Encyclopaedia of Proof Systems

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

Preface

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 63 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 to produce an encyclopedia is only at the beginning and will continue to be open to submissions for a long time.

October 2016 *Bruno Woltzenlogel Paleo*

Contents

Part I *Proof Systems*

Part II *Indexes*

Part I
Proof Systems

Intuitionistic Natural Deduction NJ (1935)

$$\frac{\mathfrak{A} \quad \mathfrak{B}}{\mathfrak{A}\&\mathfrak{B}}\ UE \qquad \frac{\mathfrak{A}\&\mathfrak{B}}{\mathfrak{A}}\ UB \qquad \frac{\mathfrak{A}\&\mathfrak{B}}{\mathfrak{B}}\ UB$$

$$\frac{\mathfrak{A}}{\mathfrak{A}\lor\mathfrak{B}}\ OE \qquad \frac{\mathfrak{B}}{\mathfrak{A}\lor\mathfrak{B}}\ OE \qquad \frac{\mathfrak{A}\lor\mathfrak{B} \quad \begin{matrix}[\mathfrak{A}]\\ \vdots \\ \mathfrak{C}\end{matrix} \quad \begin{matrix}[\mathfrak{B}]\\ \vdots \\ \mathfrak{C}\end{matrix}}{\mathfrak{C}}\ OB$$

$$\frac{\mathfrak{F}\mathfrak{a}}{\forall\mathfrak{x}\mathfrak{F}\mathfrak{x}}\ AE \qquad \frac{\forall\mathfrak{x}\mathfrak{F}\mathfrak{x}}{\mathfrak{F}\mathfrak{a}}\ AB \qquad \frac{\mathfrak{F}\mathfrak{a}}{\exists\mathfrak{x}\mathfrak{F}\mathfrak{x}}\ EE \qquad \frac{\exists\mathfrak{x}\mathfrak{F}\mathfrak{x} \quad \begin{matrix}[\mathfrak{F}\mathfrak{a}]\\ \vdots \\ \mathfrak{C}\end{matrix}}{\mathfrak{C}}\ EB$$

$$\frac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{\mathfrak{A}\supset\mathfrak{B}}\ FE \qquad \frac{\mathfrak{A} \quad \mathfrak{A}\supset\mathfrak{B}}{\mathfrak{B}}\ FB \qquad \frac{\begin{matrix}[\mathfrak{A}]\\ \vdots \\ \text{⅄}\end{matrix}}{\neg\mathfrak{A}}\ NE \qquad \frac{\mathfrak{A} \quad \neg\mathfrak{A}}{\text{⅄}}\ NB \qquad \frac{\text{⅄}}{\mathfrak{D}}$$

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 $\mathfrak{F}\mathring{\mathfrak{a}}$.

Clarifications: The names of the rules are those originally given by Gentzen [**Gentzen1935**]:
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 [**Gentzen1935**], 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 {3}.

Entry 1 by: Bruno Woltzenlogel Paleo

Classical Sequent Calculus LK (1935)

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

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

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

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

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

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

$$\frac{\mathit{\Gamma} \vdash \mathit{\Lambda}, A \quad B, \mathit{\Delta} \vdash \mathit{\Theta}}{A \rightarrow B, \mathit{\Gamma}, \mathit{\Delta} \vdash \mathit{\Lambda}, \mathit{\Theta}}\ \rightarrow_l \qquad\qquad \frac{A, \mathit{\Gamma} \vdash \mathit{\Theta}, B}{\mathit{\Gamma} \vdash \mathit{\Theta}, A \rightarrow B}\ \rightarrow_r$$

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

The eigenvariable α should not occur in $\mathit{\Gamma}$, $\mathit{\Theta}$ 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[**lk:Gentzen1935**], using modern notations and rule names.

Remarks: LK is complete relative to **NK** (i.e. **NJ** {1} with the axiom of excluded middle) and sound relative to a Hilbert-style calculus **LHK** [**lk:Gentzen1935a**]. Cut is eliminable (*Hauptsatz* [**lk:Gentzen1935**]), 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* [**lk:Gentzen1935a**].

Entry 2 by: Martin Riener

Intuitionistic Sequent Calculus LJ (1935)

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

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

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

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

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

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

$$\dfrac{\Gamma \vdash A \quad B, \Delta \vdash \Theta}{A \rightarrow B, \Gamma, \Delta \vdash \Theta}\ \rightarrow_l \qquad \dfrac{A, \Gamma \vdash B}{\Gamma \vdash A \rightarrow B}\ \rightarrow_r$$

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

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

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

Clarifications: **LJ** and **LK** {2} have exactly the same rules, but in **LJ** the succedent of every sequent may have at most one formula. This restriction is equivalent to forbidding the axiom of excluded middle in natural deduction.

Remarks: The cut rule is eliminable (*Hauptsatz* [**Gentzen1935**]), and hence intuitionistic predicate logic is consistent and its propositional fragment is decidable [**Gentzen1935a**]. **LJ** is complete relative to **NJ** {1} and sound relative to the Hilbert-style calculus **LHJ** [**Gentzen1935a**].

Entry 3 by: Giselle Reis

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: The 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 [**Kleene:1952**] and [**Kleene:1967**].

Remarks: Based on Gentzen's sequent calculus **LK** {2} (called classical **G1** in [**Kleene:1952**]). Seems to be the first system (with {5}) 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 {5} to uniformly obtain decidability of propositional classical and intuitionistic logic via backwards proof search in [**Kleene:1952**].

Entry 4 by: Björn Lellmann, Valeria de Paiva

Kleene's Intuitionistic G3 System (1952)

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

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

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

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

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

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

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

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: The 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 [**Kleene:1952**] and [**Kleene:1967**].

Remarks: Based on Gentzen's sequent calculus **LJ** {3} (corresponding to intuitionistic **G1** in [**Kleene:1952**]). Seems to be the first system (with {4}) 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 {4} to uniformly obtain decidability of propositional classical and intuitionistic logic via backwards proof search in [**Kleene:1952**].

Entry 5 by: Björn Lellmann, Valeria de Paiva

Multi-Conclusion Sequent Calculus LJ′ (1954)

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

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

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

$$\frac{\Gamma \vdash \Theta, A \quad B, \Delta \vdash \Lambda}{A \rightarrow B, \Gamma, \Delta \vdash \Theta, \Lambda}\ \rightarrow_l \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \rightarrow B}\ \rightarrow_r$$

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

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

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

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** {3} is defined by restricting **LK** {2} to single conclusion, in **LJ′** only the rules $\neg_r$, $\rightarrow_r$ and $\forall_r$ have this restriction.

History: **LJ′** was proposed in [**Maehara1954**] and used to prove the completeness of **LJ** {3} in [**takeuti87**]. It also appears in [**dragalin88**] (as GHPC) and [**dummett77**] (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 [**Schellinx91**] and Reis [**GisellePhD**].

