gamma}{\hat{\blacklozenge}X \vdash \Gamma} \qquad \frac{\tilde{\circ}X \vdash \Gamma}{X \vdash \tilde{\bullet}\Gamma}\ \text{adj}$$

$$\tilde{\circ}\ \frac{X \vdash Y}{\tilde{\circ}X \vdash \tilde{\circ}Y} \qquad \tilde{\bullet}\ \frac{\tilde{\bullet}\Gamma \vdash \tilde{\bullet}\Delta}{\Gamma \vdash \Delta} \qquad \tilde{\circ}\breve{\Box}\ \frac{\Gamma \vdash \tilde{\circ}\breve{\Box}\Delta}{\Gamma \vdash \Delta} \qquad \hat{\blacklozenge}\hat{1}\ \frac{\hat{1} \vdash \Gamma}{\hat{\blacklozenge}\hat{\top} \vdash \Gamma} \qquad \breve{\Box}\breve{0}\ \frac{X \vdash \breve{\Box}\breve{0}}{X \vdash \hat{\bot}}$$

Clarifications: The language $\mathcal{L}_{\mathrm{MT}}(\mathcal{F},\mathcal{G})$ of MtD.SDM consists of *logical* and *structural terms* in the types $\mathsf{T}_1 :=$ DeMorgan and $\mathsf{T}_2 :=$ Distributive. Following the notation of {83}, the set of logical terms takes as parameters: 1) a denumerable set of atomic terms At(Distributive), elements of which are denoted p, possibly with indexes; 2) sets of connectives $\mathcal{F} := \mathcal{F}_{\mathsf{DeMorgan}} \uplus \mathcal{F}_{\mathsf{Distributive}} \uplus \mathcal{F}_{\mathrm{MT}}$ and $\mathcal{G} := \mathcal{G}_{\mathsf{DeMorgan}} \uplus \mathcal{G}_{\mathsf{Distributive}} \uplus \mathcal{G}_{\mathrm{MT}}$ defined as follows: $\mathcal{F}_{\mathsf{DeMorgan}} := \{\cap, 1, \sim\}$, $\mathcal{F}_{\mathsf{Distributive}} := \{\wedge, \top\}$, $\mathcal{F}_{\mathrm{MT}} := \{\diamond\}$, where $n_1 = n_\top = 0$, $n_\sim = n_\diamond = 1$, and $n_\cap = n_\wedge = 2$, and $\varepsilon_\sim(1) = \partial$, and $\varepsilon_\diamond(1) = \varepsilon_\cap(i) = \varepsilon_\wedge(i) = 1$ for every $i \in \{1,2\}$, and $\mathcal{G}_{\mathsf{DeMorgan}} := \{\cup, 0, \neg\}$, $\mathcal{G}_{\mathsf{Distributive}} := \{\vee, \bot\}$, $\mathcal{G}_{\mathrm{MT}} := \{\Box, \boxdot\}$, where $n_0 = n_\bot = 0$, $n_\cup = n_\vee = 2$, and $n_\neg = n_\Box = n_\boxdot = 1$, and $\varepsilon_\neg(1) = \partial$, $\varepsilon_\Box(1) = \varepsilon_\boxdot(1) = \varepsilon_\cup(i) = \varepsilon_\vee(i) = 1$ for every $i \in \{1,2\}$. The functional type of the heterogeneous connective $\Box$ is DeMorgan $\to$ Distributive, while functional type of the heterogeneous connectives $\diamond$ and $\boxdot$ is Distributive $\to$ DeMorgan.

The structural terms are built by means of structural connectives, taking logical terms as atomic structures. The set of structural connectives includes $\hat{\top}$, $\hat{\bot}$, $\hat{\wedge}$, $\check{\vee}$, $\hat{1}$, $\check{0}$, $\hat{\cap}$, $\check{\cup}$, $\breve{\Box}$ which are the structural counterparts of $\top, \bot, \wedge, \vee, 1, 0, \cap, \cup, \Box$, respectively. It also includes $\tilde{*}$ as the structural counterpart of both $\sim$ (when occurring in antecedent position) and $\neg$ (when occurring in succedent position) and $\tilde{\circ}$ as structural counterpart of both $\diamond$ (when occurring in antecedent position) and $\boxdot$ (when occurring in succedent position). Finally, it includes $\hat{\blacklozenge}$, $\tilde{\bullet}$, $\mathrel{\hat{>}}$, $\mathrel{\check{\to}}$, $\mathrel{\hat{\supset}}$, $\mathrel{\check{\multimap}}$, where $\hat{\blacklozenge}$ is the left adjoint of $\breve{\Box}$, and $\mathrel{\hat{>}}$ and $\mathrel{\hat{\supset}}$ are the left residuals of $\check{\vee}$ and $\check{\cup}$ respectively, and $\mathrel{\check{\to}}$ and $\mathrel{\check{\multimap}}$ are the right residuals of $\hat{\wedge}$ and $\hat{\cap}$ respectively, and $\tilde{\bullet}$ is right and left adjoint of $\tilde{\circ}$. Hence, the functional type of $\tilde{\bullet}$ is DeMorgan $\to$ Distributive, while the functional type of $\hat{\blacklozenge}$ is Distributive $\to$ DeMorgan.

