Canonical Logical Frameworks

Robin Adams

Department of Computer Science
Royal Holloway, University of London

26 November 2009

Introduction

- There are several different logical frameworks, which use inconsistent notation and terminology.
- Canonical logical frameworks have been invented independently 3.5 times, each time with different notation and terminology.
- But with the appropriate dictionary, . . .
- . . . the frameworks all fit together into a *modular hierarchy*.

History of Logical Frameworks

- AUTOMATH (1968–)
  PAL, AUT-68, AUT-QE, ...
  "The AUTOMATH café"
- $$\begin{array}{ccc} \text{Martin-Löf's theory of types} & \longrightarrow & \text{Martin-Löf's logical framework} \\ \text{objects and types} & & \text{objects and kinds} \\ \hline \text{Martin-Löf's theory of sets} & \longrightarrow & \text{Martin-Löf type theory} \\ \text{objects and sets} & & \text{objects and types} \end{array}$$
  - LF (Luo 1994)
    objects and kinds
  - PAL$^+$ (Luo 2000)
    objects and kinds
- Edinburgh Logical Framework, ELF$\longrightarrow$ LF (1993)
  objects, types $\subseteq$ families and kinds
  - Linear Logical Framework, LLF (1996)
  - Concurrent Logical Framework, CLF (2003)
  - Canonical Logical Framework, Canonical LF (2007)

Logical Frameworks

A logical framework F is a type theory
that we use to represent the expressions of another formal language, the object theory — which is often itself a type theory.
The logical framework has *objects* and *kinds*.
The object theory has *terms* and *types*.

There is a kind **Type**.
For any A : **Type**, there is a kind $\mathrm{El}\,(A)$.

We declare *constants* and *computation rules* such that

- the objects in **Type** behave like the types in T;
- the objects in $\mathrm{El}\,(A)$ behave like the terms of type A in T.

The Idea of a Canonical Logical Framework

Typically:

- Each entity of the object theory is represented by a $\beta\eta$-*equivalence class* of objects.
- There are objects that do not represent entities (partial application).
- Framework-level reduction and object theory reduction often interact.

Example

The object ΣA does not correspond to any object theory entity.
Instead, it represents the *meta-function* that maps B to $\Sigma x : A.B$.
These features are desirable in practice (e.g. abbreviation mechanisms) but awkward for theoretical work. **Idea**:

- construct a logical framework L which deals with only β-*short*, η-*long forms.*
- Each object theory is represented by a *unique* object.
- Partial application is disallowed.
- There is no framework-level reduction.

The Type Framework, TF

The *Type Framework* TF, developed by Aczel (2001). Syntax and theory elaborated by Adams (2003–4).
Each variable and constant z has an *order*, $o(z)$.
An object M has the form

$$z[[x_{11}, \ldots, x_{1r_1}]M_1, \ldots, [x_{n1}, \ldots, x_{nr_n}]M_n]$$

where $o(x_{ij}) \leq o(z) - 2$.
An abstraction is an expression

$$[x_1, \ldots, x_r]M$$

Its *order* is $\max_i o(x_i) + 1$.
In place of substitution, use instantiation (*hereditary substitution*):
$\{F/x\}M$ is the normal form of $[F/x]M$.
Definition

$$\begin{aligned} \{F/x\}z[\vec{G}] &\equiv z[\{F/x\}\vec{G}] \qquad (x \not\equiv z) \\ \{[\vec{y}]M/x\}x[\vec{G}] &\equiv \{\{[\vec{y}]M/x\}\vec{G}/\vec{y}\}M \end{aligned}$$

History of Canonical Logical Frameworks

- 1996 — Linear Logical Framework (Cervesato)
- 2000 — PAL$^+$ (Luo)
  Does not allow partial application
  Does have abstractions, framework-level reduction
- 2001 — Type Framework, TF (Aczel, Adams)
- 2003 — Concurrent Logical Framework (Watkins, Cervesato, Pfenning, Walker)
- 2004 — Modular Hierarchy of Logical Frameworks (Adams)
- 2006 — DMBEL (Plotkin)
- 2007 — Canonical LF (Harper, Licata)