Entry 6 by: Giselle Reis, Valeria de Paiva

Lambek Calculus (1958)

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

$$\frac{\Gamma_1 \vdash A \quad \Gamma_2 \vdash B}{\Gamma_1, \Gamma_2 \vdash A \otimes B}\, \otimes_r \qquad \frac{\Gamma_1, A, B, \Gamma_2 \vdash C}{\Gamma_1, A \otimes B, \Gamma_2 \vdash C}\, \otimes_l \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightharpoonup B}\, \rightharpoonup_r \qquad \frac{\Gamma_1 \vdash A \quad \Gamma_2, B, \Gamma_3 \vdash C}{\Gamma_1, A \rightharpoonup B, \Gamma_2, \Gamma_3 \vdash C}\, \rightharpoonup_l$$

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

Clarifications: The Lambek Calculus described here was introduced by Joachim Lambek to study sentence structure in 1958 [**lambek1958**]. Actually the calculus Lambek first introduced, despite being motivated by algebraic considerations as we are told in [**lambek1988**], 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) {3} if we remove the structural rules of contraction, weakening and commutation. Lambek also introduced another calculus [**lambek1961**] 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" [**lambek1958**]. 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 [**moortgat2010**]. 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 {20}, 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 [**moortgat2010**].

Entry 7 by: Harley Eades III, Valeria de Paiva

First-Order Unification (1965)

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

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

Clarifications: x is a variable. $v_1,\dots,v_n,u_1,\dots,u_n$ 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.

History: The unification principle was first described by Herbrand in his thesis [**herbrand1930recherches**] but was overlooked until rediscovered independently by Prawitz [**Prawitz1960**] and Gould [**guard1964automated**] (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 [**Robinson1965JACM**] (see {9}). The above set of rules is taken from Snyder and Gallier [**Snyder1989101**].

Remarks: The set S is considered solved if it contains only solved pairs. 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 [**Robinson1965JACM**] of a most general unifier of S. The choice of which equation to process is a ”don't-care” non-determinism, which means that the resulting substitutions, if they exist, are identical up to the renaming of free variables.

Entry 8 by: Tomer Libal

Resolution (1965)

$$\frac{D \lor B_1 \lor \ldots \lor B_m \quad C \lor \neg A_1 \lor \ldots \lor \neg A_n}{(D \lor C)\sigma}\ \mathit{Resolution}$$

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 [**DavisPutnam1960JACM**]. To refute a set of non-ground clauses, the rule was combined with a naïve enumeration of ground instances. Robinson’s fundamental achievement [**Robinson1965JACM**] 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* [**Maslov1964**, **Maslov1971**].

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 {10}, 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 {11}). 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., {11}.

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

Entry 9 by: Uwe Waldmann

Ordered Resolution (1969)

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

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

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 [**Maslov1964**, **Maslov1968**, **Maslov1971**] 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 [**KowalskiHayes1969**] for resolution itself (the requirements for the ordering differ slightly).

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

Entry 10 by: Uwe Waldmann

Paramodulation (1969)

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

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

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

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

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 {9} is impractical due to the huge search space generated in particular by the transitivity axiom. The paramodulation calculus developed by Robinson and Wos [**RobinsonWos1969**] 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 [**Brand1975SIAMJC**]. Many refinements were developed in the sequel, aiming in particular at reducing the number of possible inferences, see {24}.

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

Entry 11 by: Uwe Waldmann

(Unfailing) Completion (1970/1986)

Standard Completion:

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

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

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

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

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

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

plus, for Unfailing Completion:

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

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 [**KnuthBendix1970**]; the presentation as an inference system given here and the extension to unfailing completion are due to Bachmair, Dershowitz, and Hsiang [**BachmairDershowitzHsiang1986LICS**, **Bachmair1991**]. An extension of completion to completion modulo associativity and/or commutativity was presented in [**PetersonStickel1981**].

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 $E \models s \approx t$.

Entry 12 by: Uwe Waldmann

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

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

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

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

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

Clarifications: The presentation from [**AspLongo:91**] 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 [**Gir:71**, **Rey:74**] and included in the λ-cube [**Bar:91**]. It is important for functional programming and inspired works on higher order type systems and many extensions (e.g. $\mathbf{F}_c$ [**LMS:93**], **F** with subtyping [**CMMS:91**, **LMS:00**]).

Remarks: A strong normalization theorem for **F** was proved by Girard [**Gir:72**]. It implies a normalization theorem and consistency for second order arithmetic PA_2. For $\mathbf{F}_c$, a *genericity theorem* holds [**LMS:93**].

Entry 13 by: Giuseppe Longo, Kathleen Milsted , Sergei Soloviev

Higher-Order Pre-Unification (1975)

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

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

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

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. $\overline{o_p}=o_1,\dots,o_p$. $z,x,\overline{x_k}$ 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 $\beta_1\to\dots\to\beta_n\to\gamma$. 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 [**goldfarb81tcs**]. The first complete procedure for higher-order unification was given by Jensen and Pietrzykowski [**jensen76tcs**]. The use of pre-unifiers, introduced by Huet, enabled the search to be less redundant and more efficient.

Remarks: Huet [**huet75tcs**] 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 [**Robinson1965JACM**] of a pre-unifier of S which can always be extended into a unifier. The application of the last two rules is a "don't-know" non-determinism, while the choice of which equation to choose 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.

Entry 14 by: Tomer Libal

Resolution for Modal Logic K (RK) (1982)

Rules for computing resolvents

$$\text{(A1)}\ \frac{}{\Sigma(p,\neg p) \longrightarrow \bot} \qquad \text{(A2)}\ \frac{}{\Sigma(\bot, A) \longrightarrow \bot} \qquad (\Sigma\vee)\ \frac{\Sigma(A,B) \longrightarrow C}{\Sigma(A \vee D_1, B \vee D_2) \longrightarrow C \vee D_1 \vee D_2}$$

$$(\Gamma\vee)\ \frac{\Gamma(A) \longrightarrow B}{\Gamma(A \vee C) \longrightarrow B \vee C} \qquad (\Gamma\Box)\ \frac{\Gamma(A) \longrightarrow B}{\Gamma(\Box A) \longrightarrow \Box B}$$

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

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

Simplification rules

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

Inference rules

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

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 [**farinas.1982**]. The current presentation is inspired by [**enjalbert-farinas.1989**]. The method is at the core of the MOLOG language [**bieber-farinas-herzig.1988**]. With slight variations of the rules, some other modal logics like S4 or S5 can be obtained [**enjalbert-farinas.1989**]. The method has been adaptated to first-order modal logic [**cialdea.1991**]. An alternative non-clausal resolution method is presented in [**abadi-manna.1985**] (for LTL).

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

Entry 15 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 [**miller87sl**, **miller83**] 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 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 {21}. That parallelism is explicitly analyzed in [**chaudhuri14jlc**] using a multi-focused version of LKF {52}.

Entry 16 by: Dale Miller

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

SPLIT: basic rules of Natural Deduction(see {1}), 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 {1}. 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** [**bledsoe:1971**] called a resolution program if necessary. Then, in [**bledsoe:1972**], 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 [**bledsoe:1977**].