Summing up, the well formed terms of MtD.SDM are generated by simultaneous induction as follows:

DeMorgan	Distributive
$\alpha ::= 1 \mid 0 \mid \diamond A \mid \boxdot A \mid {\sim}\alpha \mid \neg\alpha \mid \alpha \cap \alpha \mid \alpha \cup \alpha$	$A ::= p \mid \top \mid \bot \mid \Box\alpha \mid A \wedge A \mid A \vee A$
$\Gamma ::= \hat{1} \mid \check{0} \mid \hat{\blacklozenge}X \mid \tilde{\circ}X \mid \tilde{*}\Gamma \mid \Gamma \hat{\cap} \Gamma \mid \Gamma \check{\cup} \Gamma \mid \Gamma \mathrel{\hat{\supset}} \Gamma \mid \Gamma \mathrel{\check{\multimap}} \Gamma$	$X ::= \hat{\top} \mid \hat{\bot} \mid \breve{\Box}\Gamma \mid \tilde{\bullet}\Gamma \mid X \hat{\wedge} X \mid X \check{\vee} X \mid X \mathrel{\hat{>}} X \mid X \mathrel{\check{\to}} X$

Entry 92 by: Fei Liang

The introduction rules instantiate the general template described in {83}, and hence are omitted. Also, the pure-type structural rules are the standard ones capturing negation-free classical propositional logic (for Distributive) and De Morgan logic (for DeMorgan) and are also omitted.

History: Semi De Morgan logic, introduced in an algebraic setting by H.P. Sankappanavar [1], is designed to capture the salient features of intuitionistic negation in a paraconsistent setting. In [4] a cut-free G3-style sequent calculus is introduced for semi De Morgan logic, which includes introduction rules under the scope of structural negation. In [3], the algebraic semantics for semi De Morgan logic is recast in a multi-type algebraic framework which provides the guidelines for the design of the proper multi-type display calculus MtD.SDM.

Remarks: Every known axiomatization of semi De Morgan logic is not analytic inductive according to the definition of [2]. Hence, semi De Morgan logic cannot be captured by a single-type proper display calculus on the basis of any known axiomatization. The calculus above is sound and complete w.r.t. the equivalent heterogeneous presentation of semi De Morgan algebras introduced in [3]; it is conservative, and enjoys the cut elimination and subformula property as immediate consequences of the general theory of multi-type calculi.

---

[1] Sankappanavar Hanamantagouda P. "Semi-De Morgan algebras". In: *The Journal of symbolic logic* 52.03 (1987), pp. 712–724.

[2] Giuseppe Greco, Minghui Ma, Alessandra Palmigiano, Apostolos Tzimoulis, and Zhiguang Zhao. "Unified Correspondence as a Proof-Theoretic Tool". In: *Journal of Logic and Computation* (2016).

[3] Giuseppe Greco, Fei Liang, M Andrew Moshier, and Alessandra Palmigiano. "Multi-type display calculus for semi De Morgan logic". In: *International Workshop on Logic, Language, Information, and Computation*. Springer. 2017, pp. 199–215.

[4] Minghui Ma and Fei Liang. "Sequent Calculi for Semi-De Morgan and De Morgan Algebras". In: *ArXiv preprint :1611.05231* (Submitted).

Epsilon-Sound Sequent Calculus LJ* (2017)

$$\frac{\Gamma_1, A \vdash F \quad \Gamma_2, B \vdash F}{\Gamma_1, \Gamma_2, A \vee B \vdash F}\ \vee_l \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee_r^1 \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee_r^2 \qquad \frac{\Gamma \vdash A}{\Gamma, \neg A \vdash}\ \neg_l \qquad \frac{\Gamma, A \vdash}{\Gamma \vdash \neg A}\ \neg_r$$

$$\frac{\Gamma, A, B \vdash F}{\Gamma, A \wedge B \vdash F}\ \wedge_l \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2 \vdash B}{\Gamma_1, \Gamma_2 \vdash A \wedge B}\ \wedge_r \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2, B \vdash F}{\Gamma_1, \Gamma_2, A \to B \vdash F}\ \to_l \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \to_r$$

$$\frac{\Gamma \vdash F}{\Gamma, A \vdash F}\ w_l \qquad \frac{\Gamma \vdash}{\Gamma \vdash A}\ w_r \qquad \frac{\Gamma, A, A \vdash F}{\Gamma, A \vdash F}\ c_l \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2, A \vdash F}{\Gamma_1, \Gamma_2 \vdash F}\ cut \qquad \frac{}{A[\nu_x^l\ F] \vdash A[\nu_x^l\ F]}\ a$$

$$\frac{\Gamma \vdash A[\alpha]}{\Gamma \vdash \forall x.A[x]}\ \forall_r \qquad \frac{\Gamma, A[\alpha] \vdash F}{\Gamma, \exists x.A[x] \vdash F}\ \exists_l \qquad \frac{\Gamma, A[t] \vdash F}{\Gamma, \forall x.A[x] \vdash F}\ \forall'_l \qquad \frac{\Gamma \vdash A[t]}{\Gamma \vdash \exists x.A[x]}\ \exists'_r$$

ν denotes the binders ε or τ. The term t must be accessible in the conclusion sequent (*accessibility condition*). Accessible occurrences of t or any of its ε-subterms in Γ and F must have a constant as a label (*label condition*). l is a constant in a (*initial condition*).