The Modular Hierarchy

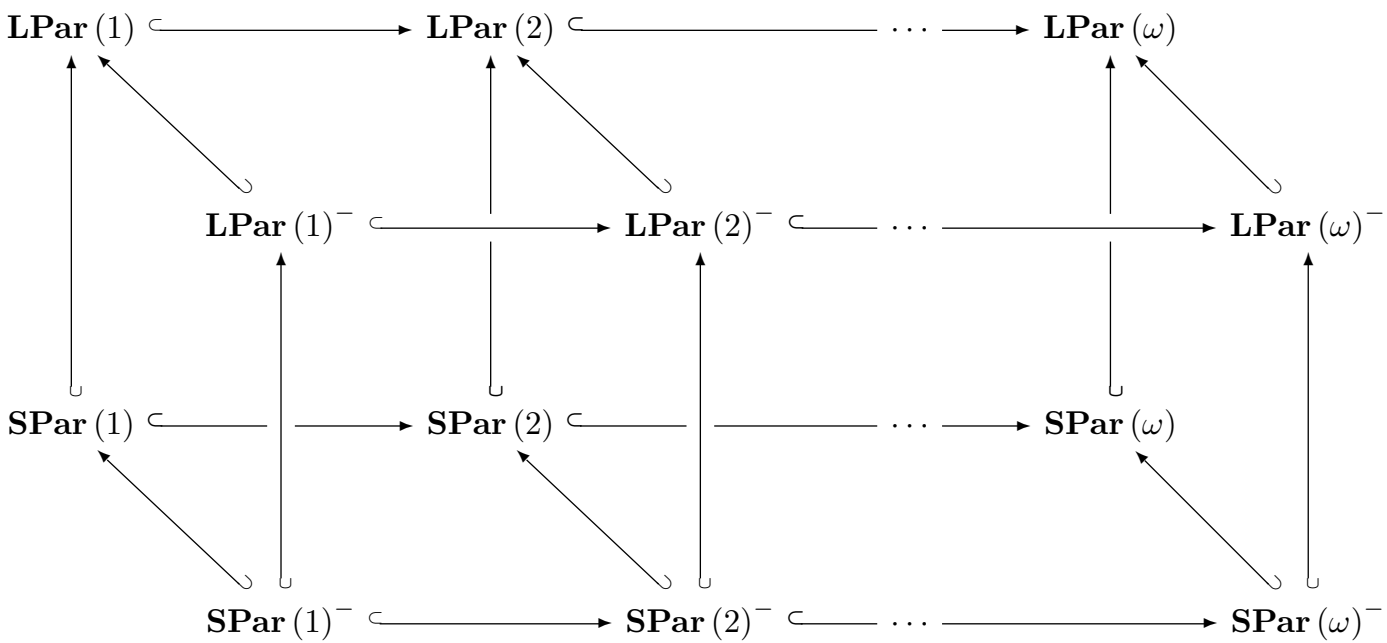
A set of subsystems of TF that extend one another conservatively.

	No equation declarations	Equation declarations
Parameters of small type only	$\mathbf{SPar}\,(n)^-$	$\mathbf{SPar}\,(n)$
Parameters of small type and large kind	$\mathbf{LPar}\,(n)^-$	$\mathbf{LPar}\,(n)$

If we declare $c : (x_1 : (\Delta_1)\,T_1) \cdots (x_m : (\Delta_m)\,T_m)\,T$, then:

- the *parameters* of c are $x_1, \ldots, x_n$;
- x_i is of *small type* if $T_i \equiv \mathrm{El}\,(A_i)$;
- x_i is of *large kind* if $T_i \equiv \mathbf{Type}$.

The number n is the largest order of constant that may appear.


LPar (1)
LPar (2)
LPar (ω)
LPar (1)⁻
LPar (2)⁻
LPar (ω)⁻
SPar (1)
SPar (2)
SPar (ω)
SPar (1)⁻
SPar (2)⁻
SPar (ω)⁻

Embedding Canonical Frameworks in Traditional Frameworks

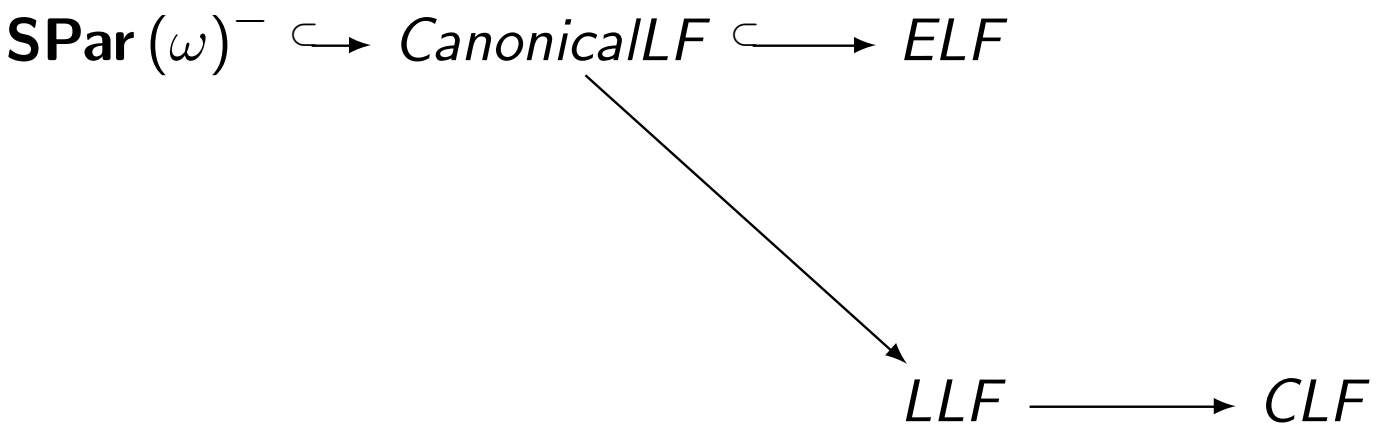
TF is a conservative subsystem of LF.
The deraivable judgements of TF are exactly
the derivable judgements of LF that are in β-normal, η-long form.