Entry 17 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 {17}).
Flatten : Replace $P(f(x))$ by $\exists y(y : f(x) \land P(y))$ 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 $\exists y(y \in E \land x \in y)$
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 {17}.
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 {17}.

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 [**pastre:1989**, **pastre:1993**] 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.

Entry 18 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** {3}. 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) [**mints1977closed**] is obtained.

History: Introduced by Girard and Lafont in [**lafont1987tapsoft**] as intuitionistic variant of **LL** {20}. **ILL** has multiple applications in categorical logic.

Remarks: Enjoys cut elimination [**lafont1987tapsoft**].

Entry 19 by: Joseph Boudou, Sergei Soloviev

Linear Sequent Calculus LL (1987)

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

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

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

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

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

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

Clarifications: If $\Gamma = A_1, \dots, A_n$ then $?\Gamma = ?A_1, \dots, ?A_n$. 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** [**ll**] come from the analysis of intuitionistic logic through Girard's decomposition of the intuitionistic implication into the linear implication: $A \rightarrow B = !A \multimap B$.

Remarks: Cut elimination holds. **LL** is sound and complete with respect to phase semantics [**ll**]. **LL** is not decidable. Sequent calculi **LK** {2} and **LJ** {3} can be translated into **LL**.

Entry 20 by: Olivier Laurent

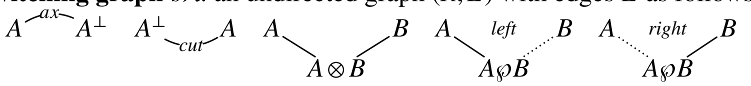
Proof Nets for MLL$^-$ (1987)

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

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* [**DanosRegnier**]).

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 $A \overset{axiom}{\underset{cut}{\overline{\qquad}}} A^\perp$ is a proof structure.
2. Proof nets are canonical representations of **MLL**$^-$ sequent calculus proofs and solve the proof identity problem for **MLL**$^-$ in linear time. The *desequentialization map* $(\,)^-$ identifies sequent calculus derivations d_1 and d_2 that differ only by permutations of inferences:

$$(\ \vdash \overline{A, A^\perp}\ )^- = \dfrac{axiom}{A \quad A^\perp} \qquad \left(\dfrac{\overset{d_1}{\vdash \Gamma, A} \qquad\qquad \overset{d_2}{\vdash \Delta, B}}{\vdash \Gamma, A \otimes B, \Delta}\right)^- = \dfrac{\overset{(d_1)^-\ (d_2)^-}{A \quad B}}{\Gamma\ A \otimes B\ \Delta} \qquad \left(\dfrac{\overset{d}{\vdash \Gamma, A, B}}{\vdash \Gamma, A \wp B}\right)^- = \dfrac{\overset{(d)^-}{A \quad B}}{\Gamma\ A \wp B}$$

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 $C_0 \wp C_1$, respectively. If $C_i \in eA_j$ but $C_0 \wp C_1 \notin eA_j$ then $A_0 \otimes A_1 \in k(C_0 \wp C_1)$.

$$\begin{array}{ccc} & e\mathbf{A_1} & \\ \begin{array}{cc} \vdots & \vdots \end{array} & & \begin{array}{cc} \vdots & \vdots \end{array} \\ \ell_1\ \dfrac{A_0 \qquad \mathbf{A_1}}{A_0 \otimes A_1} & k(C_0 \wp C_1) & \ell_2\ \dfrac{\mathbf{C_0} \quad C_1}{C_0 \wp C_1} \end{array}$$

Entry 21 by: Gianluigi Bellin

Proof. Let $C_0 \in eA_1$; clearly $C_0 \in k(C_0 \wp C_1)$ so $\mathcal{S} = eA_1 \cup k(C_0 \wp C_1) \neq \emptyset$ and by Lemma 1 is a subnet. Suppose $A_0 \otimes A_1$ does not belong to $k(C_0 \wp C_1)$; 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 **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 $A_0 \otimes B_0, \dots, A_n \otimes B_n$. 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, $C \wp D$. By Lemma 2 $A_i \otimes B_i \in k(C \wp D)$. But $C \wp D$ must occur above a link ℓ_j with conclusion $A_j \otimes B_j$. It follows that $(A_i \otimes B_i) \in k(C \wp D) \subset k(A_j \otimes B_j)$ contrary to the choice of ℓ_i. **qed**

History: Proof nets for **MLL**$^-$ were introduced by J-Y.Girard in 1987 [**Girard**]. Simplifications were given by Danos and Regnier [**DanosRegnier**] and in [**BellinDeWiele**]. 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.

Pure Type Systems (1989)

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

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

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

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

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:

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

where $\mathcal{V}$ is a set of variables and $\mathcal{C}$ is a set of constants. A PTS is parameterized by a *specification* $(\mathcal{S},\mathcal{A},\mathcal{R})$ where $\mathcal{S}\subseteq\mathcal{C}$ is the set of *sorts*, $\mathcal{A}\subseteq\mathcal{C}\times\mathcal{S}$ 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 [**barendregt˙introduction˙1991**, **geuvers˙modular˙1991**, **barendregt˙lambda˙1992**, **geuvers˙logics˙1993**]. Many important systems can be expressed as PTSs, including simply typed λ calculus ($\lambda\!\rightarrow$), $\lambda\Pi$ calculus {31} (λP), system F {13} ($\lambda 2$), and the calculus of constructions (λC):

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

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

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 $\lambda *$:

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

Entry 22 by: Ali Assaf

Full Intuitionistic Linear Logic (FILL) (1990)

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

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

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

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

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

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

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

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:

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

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 [**dePaiva:1990**] did not satisfy cut-elimination, as shown by Schellinx. Martin Hyland and Valeria de Paiva [**Hyland:1993**] 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 [**Hyland:1993**], which was corrected independently, with different proof methods, by Bierman [**Bierman:1996**], Bellin [**Bellin:1997**], Brauner/dePaiva [**Brauner:1998**], dePaiva/Ritter [**dePaiva:2006**]. The version here is the minimal modification suggested by Bellin, (who used proofnets), but using a traditional term assignment, as described in Eades/dePaiva [**Eades:2015**].

Entry 23 by: Harley Eades III, Valeria de Paiva

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 $P(t_1, \ldots, t_n) \approx \textit{true}$.

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 {25}.

Entry 24 by: Uwe Waldmann

History: The superposition calculus [**BachmairGanzinger1990CTRS**, **BachmairGanzinger1994JLC**] by Bachmair and Ganzinger refines the paramodulation calculus {11}. 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 {10} and unfailing completion {12}. In order to preserve refutational completeness, one new inference rule must be added – either the *Merging Paramodulation* rule [**BachmairGanzinger1990CTRS**] 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 {28}, theory superposition {36}, or hierarchic superposition {29}.

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.

Saturation With Redundancy (1990)

Primary Rules

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

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

Derived Rules

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

is a shorthand for

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

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 [**BachmairGanzinger1990CTRS**] for superposition {24}; it can be used for most other superposition-like calculi, such as constraint superposition {28}, superposition modulo theories {36}, or hierarchic superposition {29}, with appropriate choices for $\models$, $\mathcal{I}$, and $\mathcal{R}$.