Clarifications: A term t is *accessible* in a formula F iff at least one of the following two conditions hold:

- for any top-level (i.e., not nested inside another ε-term) ε-term $\nu_x\ G$ in t, it is the case that $F[\nu_x\ G \rightsquigarrow x]$ is a sub-formula of G ($\rightsquigarrow$ denotes term rewriting); or
- t contains a nested ε-term $\nu_y\ H$ such that $\nu_y\ H$ is accessible in F and $t[\nu_y\ H \rightsquigarrow y]$ is accessible in $F[\nu_y\ H \rightsquigarrow y]$.

t is accessible in a sequent S iff all top-level ε-terms in t are accessible in some formula occurring in S. ε-terms replace strong quantifiers in formulas (i.e., the ones that require eigenvariables). Intuitively, term accessibility corresponds to the availability of the eigenvariable in a proof of the non-epsilonized formula. Labels on ε-terms make proof epsilonization injective and allow proofs to be de-epsilonized.

History: Skolemization is known to be unsound in intuitionistic logic. *Epsilonization* is similar to skolemization, but replaces strongly quantified variables by ε-terms, instead of Skolem terms. **LJ*** was introduced in [1] as a sequent calculus for intuitionistic logic where *epsilonization* is sound: if the epsilonization of a sequent S is derivable in **LJ***, then S is derivable in **LJ**. This is achieved by restricting the rules $\forall_l$, $\exists_r$ and initial from **LJ** {7} to take into account information available in ε-terms.

Remarks: **LJ*** is sound and complete with respect to **LJ** {7} for ε-free formulas. A procedure for de-epsilonizing **LJ*** proofs, resulting in valid **LJ** proofs, is defined in [1].

[1] Giselle Reis and Bruno Woltzenlogel Paleo. "Epsilon Terms in Intuitionistic Sequent Calculus". In: *The International Federation of Computational Logic (IFCoLog) Journal of Logic and its Applications* 4 (2 Mar. 2017), pp. 401–423. ISSN: 2055-3714.

Entry 93 by: Giselle Reis, Bruno Woltzenlogel Paleo

Sequent Calculus for Logic of Partial Quasiary Predicates (2017)

Sequent rules for propositional compositions:

$$\vee_L \frac{\Phi,\Gamma\longrightarrow\Delta \qquad \Psi,\Gamma\longrightarrow\Delta}{\Phi\vee\Psi,\Gamma\longrightarrow\Delta}\ ; \qquad \vee_R \frac{\Gamma\longrightarrow\Phi,\Psi,\Delta}{\Gamma\longrightarrow\Phi\vee\Psi,\Delta}\ ;$$

$$\neg_L \frac{\Gamma\longrightarrow\Phi,\Delta}{\neg\Phi,\Gamma\longrightarrow\Delta}\ ; \qquad \neg_R \frac{\Phi,\Gamma\longrightarrow\Delta}{\Gamma\longrightarrow\neg\Phi,\Delta}\ .$$

Sequent rules for renomination compositions ($C_{\mathrm{R}\exists}$ is $u\in fu(R^{\bar{v}}_{\bar{x}}(\exists y\Phi))$, C_{RU} is $y\in fu(\Phi)$):

$$\mathrm{R}\vee_L \frac{R^{\bar{v}}_{\bar{x}}(\Phi)\vee R^{\bar{v}}_{\bar{x}}(\Psi),\Gamma\longrightarrow\Delta}{R^{\bar{v}}_{\bar{x}}(\Phi\vee\Psi),\Gamma\longrightarrow\Delta}\ ; \qquad \mathrm{R}\vee_R \frac{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}(\Phi)\vee R^{\bar{v}}_{\bar{x}}(\Psi),\Delta}{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}(\Phi\vee\Psi),\Delta}\ ;$$

$$\mathrm{R}\neg_L \frac{\neg R^{\bar{v}}_{\bar{x}}(\Phi),\Gamma\longrightarrow\Delta}{R^{\bar{v}}_{\bar{x}}(\neg\Phi),\Gamma\longrightarrow\Delta}\ ; \qquad \mathrm{R}\neg_R \frac{\Gamma\longrightarrow\neg R^{\bar{v}}_{\bar{x}}(\Phi),\Delta}{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}(\neg\Phi),\Delta}\ ;$$

$$\mathrm{RR}_L \frac{R^{\bar{v}}_{\bar{x}}\circ^{\bar{w}}_{\bar{y}}(\Phi),\Gamma\longrightarrow\Delta}{R^{\bar{v}}_{\bar{x}}(R^{\bar{w}}_{\bar{y}}(\Phi)),\Gamma\longrightarrow\Delta}\ ; \qquad \mathrm{RR}_R \frac{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}\circ^{\bar{w}}_{\bar{y}}(\Phi),\Delta}{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}(R^{\bar{w}}_{\bar{y}}(\Phi)),\Delta}\ ;$$

$$\mathrm{R}\exists_L \frac{\exists u R^{\bar{v}}_{\bar{x}}R^{y}_{u}(\Phi),\Gamma\longrightarrow\Delta}{R^{\bar{v}}_{\bar{x}}(\exists y\Phi),\Gamma\longrightarrow\Delta}\ ,C_{\mathrm{R}\exists}; \qquad \mathrm{R}\exists_R \frac{\Gamma\longrightarrow\exists u R^{\bar{v}}_{\bar{x}}R^{y}_{u}(\Phi),\Delta}{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}(\exists y\Phi),\Delta}\ ,C_{\mathrm{R}\exists};$$

$$\mathrm{R}\varepsilon s_L \frac{\varepsilon z,\Gamma\longrightarrow\Delta}{R^{\bar{v}}_{\bar{x}}(\varepsilon z),\Gamma\longrightarrow\Delta}\ ,z\notin\bar{v}; \qquad \mathrm{R}\varepsilon s_R \frac{\Gamma\longrightarrow\varepsilon z,\Delta}{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}(\varepsilon z),\Delta}\ ,z\notin\bar{v};$$

$$\mathrm{R}\varepsilon_L \frac{\varepsilon y,\Gamma\longrightarrow\Delta}{R^{\bar{v},z}_{\bar{x},y}(\varepsilon z),\Gamma\longrightarrow\Delta}\ ; \qquad \mathrm{R}\varepsilon_R \frac{\Gamma\longrightarrow\varepsilon y,\Delta}{\Gamma\longrightarrow R^{\bar{v},z}_{\bar{x},y}(\varepsilon z),\Delta}\ ;$$

$$\mathrm{R}_L \frac{\Phi,\Gamma\longrightarrow\Delta}{R(\Phi),\Gamma\longrightarrow\Delta}\ ; \qquad \mathrm{R}_R \frac{\Gamma\longrightarrow\Phi,\Delta}{\Gamma\longrightarrow R(\Phi),\Delta}\ ;$$

$$\mathrm{RI}_L \frac{R^{\bar{v}}_{\bar{x}}(\Phi),\Gamma\longrightarrow\Delta}{R^{z,\bar{v}}_{z,\bar{x}}(\Phi),\Gamma\longrightarrow\Delta}\ ; \qquad \mathrm{RI}_R \frac{\Gamma\longrightarrow R^{\bar{v}}_{\bar{x}}(\Phi),\Delta}{\Gamma\longrightarrow R^{z,\bar{v}}_{z,\bar{x}}(\Phi),\Delta}\ ;$$

$$\mathrm{RU}_L \frac{R^{\bar{v}}_{\bar{u}}(\Phi),\Gamma\longrightarrow\Delta}{R^{y,\bar{v}}_{z,\bar{u}}(\Phi),\Gamma\longrightarrow\Delta}\ ,C_{\mathrm{RU}}; \qquad \mathrm{RU}_R \frac{\Gamma\longrightarrow R^{\bar{v}}_{\bar{u}}(\Phi),\Delta}{\Gamma\longrightarrow R^{y,\bar{v}}_{z,\bar{u}}(\Phi),\Delta}\ ,C_{\mathrm{RU}}.$$

Sequent rules for quantification compositions:

$$\exists\mathrm{E}_L \frac{R^{x}_{z}(\Phi),\Gamma\longrightarrow\varepsilon z,\Delta}{\exists x\Phi,\Gamma\longrightarrow\Delta}\ ,z\in fu(\exists x\Phi,\Gamma,\Delta); \qquad \exists\mathrm{E}1_R \frac{\Gamma\longrightarrow R^{x}_{y}(\Phi),\exists x\Phi,\varepsilon y,\Delta}{\Gamma\longrightarrow\exists x\Phi,\varepsilon y,\Delta}\ ;$$

$$\exists\mathrm{E}2_R \frac{\Gamma\longrightarrow R^{x}_{z}(\Phi),\varepsilon z,\exists x\Phi,\Delta}{\Gamma\longrightarrow\exists x\Phi,\Delta}\ ,\varepsilon(\Delta)=\emptyset,z\in fu(\exists x\Phi,\Gamma,\Delta);$$

$$\exists\mathrm{E}3_R \frac{\varepsilon y,\Gamma\longrightarrow\exists x\Phi,\Delta \qquad \Gamma\longrightarrow R^{x}_{y}(\Phi),\varepsilon y,\exists x\Phi,\Delta}{\Gamma\longrightarrow\exists x\Phi,\Delta}\ ,y\in uns(\Gamma\longrightarrow\Delta).$$

Rule $\exists\mathrm{E}1_R$ is applied when at least one variable is assigned. Rule $\exists\mathrm{E}2_R$ is applied when there are no assigned variables (in this case a fresh unassigned variables is assigned). This means the first application of quantification elimination (therefore $\varepsilon(\Delta)=\emptyset$). Rule $\exists\mathrm{E}3_R$ is applied when an unspecified variable is involved into quantifier elimination. In this case two branches appear: with this variable being unassigned and assigned.