Entry 25 by: Uwe Waldmann

Constructive Classical Logic LC (1991)

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

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

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

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

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

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

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

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: $M, N ::= \neg X \mid \neg V \mid \neg F \mid M \vee N \mid M \vee Q \mid P \vee N \mid M \wedge N$

$\mathit{\Gamma}$ and $\mathit{\Delta}$ are lists of formulas, and $\mathit{\Pi}$ 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 $\vdash \mathit{\Gamma} \,;\, \mathit{\Pi}$ where $\mathit{\Pi}$ is called the stoup.

History: **LC** [**lc**] comes from the analysis of classical logic inside the coherent semantics of linear logic [**ll**] together with the use of the focusing property [**focal**].

Remarks: Cut elimination holds. **LK** {2} 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 [**lc**] (a variant of coherence spaces [**ll**]).

Entry 26 by: Olivier Laurent

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 {20}.
History: This formalization first appeared in [**Troelstra:1992**].

Entry 27 by: Elaine Pimentel, Harley Eades III

Constraint Superposition (1992/1995)

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

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

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 $>$), 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 $C \; \llbracket T \rrbracket$ 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, $C \; \llbracket \textit{true} \rrbracket$. 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. {25}). In particular, every constrained clause with an unsatisfiable constraint is redundant.

History: The idea to use constrained formulas in automated reasoning originated in [**KirchnerKirchnerRusinowitch1990RFIA**]. There are several reasons to switch from standard superposition {24} to superposition with constrained clauses [**NieuwenhuisRubio1992ESOP**, **NieuwenhuisRubio1992CADE**, **NieuwenhuisRubio1995JSC**]. 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 [**NieuwenhuisRubio1994CADE**], 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).

Entry 28 by: Uwe Waldmann

Hierarchic Superposition (1992/2013)

Abstraction

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

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

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

and the rules of the standard superposition calculus {24}, 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 {25}.

History: The hierarchic superposition calculus [**BachmairGanzingerWaldmann1992ALP**, **BachmairGanzingerWaldmann1994A** 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 [**BaumgartnerWaldmann2013CADE**]; this calculus uses a weaker form of abstraction that is guaranteed to preserve sufficient completeness but requires an additional abstraction step after each inference.

Entry 29 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

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

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

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

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

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

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

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

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 [**Parigot92**], 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 [**FelFriKohDub86**]) that were known at this time to

Entry 30 by: Hugo Herbelin

provide computational content to classical logic [**Griffin90**], as well as a decomposition of Prawitz's classical elimination rule of negation [**Prawitz65**].

The original presentation [**Parigot92**] 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. [**HerbelinHdR**]). Variants with multiplicative disjunctions can be found in [**Selinger01**] or [**PymRitter01**], or multiplicative conjunctions in [**HerbelinHdR**].

A standard variant originating in [**deGroote94**] 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 [**HerbelinSaurin10**].

Typed LF for Type Theories (1994)

$$\frac{}{<>\vdash \textbf{valid}} \qquad \frac{\Gamma \vdash K\,\textbf{kind} \quad x \notin FV(\Gamma)}{\Gamma, x : K \vdash \textbf{valid}} \qquad \frac{\Gamma, x : K, \Gamma' \vdash \textbf{valid}}{\Gamma, x : K, \Gamma' \vdash x : K} \ (1) \qquad \frac{\Gamma \vdash \textbf{valid}}{\Gamma \vdash \textbf{Type}\,\textbf{kind}} \qquad \frac{\Gamma \vdash A : \textbf{Type}}{\Gamma \vdash El(A)\,\textbf{kind}} \ (5)$$

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

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

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

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

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

Clarifications: We follow [**Luo:94**]. Terms of **LF** are of the forms **Type**, $El(A)$, $(x : K)K'$ (dependent product), $[x : K]K'$ (abstraction), $f(k)$, and judgements of the forms $\Gamma \vdash$ **valid** (validity of context), $\Gamma \vdash K$**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 [**Luo:94**], ch. 9, **LF** is a typed version of Martin-Löf's logical framework [**NPS:90**]. 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 [**Luo:99**, **SolLuo:02**, **LuoSolXue:13**].

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 [**Luo:94**]. H. Goguen defined a typed operational semantics for UTT and proved strong normalization theorem [**HG:94**]. For **LF** with coercive subtyping conservativity results were obtained [**Luo:99**, **SolLuo:02**, **LuoSolXue:13**].

Entry 31 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

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

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

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:

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

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 $\overline{\lambda}$-calculus has been designed in [**Herbelin94**, **HerbelinPhD**]. 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 [**girard91mscs**] which himself credits it to Andreoli [**andreoli92jlc**] (see also {52}). With proof annotations removed, the calculus can be seen as the intuitionistic fragment LJT of the subsystem LKT of LK [**danos93wll**], 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 {30} 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 [**HerbelinPhD**]). A variant with implicit substitution is possible.

The symmetrization of $\overline{\lambda}$-calculus led to $\mathbf{LK}_{\mu\tilde{\mu}}$ {40}.

Entry 32 by: Hugo Herbelin

Full Intuitionistic Logic (FIL) (1995)

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

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

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

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

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

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

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

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

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 $\Rightarrow\rightarrow$. See [**dePaiva:2005**] for more details.

History: The system FIL was announced in the abstract [**dePaiva:1995**] but only published officially ten years later in [**dePaiva:2005**]. The system was conceived after the remark in the paper describing FILL {23} 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.

Entry 33 by: Harley Eades III, Valeria de Paiva

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

Basic subkinding rule

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

Subkinding for dependent product kinds

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

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

$$\frac{\mathit{\Gamma} \vdash K_1' <_{c_1} K_1 \quad \mathit{\Gamma}, x : K_1' \vdash [c_1x/x]K_2 <_{c_2} K_2' \quad \mathit{\Gamma}, x : K_1 \vdash K_2 : \mathbf{kind}}{\mathit{\Gamma} \vdash (x : K_1)K_2 <_{[f:(x:K_1)K_2][x:K_1']c_2(f(c_1x))} (x : K_1')K_2'}$$

Coercive application rules

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

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

Coercive definition rule

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

Structural rules

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

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

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

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

Clarifications: We follow [**LuoSolXue:13**]. In this entry the extensions $T[C]$ of the logical framework **LF**{31} 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 $\mathit{\Gamma} \vdash A <_c B : Type$. The set C itself may be generated by some user-defined rules. As coercive definition rule above shows, coercive subtyping is considered as

Entry 34 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

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

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 $\mathit{\Gamma} \vdash A <_c B : Type$ then $\mathit{\Gamma} \vdash A : Type$, $\mathit{\Gamma} \vdash B : Type$ and $\mathit{\Gamma} \vdash c : (El(A))El(B)$.
- $\mathit{\Gamma} \nvdash A <_c A : Type$ for any $\mathit{\Gamma}, A, c$.
- If $\mathit{\Gamma} \vdash A <_c B : Type$ and $\mathit{\Gamma} \vdash A <_{c'} B : Type$ then $\mathit{\Gamma} \vdash c = c' : (El(A))El(B)$.