Entry 94 by: Mykola Nikitchenko, Stepan Skilniak

Clarifications: Quasiary predicates are predicates over partial assignments of variables. Such predicates do not have fixed arity and generalize n-ary predicates. This specific feature of quasiary predicates requires additional operations (compositions): renomination $R^{v_1,\dots,v_n}_{x_1,\dots,x_n}$ ($R^{\bar{v}}_{\bar{x}}$) and variable unassignment predicate εx. Thus, the language of the logic is defined by Kleene-like compositions: disjunction $\vee$, negation $\neg$, existential quantifier $\exists x$, $R^{\bar{v}}_{\bar{x}}$, and εx. Quasiary predicates are sensitive to unassigned variables therefore rules of sequent calculus classify variables as assigned, unassigned, and unspecified. To make correct transformations of formulas from Γ, unessential variables (analog of fresh variables in classical logic) are required; their set is denoted $fu(\Gamma)$.

History: Quasiary predicates (and functions) represent semantics of programs and their components. Thus, logics of quasiary predicates are program-oriented logics. Various types of such logics are described in [1, 4]. The presented QE-calculus, proof of its soundness and completeness can be found in [5]. Hoare-like program logic based on partial quasiary predicates is described in [3]. Satisfiability-preserving translation of formulas of quasiary logic to formulas of classical first-order logic is formulated in [2].

---

[1] Mykola Nikitchenko and Stepan Shkilniak. *Mathematical logic and theory of algorithms*. (In Ukrainian). Publishing house of Taras Shevchenko National University of Kyiv, 2008, p. 528.

[2] Mykola Nikitchenko and Valentin Tymofieiev. "Satisfiability in Composition-Nominative Logics". In: *Central European Journal of Computer Science* 2.3 (2012), pp. 194–213.

[3] Andrii Kryvolap, Mykola Nikitchenko, and Wolfgang Schreiner. "Extending Floyd-Hoare logic for partial pre- and postconditions". In: *Information and Communication Technologies in Education, Research, and Industrial Applications, 9th International Conference, ICTERI 2013, Kherson, Ukraine, June 19-22, 2013, Revised Selected Papers*. Vol. 412. Communications in Computer and Information Science. Springer, 2013, pp. 355–378.

[4] Mykola Nikitchenko and Stepan Shkilniak. *Applied Logic*. (In Ukrainian). Publishing house of Taras Shevchenko National University of Kyiv, 2013, p. 278.

[5] Mykola Nikitchenko and Stepan Shkilniak. "Algebras and Logics of Partial Quasiary Predicates". In: *Algebra and Discrete Mathematics* 23.2 (2017), pp. 263–278.

Part II
Indexes

Proof Systems Grouped by Logics

Ancestral Logic
- Higher-Order
  - Sequent Calculus TC, 63

Classical Logic
- First-Order, 60
  - (Unfailing) Completion, 20
  - $\mathbf{LK}_{\mu\tilde{\mu}}$ − **tree**, 72
  - Bledsoe's Natural Deduction, 25
  - Cancellative Superposition, 54
  - Classical Sequent Calculus, 9
  - Conflict Resolution, 103
  - Constraint Superposition, 42
  - Epsilon Calculus, 11
  - Expansion Proofs, 24
  - Focused LK, 78
  - G3c, 53
  - Hierarchic Superposition, 43
  - Hilbert and Ackermann's Calculus, 7
  - Hilbert's Axiomatic Calculus, 5
  - Kleene's Classical **G3**, 12
  - LambdaMu, 45
  - Model Evolution, 66
  - Natural Knowledge Bases, 26
  - Ordered Resolution, 18
  - Paramodulation, 19
  - Principia Mathematica, 4
  - Socratic Proofs for FOL, 69
- Higher-Order
  - Extensional HO RUE-Resolution, 77
- Propositional
  - **LC**, 38
  - Bernays's Propositional Calculus, 6
  - Erotetic Dual Resolution for Classical Propositional Logic, 96
  - Socratic Proofs for CPL, 68
  - Synthetic Tableaux, 57
- Quantified Propositional
  - IR, 93

Conditional Logics
- Propositional
  - Conditional Labelled Sequent Calculi SeqS, 73
  - Conditional Nested Sequent Calculi $\mathcal{NS}$, 89

Counterfactual Logics
- Propositional
  - Counterfactual Sequent Calculi I, 87
  - Counterfactual Sequent Calculi II, 88

Finite-Valued Logic
- Propositional
  - Signed Analytic Calculi for Finite-Valued Logics, 34

General
- Propositional
  - Multi-type Sequent Calculi Mt.SC, 99

Inquisitive Logic
- Propositional
  - Proper Multi-type Display Calculus for Inquisitive Logic MtD.InqL, 107