History: Coercive subtiping as an abbreviation mechanism was introduced in a conference paper [**Luo96:CSL**]. It was described for type theories specified in Z. Luo's typed **LF** (extensions of **LF**) {31}, but the idea itself is much more general and may apply to other type theories. The approach was further developed in [**jls:TYPES96**, **Luo:99**, **SolLuo:02**, **LuoSolXue:13**].

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.

Sequent Calculus G3c (1996)

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

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

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

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

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

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

P should be atomic in Ax and y should not be free in the conclusion of R∀ and L∃

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.

Entry 35 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 {24} 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 {25}.

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 [**GanzingerWaldmann1996CADE**] 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 [**Plotkin1972**, **Slagle1974JACM**, **RusinowitchVigneron1995**, **BachmairGanzinger1994CTRS**], superposition modulo abelian groups [**GodoyNieuwenhuis2004**], chaining calculi [**Slagle1972JACM**, **Hines1992JAR**, **BachmairGanzinger1994LICS**, **BachmairGanzinger1994CADE**], or superposition modulo divisible torsion-free abelian groups or ordered divisible abelian groups [**Waldmann2002abJSC**, **Waldmann2001IJCAR**].

Entry 36 by: Uwe Waldmann

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

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

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

$$\frac{\begin{tikzcd}[row sep=tiny] & \bullet \Delta_1, \Box A \\ \Gamma \bullet \arrow[ur] \arrow[dr] & \\ & \bullet \Delta_2 \end{tikzcd}}{\begin{tikzcd}[row sep=tiny] & \bullet \Delta_1, \Box A \\ \Gamma \bullet \arrow[ur] \arrow[dr] & \\ & \bullet \Delta_2, \Box A \end{tikzcd}}\,(5_{\rightarrow}) \qquad \frac{\Gamma \bullet \longrightarrow \bullet \Delta, \Box A}{\Gamma, \Box A \bullet \longrightarrow \bullet \Delta, \Box A}\,(5_{\uparrow}) \qquad \frac{\Gamma \bullet \longrightarrow \overset{\Delta_1, \Box A}{\bullet} \longrightarrow \bullet \Delta_2}{\Gamma \bullet \longrightarrow \overset{\Delta_1, \Box A}{\bullet} \longrightarrow \bullet \Delta_2, \Box A}\,(5_{\downarrow})$$

STRUCTURAL RULES

$$\frac{\Gamma \bullet}{\Gamma \bullet \longrightarrow \bullet \emptyset}\,(D) \qquad \frac{\Gamma \bullet \longrightarrow \bullet \Delta}{\begin{tikzcd}[row sep=tiny] \Gamma \bullet \arrow[rr] \arrow[dr] & & \bullet \Delta \\ & \underset{\emptyset}{\bullet} \arrow[ur] & \end{tikzcd}}\,(De)$$

$$\frac{\begin{tikzcd}[row sep=tiny] & \bullet \Delta_1 \\ \Gamma \bullet \arrow[ur] \arrow[dr] & \\ & \bullet \Delta_2 \end{tikzcd}}{\begin{tikzcd}[row sep=tiny] & \overset{\Delta_1}{\bullet} \arrow[dr] & \\ \Gamma \bullet \arrow[ur] \arrow[dr] & & \bullet \emptyset \\ & \underset{\Delta_2}{\bullet} \arrow[ur] & \end{tikzcd}}\,(C_0) \qquad \frac{\Gamma \bullet \longrightarrow \bullet \Delta}{\Gamma \bullet \longrightarrow \underset{\Delta}{\bullet} \longrightarrow \bullet \emptyset}\,(C_1)$$

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 37.1 below.

Entry 37 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 37.1 Correspondences between modal axioms and graph-based tableaux rules.

History: Tableaux methods for modal logics have a long history started by Kripke [**kripke.1959**]. The present method, introduced in [**castilho-farinas-gasquet-herzig.1997**] and extended in [**farinas-gasquet.2002**], 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 [**gasquet-herzig-said-schwarzentruber.2014**].

Remarks: The method is sound and complete for any combination of axioms. Termination is more problematic and has been investigated in [**farinas-gasquet.2002**, **gasquet-herzig-sahade.2006**, **gasquet-herzig-said-schwarzentruber.2014**].

Synthetic Tableaux (2000)

The synthesizing rules are:

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

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

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

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

The branching rule:

$$\overset{\frown}{p_i \ \neg p_i}$$

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 [**Urbanski2001a**], [**Urbanski2001b**], [**Urbanski2002a**]. In [**Urbanski2002a**], [**Urbanski2002b**] and [**Urbanski2004**] 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 [**Urbanski2002a**], [**Urbanski2002b**], [**Urbanski2004**].

Entry 38 by: Dorota Leszczyńska-Jasion

Polarized Linear Sequent Calculus LLP (2000)

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

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

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

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

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

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** {20}. Their dual, the positive connectives ($\otimes$, 1, $\oplus$, 0) have the focusing property [**focal**], related here with the "at most one positive formula" property of sequents.

History: **LLP** [**phdlaurent**] comes from the natural embedding of Girard's **LC** {26} into linear logic {20}. 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.

Entry 39 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

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

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

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

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

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

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

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 {2} 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 [**CurienHerbelin00**]. Thanks to the

Entry 40 by: Hugo Herbelin

presence of two dual axiom rules and implicit contraction rules, the system supports a tree-like sequent-free presentation like originally presented by Gentzen for natural deduction [**HerbelinHdR**] (see {47}). The system can be seen as a symmetric variant of $\overline{\lambda}$-calculus {32}.

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 [**Wadler03**, **HerbelinHdR**]. A variant with only commands, called $\mathcal{X}$, has been studied in [**BakLenLes05**], based on previous work in [**UrbanPhD**]. 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 $\Gamma \vdash \Delta$.

Constructive Modal Logic S4 (CS4) (2000)

$$\frac{}{\varDelta, A \vdash A}\;ax \qquad \frac{\varGamma \vdash A \quad A, \varDelta \vdash B}{\varGamma, \varDelta \vdash B}\;cut \qquad \frac{}{\varGamma, \bot \vdash A}\;\bot\mathcal{L}$$

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

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

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

$$\frac{\Box\varGamma \vdash \Box A}{\Box\varGamma, \varDelta \vdash A}\;\Box\mathcal{R} \qquad \frac{\Box\varGamma, A \vdash \Diamond B}{\varDelta, \Box\varGamma, \Diamond A \vdash \Diamond B}\;\Diamond\mathcal{L} \qquad \frac{\varGamma \vdash A}{\varGamma \vdash \Diamond A}\;\Diamond\mathcal{R}$$

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

History: The intuitionistic system for S4 that we are calling constructive S4 (CS4) here, was originally described by Prawitz in his Natural Deduction book[**prawitznatural**] in 1965. This system differs from what is more widely called now IS4, originally defined by Fisher-Servi [**Fisher-Servi:1981**] and thoroughly studied in Simpson's PhD thesis [**simpson1994phd**] 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 [**bierman2000**].