Institution
- General
  - Entailment for Structured Specifications, 31
  - Refinement of Structured Specifications, 39

Intuitionistic Logic
- First-Order
  - Focused LJ, 80
  - Intuitionistic Sequent Calculus, 10
  - Intuitionistic Sequent Calculus with ε-terms, 114
  - Kleene's Intuitionistic **G3**, 13
  - Multi-Conclusion Intuitionistic Sequent Calculus, 14
  - Natural Deduction, 8
- Propositional
  - Full Intuitionistic Logic, 50
  - LambdaBar, 49

Intuitionistic Modal Logics
- Higher-Order
  - Natural Deduction for Modal Logic **K**, 100
- Propositional
  - Intuitionistic Hybrid Logic, 64

Lattice Logic
- Propositional
  - Proper Multi-type Display Calculus for Lattice Logic MtD.LatL, 111

Linear Logics
- Propositional
  - Full Intuitionistic Linear Logic, 33
  - Full Intuitionistic Linear Logic - Deep Nested Sequent Calculus, 91
  - Intuitionistic Linear Logic, 27
  - Linear Sequent Calculus, 28
  - Polarized Linear Sequent Calculus, 58
  - Proof Nets for $\mathbf{MLL}^{-}$, 30
  - Two-sided Linear Sequent Calculus, 41

Logic of Partial Quasiary Predicates
- First-order
  - Sequent Calculus for Logic of Partial Quasiary Predicates, 116

Modal Logics
- Propositional
  - Constructive Modal Logic S4, 61
  - Graph-based tableaux for modal logics, 56
  - Hybrid Logic, 62
  - Multi-type Sequent Calculus for Dynamic Epistemic Logic MtD.DEL, 109
  - Polynomial Ring Calculus with Operators, 102
  - Resolution for Modal Logic K, 23
  - Sequent Calculus for Superintuitionistic Modal Logic KM-lin, 95
  - Socratic Proofs for Modal Propositional K, 70
  - Socratic Proofs for Modal Propositional Logics, 71

Negative Modal Logics
- Propositional
  - Sequent Systems for Negative Modalities, 104

Paraconsistent Logics
- First-Order
  - Logic of Epistemic Inconsistency, 47
- Propositional
  - Erotetic Dual Resolution for mbC, 97
  - Sequent Calculi for Paraconsistent Logics, 86

Preferential Logics
- Propositional

Preferential Tableau Calculi $\mathcal{T}P^{\mathbf{T}}$, 74

Semi De Morgan Logic
Propositional
Proper Multi-type Display Calculus for semi De Morgan Logic MtD.SDM, 113
Substructural Logics
Propositional
Lambek Calculus, 15
Update Logic, 105

Temporal Logics
First-Order
Ordered Fine-Grained Resolution with Selection, 83
Resolution for First-Order Temporal Logic, 65, 67
Propositional
Resolution for Computational Tree Logic, 84
Resolution for Propositional Linear Time Temporal Logic, 40
Type Theory
First-Order
LambdaPiModulo, 82
Higher-Order
LF with Coercive Subtyping, 52
Pure Type Systems, 32
Typed LF for Type Theories, 48
Quantified Propositional
Second Order λ-Calculus (System **F**), 21

Unification
First-Order
First-Order Unification, 17
Higher-Order
Higher-Order Pre-Unification, 22

Proof Systems Grouped by Type

Axiomatic
- Bernays's Propositional Calculus, 6
- Entailment for Structured Specifications, 31
- Epsilon Calculus, 11
- Hilbert and Ackermann's Calculus, 7
- Hilbert's Axiomatic Calculus, 5
- Refinement of Structured Specifications, 39

Display Calculus
- Update Logic, 105

Hilbert-Style
- Principia Mathematica, 4

Hybrid
- Model Evolution, 66

Matrix-Based
- Expansion Proofs, 24

Meta-Calculus
- Saturation With Redundancy, 37

Multi-type Sequent Calculus
- Multi-type Sequent Calculi Mt.SC, 99
- Multi-type Sequent Calculus for Dynamic Epistemic Logic MtD.DEL, 109
- Proper Multi-type Display Calculus for Inquisitive Logic MtD.InqL, 107
- Proper Multi-type Display Calculus for Lattice Logic MtD.LatL, 111
- Proper Multi-type Display Calculus for semi De Morgan Logic MtD.SDM, 113

Natural Deduction
- Bledsoe's Natural Deduction, 25
- Conflict Resolution, 103
- Contextual Natural Deduction, 92
- Hybrid Logic, 62
- Intuitionistic Hybrid Logic, 64
- LambdaMu, 45
- LambdaPiModulo, 82
- LF with Coercive Subtyping, 52
- Logic of Epistemic Inconsistency, 47
- Natural Deduction, 8
- Natural Deduction for Modal Logic **K**, 100
- Natural Knowledge Bases, 26
- Pure Type Systems, 32
- Second Order λ-Calculus (System **F**), 21
- Typed LF for Type Theories, 48