Entry 41 by: Harley Eades III, Valeria de Paiva

Model Evolution (2003)

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

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

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 [**Baumgartner:Tinelli:ModelEvolutionCalculus:CADE:2003**]. Additional optional inference rules, not shown here, help improve performance in practice.

History: Model Evolution [**Baumgartner:Tinelli:ModelEvolutionCalculus:CADE:2003**] improves the earlier FDPLL calculus [**Baumgartner:FDPLL:CADE:00**], lifting the core of the propositional DPLL method to the first-order level. It has been extended by ordered paramodulation rules for equality reasoning [**Baumgartner:Tinelli:ModelEvolutionCalculusEquality:CADE:2005**], by lemma learning techniques inspired by modern CDCL SAT solvers [**Baumgartner:etal:ModelEvolutionLearning:LPAR:2006**] and by reasoning modulo background theories [**Baumgartner:Fuchs:Tinelli:MELIA:LPAR:2008**, **Baumgartner:Tinelli:MEET:C**
It has been combined with the superposition calculus in [**Baumgartner:Waldmann:MESUP:CADE:2009**].

Entry 42 by: Peter Baumgartner

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	$\neg A$
$\neg(A \to B)$	A	$\neg B$	$A \to B$	$\neg A$	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 [**AW:2004**]. 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. [**AW:2013**]).

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.

Entry 43 by: Dorota Leszczyńska-Jasion

Socratic Proofs for FOL (2004)

The rules of calculus $\mathbf{E}^*$ (see {43}) and the quantifier rules:

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

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

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.

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

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 **??**. 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 [**AW:2006SPQ**], 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 [**AW:2006SPQ**] and later examined in [**LJUW:2013**].

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.

Entry 44 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 {43}, {44}). 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 [**DLJ:2004**], the completeness proof may be found in [**DLJ:2007**].

Remarks: A sequent $\vdash (A)^1$ has a Socratic proof in $\mathbf{E}^{\mathsf{K}}$ iff A is K-valid.

Entry 45 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 {45}), 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 $i_1, \dots, i_n$ 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 $i_1, \dots, i_n$ 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 $i_1, \dots, i_n$ and $j_1, \dots, j_m$ 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 {45}, {43}, {44} for more comments.

History: The proof system has been presented in [**DLJ:2004**], the completeness proof may be found in [**DLJ:2007**], and extensions to some non-basic modal logics in [**DLJ:2008**].

Remarks: A sequent $\vdash (A)^1$ has a Socratic proof in $\mathbf{E}^{\mathsf{L}}$ iff A is L-valid.

Entry 46 by: Dorota Leszczyńska-Jasion

LK$\mu\tilde{\mu}$ in sequent-free tree form (2005)

Structural subsystem

$$\frac{\vdash A \quad A \vdash}{\vdash}\ \mathit{Cut}$$

$$\begin{array}{c}[\vdash A]\\ \vdots\\ \dfrac{\vdash}{A \vdash}\end{array}\ \mathit{Focus}_L \qquad \begin{array}{c}[A \vdash]\\ \vdots\\ \dfrac{\vdash}{\vdash A}\end{array}\ \mathit{Focus}_R$$

Introduction rules

$$\frac{A_i \vdash}{A_1 \wedge A_2 \vdash}\ \wedge^i_L \qquad \frac{\vdash A_1 \quad \vdash A_2}{\vdash A_1 \wedge A_2}\ \wedge_R$$

$$\frac{A_1 \vdash \quad A_2 \vdash}{A_1 \vee A_2 \vdash}\ \vee_L \qquad \frac{\vdash A_i}{\vdash A_1 \vee A_2}\ \vee^i_R$$

$$\frac{\vdash A \quad B \vdash}{A \to B \vdash}\ \to_L \qquad \begin{array}{c}[\vdash A]\\ \vdots\\ \dfrac{\vdash B}{\vdash A \to B}\end{array}\ \to_R$$

$$\frac{A[y] \vdash}{\exists x A[x] \vdash}\ \exists_L \qquad \frac{\vdash A[t]}{\vdash \exists x A[x]}\ \exists_R$$

$$\frac{A[t] \vdash}{\forall x A[x] \vdash}\ \forall_L \qquad \frac{\vdash A[y]}{\vdash \forall x A[x]}\ \forall_R$$

$$\frac{}{\bot \vdash}\ \bot_L \qquad \frac{}{\vdash \top}\ \top_R$$

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 [**Gentzen1935**] 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 [**Howard80**] 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 [**HerbelinHdR**].

The terminology "sequent calculus" seems to have become popular from [**Prawitz65**] followed then e.g. by [**Troelstra73**] 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 [**Munch-Maccagnoni09**].

Remarks: As pointed out e.g. in [**GeuversPhD**] 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 {40}.

Entry 47 by: Hugo Herbelin

Conditional Labelled Sequent Calculi SeqS (2003-2007)

$$(\mathbf{AX})\ \Gamma, x : P \vdash \Delta, x : P \quad (P \text{ atomic}) \qquad\qquad (\mathbf{A}\bot)\ \Gamma, x : \bot \vdash \Delta$$

$$\frac{\Gamma, x : A \Rightarrow B \vdash x \xrightarrow{A} y, \Delta \qquad \Gamma, x : A \Rightarrow B, y : B \vdash \Delta}{\Gamma, x : A \Rightarrow B \vdash \Delta}\ (\Rightarrow \mathbf{L}) \qquad\qquad \frac{\Gamma, x \xrightarrow{A} y \vdash \Delta, y : B}{\Gamma \vdash \Delta, x : A \Rightarrow B}\ (\Rightarrow \mathbf{R}) \quad (y \notin \Gamma, \Delta)$$

$$\frac{u : A \vdash u : B \qquad u : B \vdash u : A}{\Gamma, x \xrightarrow{A} y \vdash x \xrightarrow{B} y, \Delta}\ (\mathbf{EQ}) \qquad \frac{\Gamma, x \xrightarrow{A} y, y : A \vdash \Delta}{\Gamma, x \xrightarrow{A} y \vdash \Delta}\ (\mathbf{ID}) \qquad \frac{\Gamma \vdash x \xrightarrow{A} x, x : A, \Delta}{\Gamma \vdash x \xrightarrow{A} x, \Delta}\ (\mathbf{MP})$$

$$\frac{\Gamma, x \xrightarrow{A} y \vdash \Delta, x : A \qquad \Gamma[x/u, y/u], u \xrightarrow{A} u \vdash \Delta[x/u, y/u]}{\Gamma, x \xrightarrow{A} y \vdash \Delta}\ (\mathbf{CS})\ (x \neq y, u \notin \Gamma, \Delta)$$

$$\frac{\Gamma x \xrightarrow{A} y \vdash \Delta, x \xrightarrow{A} z \qquad (\Gamma x \xrightarrow{A} y \vdash \Delta)[y/u, z/u]}{\Gamma x \xrightarrow{A} y \vdash \Delta}\ (\mathbf{CEM})\ (y \neq z, u \notin \Gamma, \Delta)$$

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 (**EQ**) takes care of normality. Besides the rules shown, SeqS includes standard rules for propositional connectives.