Nested Sequent Calculus
- Full Intuitionistic Linear Logic - Deep Nested Sequent Calculus, 91

Polynomial Ring Calculi
- Polynomial Ring Calculus with Operators, 102

Proof Net
- Proof Nets for $\mathbf{MLL}^{-}$, 30

Resolution
- Erotetic Dual Resolution for Classical Propositional Logic, 96
- Erotetic Dual Resolution for mbC, 97
- Extensional HO RUE-Resolution, 77
- IR, 93
- Ordered Fine-Grained Resolution with Selection, 83
- Ordered Resolution, 18
- Paramodulation, 19
- Resolution, 16
- Resolution for Computational Tree Logic, 84
- Resolution for First-Order Temporal Logic, 65, 67
- Resolution for Modal Logic K, 23
- Resolution for Propositional Linear Time Temporal Logic, 40

Sequent Calculus, 60
- **LC**, 38
- $\mathbf{LK}_{\mu\tilde{\mu}}$ − **tree**, 72
- Classical Sequent Calculus, 9
- Conditional Labelled Sequent Calculi SeqS, 73
- Conditional Nested Sequent Calculi $\mathcal{NS}$, 89
- Constructive Modal Logic S4, 61
- Counterfactual Sequent Calculi I, 87
- Counterfactual Sequent Calculi II, 88
- Focused LJ, 80
- Focused LK, 78
- Full Intuitionistic Linear Logic, 33
- Full Intuitionistic Logic, 50
- G3c, 53
- Intuitionistic Linear Logic, 27
- Intuitionistic Sequent Calculus, 10
- Intuitionistic Sequent Calculus with ε-terms, 114
- Kleene's Classical **G3**, 12
- Kleene's Intuitionistic **G3**, 13
- LambdaBar, 49
- Lambek Calculus, 15
- Linear Sequent Calculus, 28
- Multi-Conclusion Intuitionistic Sequent Calculus, 14
- Polarized Linear Sequent Calculus, 58
- Sequent Calculi for Paraconsistent Logics, 86
- Sequent Calculus for Logic of Partial Quasiary Predicates, 116
- Sequent Calculus for Superintuitionistic Modal Logic KMlin, 95
- Sequent Calculus TC, 63
- Sequent Systems for Negative Modalities, 104
- Signed Analytic Calculi for Finite-Valued Logics, 34
- Two-sided Linear Sequent Calculus, 41
- Untyped Lambda Reduction, 85

Socratic Proof System
- Socratic Proofs for CPL, 68
- Socratic Proofs for FOL, 69
- Socratic Proofs for Modal Propositional K, 70
- Socratic Proofs for Modal Propositional Logics, 71

Superposition
- (Unfailing) Completion, 20
- Cancellative Superposition, 54
- Constraint Superposition, 42
- Hierarchic Superposition, 43
- Superposition, 36

Tableau
- Graph-based tableaux for modal logics, 56
- Preferential Tableau Calculi $\mathcal{T}P^{\mathbf{T}}$, 74
- Synthetic Tableaux, 57

Unification
- First-Order Unification, 17
- Higher-Order Pre-Unification, 22

Entry Authors

Ali Assaf, 32
Ana Teresa Martins, 46
Andrzej Tarlecki, 31, 39
Anna Zamansky, 86
Arnon Avron, 63

Bernard Linsky, 3
Björn Lellmann, 12, 13, 53, 87, 88
Boris Konev, 40, 65, 67
Bruno Woltzenlogel Paleo, 8, 92, 100, 103, 114

Camilla Schwind, 73
Christoph Benzmüller, 75, 76
Clare Dixon, 40, 65, 67, 84

Dale Miller, 24, 78, 79
Dominique Pastre, 25, 26
Donald Sannella, 31, 39
Dorota Leszczyńska-Jasion, 57, 68–71, 96, 97

Elaine Pimentel, 41

Fei Liang, 112
Francicleber Ferreira, 46

Georg Moser, 11
Gian Luca Pozzato, 73, 74, 89
Gianluigi Bellin, 29
Giselle Reis, 10, 14, 114
Giuseppe Greco, 98, 106, 108, 110
Giuseppe Longo, 21
Guillaume Aucher, 105

Harley Eades III, 15, 33, 41, 50, 61, 64
Hugo Herbelin, 44, 49, 59, 72

Joseph Boudou, 23, 27, 55

Kathleen Milsted, 21
Kenji Miyamoto, 11

Laura Giordano, 74
Leroy Chew, 93
Liron Cohen, 63
Luis Fariñas del Cerro, 23

Martin Riener, 9
Martin Wirsing, 31, 39
Michael Fisher, 40, 65, 67
Michael Gabbay, 85
Mikoláš Janota, 93
Mykola Nikitchenko, 115

Nicola Olivetti, 73, 74, 89

Olivier Gasquet, 55
Olivier Laurent, 28, 38, 58

Peter Baumgartner, 66

Rajeev Goré, 90, 94
Reiner Hähnle, 34
Richard Zach, 5–7
Rolf Hennicker, 31, 39
Ronan Saillard, 81

Sergei Soloviev, 21, 27, 48, 51
Stepan Skilniak, 115

Tomer Libal, 17, 22
Torben Braüner, 62, 64

Ullrich Hustadt, 40, 65, 67, 83, 84
Uwe Waldmann, 16, 18–20, 35, 37, 42, 43, 54

Valentina Gliozzi, 74
Valeria de Paiva, 12–15, 33, 50, 61, 64

Walter Carnielli, 101

Yoni Zohar, 86, 104

Zhaohui Luo, 48, 51

Proof Systems' Authors

A.S. Troelstra , 53
Albert Rubio , 42
Alessandra Palmigiano , 99, 107, 109, 111, 113
Alexander Kurz , 99, 109
Alfred N. Whitehead , 4
Alwen Tiu , 91
Ana Teresa Martins , 47
Anatoli Degtyarev , 65, 67
Andreas Herzig , 56
Andrzej Tarlecki , 31
Andrzej Wiśniewski , 68, 69
Anna Zamansky , 86
Apostolos Tzimoulis , 99
Arnon Avron , 63, 86
Barendregt , 32
Beata Konikowska , 86
Berardi , 32
Bertrand A. Russel , 4
Boris Konev , 40, 65, 67
Bruno Woltzenlogel Paleo , 92, 100, 103, 114
Cesare Tinelli , 66
Chad Brown , 75
Christoph Benzmüller , 75, 77
Chuck Liang , 78, 80
Clare Dixon , 40, 84
Curien , 60
Dale Miller , 24, 78, 80
David Hilbert , 5, 7, 11
de Swart , 87
Denis Cousineau , 82
Dominique Pastre , 26
Don Sannella , 31
Donald E. Knuth , 20
Dorota Leszczyńska-Jasion , 70, 71, 96, 97
Fan Yang , 107
Fei Liang , 113
Felipe Ferreira de Morais , 47
Gavin Bierman , 61
Gent , 87
George Robinson , 19
Gerhard Karl Erich Gentzen , 8–10
Geuvers , 32
Gian Luca Pozzato , 73, 74, 89
Gilles Dowek , 82
Giselle Reis , 114
Giuseppe Greco , 99, 107, 109, 111, 113
Guillaume Aucher , 105
Grard Pierre Huet , 22
H. Schwichtenberg , 53
Harald Ganzinger , 36, 37, 43, 54
Herbelin , 49, 60, 72
Jean-Yves Girard , 21, 27, 28, 30, 38, 41
Jieh Hsiang , 20
João Marcos , 104
Joachim Lambek , 15
John Alan Robinson , 16, 17
Juan Carlos Agudelo-Agudelo , 102
Lan Zhang , 84
Larry Wos , 19
Laura Giordano , 74
Lellmann , 87, 88
Leo Bachmair , 20, 36, 37, 43
Leroy Chew , 93
Lilia Ramalho Martins , 47
Liron Cohen , 63
Luis Fariñas del Cerro , 23, 56
Luiz Pereira , 50
M. Andrew Moshier , 113
Marcos A. Castilho , 56
Mariusz Urbański , 57
Martin Hyland , 33
Martin Peim , 40
Martin Wirsing , 31, 39
Michael Fisher , 40, 65, 67
Michael Gabbay , 85
Michael Kohlhase , 75
Michael Ludwig , 83
Mikoláš Janota , 93
Minghui Ma , 99
Mykola Nikitchenko , 116
Nachum Dershowitz , 20
Nicola Olivetti , 73, 74, 89
Olaf Beyersdorff , 93
Olivier Gasquet , 56
Olivier Laurent , 58
Ori Lahav , 104
Parigot , 45
Patrick J. Hayes , 18
Pattinson , 87, 88
Paul Bernays , 6
Peter B. Bendix , 20
Peter Baumgartner , 66
Régis Alenda , 73, 89
Rajeev Goré , 95
Ranald Clouston , 95
Reiner Hähnle , 34
Robert Kowalski , 18
Robert Nieuwenhuis , 42
Sabine Frittella , 99, 107, 109
Sergey Maslov , 18
Shôji Maehara , 14
Stepan Skilniak , 116
Stephen Cole Kleene , 12, 13
Szymon Chlebowski , 96, 97
Terlouw , 32
Torben Braüner , 62, 64
Ullrich Hustadt , 40, 65, 67, 83, 84
Uwe Waldmann , 43, 54
Valentina Gliozzi , 74
Valeria de Paiva , 33, 50, 61, 64
Vasilyi Shangin , 69
Vlasta Sikimić , 99, 109
Walter Carnielli , 102
Wilhelm Ackermann , 7
Woody W. Bledsoe , 25
Yoni Zohar , 104
Yves Lafont , 27
Zhaohui Luo , 48, 52
Zhiguang Zhao , 99

Implementations of Proof Systems

Conflict Resolution
Scavenger , 103

Ordered Fine-Grained Resolution with Selection
TSPASS , 83

Resolution for First-Order Temporal Logic
TeMP , 65, 67