History: The calculi SeqS have been introduced in [**toclpozz**]. The theorem prover CondLean, implementing SeqS calculi in Prolog, has been presented in [**tab2003pozz**, **tab2005pozz**].

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.

Entry 48 by: Nicola Olivetti, Gian Luca Pozzato , Camilla Schwind

Preferential Tableau Calculi $\mathcal{T}P^{\mathbf{T}}$ (2005-2009)

$$\Gamma, P, \neg P \ (\mathbf{AX}) \quad \text{with } P \text{ atomic} \qquad\qquad \frac{\Gamma, \neg\Box\neg A; \Sigma}{A, \Box\neg A, \Gamma^{\Box}, \Gamma^{\Box^{\downarrow}}, \Gamma^{\mid\sim^{\pm}}, \Sigma; \emptyset} \ (\Box^{-})$$

$$\frac{\Gamma, \neg(A \mid\sim B); \Sigma}{A, \Box\neg A, \neg B, \Gamma^{\mid\sim^{\pm}}; \emptyset} \ (\mid\sim^{-}) \qquad \frac{\Gamma, A \mid\sim B; \Sigma}{\Gamma, \neg A; \Sigma, A \mid\sim B \qquad \Gamma, \neg\Box\neg A; \Sigma, A \mid\sim B \qquad \Gamma, B; \Sigma, A \mid\sim B} \ (\mid\sim^{+})$$

Clarifications: According to Kraus, Lehmann and Magidor (KLM) [**KrausLehmannMagidor:90**], defeasible knowledge is represented by a (finite) set of nonmonotonic conditionals $A \mid\sim B$ (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 $A \mid\sim B$ 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 $A \mid\sim B$ 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 $A \mid\sim B$. Σ is used to keep track of positive conditionals $A \mid\sim B$ to which the rule $(\mid\sim^{+})$ has already been applied: the idea is that one does not need to apply $(\mid\sim^{+})$ on the same conditional formula $A \mid\sim B$ *more than once in the same world*. When $(\mid\sim^{+})$ is applied to a formula $A \mid\sim B \in \Gamma$, then $A \mid\sim B$ 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 $(\mid\sim^{+})$ in new worlds.

Given Γ, we define:

- $\Gamma^{\Box} = \{\Box\neg A \mid \Box\neg A \in \Gamma\}$
- $\Gamma^{\Box^{\downarrow}} = \{\neg A \mid \Box\neg A \in \Gamma\}$
- $\Gamma^{\mid\sim^{+}} = \{A \mid\sim B \mid A \mid\sim B \in \Gamma\}$
- $\Gamma^{\mid\sim^{-}} = \{\neg(A \mid\sim B) \mid \neg(A \mid\sim B) \in \Gamma\}$
- $\Gamma^{\mid\sim^{\pm}} = \Gamma^{\mid\sim^{+}} \cup \Gamma^{\mid\sim^{-}}$

Besides the rules shown above, the calculus $\mathcal{T}P^{\mathbf{T}}$ also includes standard rules for propositional connectives.

History: In [**KrausLehmannMagidor:90**] 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 $A \mid\sim B$, whose reading is *normally (or typically) the A's are B's*. The operator "$\mid\sim$" is nonmonotonic, in the sense that $A \mid\sim B$ does not imply $A \wedge C \mid\sim B$. The calculus $\mathcal{T}P^{\mathbf{T}}$ and extensions for all the logics of the KLM family are proposed in [**toclKLMpozz**]. The theorem provers KLMLean and FreeP implementing the tableau calculi have been presented at [**tableaux2007pozz**, **aiia207pozz**].

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.

Entry 49 by: Laura Giordano, Valentina Gliozzi , Nicola Olivetti , Gian Luca Pozzato

HO Sequent Calculi $\mathcal{G}_\beta$ and $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ (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 $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ 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) [**sep-type-theory-church**]. 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 [**J18**], and earlier (two-sided) related versions in [**R19**] and [**BrownPhD**].

Remarks: $\mathcal{G}_\beta$ is sound and complete for elementary type theory ($\mathcal{G}_\beta$ is thus also sound for HOL). $\mathcal{G}_{\beta\mathfrak{f}\mathfrak{b}}$ is sound and complete for HOL. Moreover, both calculi are cut-free and they do not admit cut-simulation [**J18**].

Entry 50 by: Christoph Benzmüller

Extensional HO RUE-Resolution (1999-2013)

Normalisation Rules

$$\frac{\mathbf{C} \vee [\mathbf{A} \vee \mathbf{B}]^{\mathrm{tt}}}{\mathbf{C} \vee [\mathbf{A}]^{\mathrm{tt}} \vee [\mathbf{B}]^{\mathrm{tt}}}\, \vee^{\mathrm{tt}} \qquad \frac{\mathbf{C} \vee [\mathbf{A} \vee \mathbf{B}]^{\mathrm{ff}}}{\begin{array}{c}\mathbf{C} \vee [\mathbf{A}]^{\mathrm{ff}} \\ \mathbf{C} \vee [\mathbf{B}]^{\mathrm{ff}}\end{array}}\, \vee^{\mathrm{ff}} \qquad \frac{\mathbf{C} \vee [\neg \mathbf{A}]^{\mathrm{tt}}}{\mathbf{C} \vee [\mathbf{A}]^{\mathrm{ff}}}\, \neg^{\mathrm{tt}} \qquad \frac{\mathbf{C} \vee [\neg \mathbf{A}]^{\mathrm{ff}}}{\mathbf{C} \vee [\mathbf{A}]^{\mathrm{tt}}}\, \neg^{\mathrm{ff}}$$

$$\frac{\mathbf{C} \vee [\Pi^{\tau} \mathbf{A}]^{\mathrm{tt}} \quad X^{\tau} \text{ fresh variable}}{\mathbf{C} \vee [\mathbf{A}\, X]^{\mathrm{tt}}}\, \Pi^{\mathrm{tt}} \qquad \frac{\mathbf{C} \vee [\Pi^{\tau} \mathbf{A}]^{\mathrm{ff}} \quad \mathtt{sk}^{\tau} \text{ Skolem term}}{\mathbf{C} \vee [\mathbf{A}\, \mathtt{sk}^{\tau}]^{\mathrm{ff}}}\, \Pi^{\mathrm{ff}}$$

Resolution, Factorisation and Primitive Substitution

$$\frac{[\mathbf{A}]^{p_1} \vee \mathbf{C} \quad [\mathbf{B}]^{p_2} \vee \mathbf{D} \quad p_1 \neq p_2}{\mathbf{C} \vee \mathbf{D} \vee [\mathbf{A} = \mathbf{B}]^{\mathrm{ff}}}\, \mathsf{res} \qquad \frac{\mathbf{C} \vee [\mathbf{A}]^{p} \vee [\mathbf{B}]^{p}}{\mathbf{C} \vee [\mathbf{A}]^{p} \vee [\mathbf{A} = \mathbf{B}]^{\mathrm{ff}}}\, \mathsf{fac}$$

$$\frac{[Q_{\tau} \overline{\mathbf{A}}^{n}]^{p} \vee \mathbf{C} \quad \mathbf{P} \in \mathcal{AB}_{\tau}^{(k)} \text{ for logic connective } k}{([Q_{\tau} \overline{\mathbf{A}}^{n}]^{p} \vee \mathbf{C})[\mathbf{P}/Q]}\, \mathsf{prim\_subst}$$

Extensionality and Pre-unification

$$\frac{\mathbf{C} \vee [\mathbf{A}^{\sigma\tau} = \mathbf{B}^{\sigma\tau}]^{\mathrm{tt}} \quad X^{\tau} \text{ fresh variable}}{\mathbf{C} \vee [\mathbf{A}\, X = \mathbf{B}\, X]^{\mathrm{tt}}}\, \textsc{FuncPos} \qquad \frac{\mathbf{C} \vee [\mathbf{A}^{o} = \mathbf{B}^{o}]^{\mathrm{tt}}}{\mathbf{C} \vee [\mathbf{A}^{o} \longleftrightarrow \mathbf{B}^{o}]^{\mathrm{tt}}}\, \textsc{BoolPos}$$

$$\frac{\mathbf{C} \vee [\mathbf{A}^{\sigma\tau} = \mathbf{B}^{\sigma\tau}]^{\mathrm{ff}} \quad \mathtt{sk}^{\tau} \text{ Skol. term}}{\mathbf{C} \vee [\mathbf{A}\, \mathtt{sk} = \mathbf{B}\, \mathtt{sk}]^{\mathrm{ff}}}\, \textsc{FuncNeg} \qquad \frac{\mathbf{C} \vee [\mathbf{A}^{o} = \mathbf{B}^{o}]^{\mathrm{ff}}}{\mathbf{C} \vee [\mathbf{A}^{o} \longleftrightarrow \mathbf{B}^{o}]^{\mathrm{ff}}}\, \textsc{BoolNeg}$$

$$\frac{\mathbf{C} \vee [h^{\sigma\tau} \overline{\mathbf{A}^{\alpha}}^{k} \ = \ h^{\sigma\tau} \overline{\mathbf{B}^{\alpha}}^{k}]^{\mathrm{ff}}}{\mathbf{C} \vee \overline{[\mathbf{A}_i \ = \ \mathbf{B}_i]^{\mathrm{ff}}}^{i \leq k}}\, \textsc{Dec} \qquad \frac{\mathbf{C} \vee [X = \mathbf{A}]^{\mathrm{ff}} \quad X \notin \mathrm{FV}(\mathbf{A})}{\mathbf{C}[\mathbf{A}/X]}\, \textsc{Subst}$$

$$\frac{\mathbf{C} \vee [\mathbf{A} = \mathbf{A}]^{\mathrm{ff}}}{\mathbf{C}}\, \textsc{Triv} \qquad \frac{\mathbf{C} \vee [F^{\tau} \overline{\mathbf{A}}^{n} \ = \ h \overline{\mathbf{B}}^{m}]^{\mathrm{ff}} \quad \mathbf{G} \in \mathcal{AB}_{\tau}^{(h)}}{\mathbf{C} \vee [F = \mathbf{G}]^{\mathrm{ff}} \ \vee \ [F \overline{\mathbf{A}}^{n} = h \overline{\mathbf{B}}^{m}]^{\mathrm{ff}}}\, \textsc{FlexRigid}$$

Choice

$$\frac{C := \mathbf{C}' \vee [\mathbf{A}[E_{(\alpha \to o) \to \alpha} \mathbf{B}]]^{p} \qquad \begin{array}{c}\epsilon \in \mathsf{CFs},\ E = \epsilon \textit{ or } E \in \textit{freeVars(C)}, \\ \textit{freeVars}(\mathbf{B}) \subseteq \textit{freeVars}(C), Y \textit{fresh}\end{array}}{[\mathbf{B}\ Y]^{\mathrm{ff}} \vee [\mathbf{B}\ (\epsilon_{\alpha(\alpha o)} \mathbf{B})]^{\mathrm{tt}}}\, \mathsf{choice}$$

$$\frac{[PX]^{\mathrm{ff}} \vee [P(f_{(\alpha \to o) \to \alpha} P)]^{\mathrm{tt}}}{\mathsf{CFs} \longleftarrow \mathsf{CFs} \cup \{f_{(\alpha \to o) \to \alpha}\}}\, \mathsf{detectChoiceFn}$$

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 [**sep-type-theory-church**]; see also {50}). 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

Entry 51 by: Christoph Benzmüller

of literals. 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 $\mathtt{tt}$ or $\mathtt{ff}$), e.g. $[\neg X]^{\mathtt{ff}}$ denotes the negation of $\neg X$. $\mathrm{FV}(\mathbf{A})$ denotes the free variables of term $\mathbf{A}$. $\mathcal{AB}_{\tau}^{(h)}$ 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 [**C5**] and [**J5**]. Recent modifications and extensions (e.g. choice) are discussed in [**W47**] and [**EasyChair:215**]. The calculus is inspired by and extends Huet's constrained resolution [**Huet:amott73**, **Huet:cracmfhol72**] and the extensional resolution calculus in [**C2**].

Remarks: The calculus works for classical higher-order logic with Henkin semantics and choice. Soundness and completeness has been discussed in [**C5**] and [**J5**]. 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.

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 [**liang09tcs**, **liang07csl**]. A multifocus variant of **LKF** has been described in [**chaudhuri14jlc**]. The design of **LKF** borrows strongly from Andreoli's focused proof system for linear logic [**andreoli92jlc**] and Girard's LC proof system [**girard91mscs**]. The first-order versions of the LKT and LKQ proof systems of [**danos93wll**] can be seen subsystems of **LKF**.

Entry 52 by: Dale Miller

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 {52}.) 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 53 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 [**liang09tcs**, **liang07csl**]. **LJF** can be seen as a generalization to the MJ sequent system of Howe [**howe98phd**]. Other focused proof systems, such as LJT [**herbelin95phd**], LJQ/LJQ' [**dyckhoff06cie**], and λRCC [**jagadeesan05fsttcs**] can be directly emulated within **LJF** by making the appropriate choice of polarization.

$\lambda\Pi$-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 [**DBLP:conf/tlca/CousineauD07**] as an expressive logical framework. It has been used to design *shallow encodings* of many logics and calculus such as functional Pure Type Systems [**DBLP:conf/tlca/CousineauD07**], Higher-Order Logic [**AliHOL**], the Calculus of Inductive Constructions [**Coqine**], resolution and superposition [**Resolution**], or the ς-calculus [**RaphaelSigma**]. The well-formedness rules for global contexts were not part of the original type system and have been introduced by Saillard [**ModuloBeta**]. The $\lambda\Pi$-Calculus Modulo is implemented in the proof checker Dedukti [**Dedukti**].

Entry 54 by: Ronan Saillard

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 [**ModuloBeta**]. Decidability of type inference depends on strong normalization.