Relating ELF and $\mathbf{SPar}\,(\omega)^{-}$

$\mathbf{SPar}\,(\omega)^{-}$ is a conservative subsystem of Canonical LF, which is a conservative subsystem of ELF.

ELF	$\mathbf{SPar}\,(\omega)^{-}$
Kind	Kind of form $(\Delta)\mathbf{Type}$
Type	Kind of form $\mathrm{El}\,(M)$
	or object of kind **Type**
Family	Abstraction of form $[\Delta](\Theta)M$
	where M : **Type**
Object	Object of kind $(\Delta)\mathrm{El}\,(M)$

Canonical LF	$\mathbf{SPar}\,(\omega)^{-}$
Kind	Kind of form $(\Delta)\mathbf{Type}$
Canonical Type Family	Kind of form $(\Delta)\mathrm{El}\,(M)$
Atomic Type Family	Object of kind **Type**
Canonical Term	Abstraction of kind $(\Delta)\mathrm{El}\,(M)$
Atomic Term	Object of kind $\mathrm{El}\,(M)$

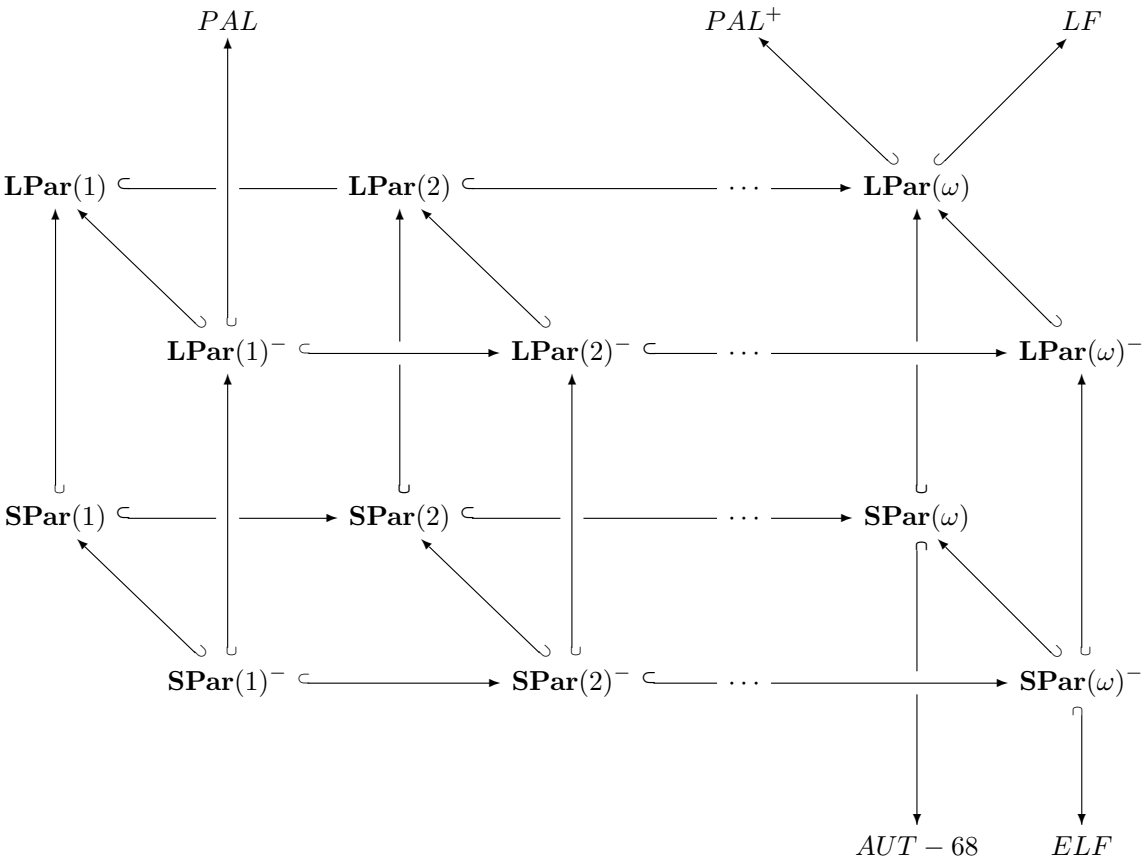

SPar $(\omega)^-$
CanonicalLF
ELF
LLF
CLF

Relating DMBEL and **SPar** (3)

SPar $(2)^-$ is a subsystem of DMBEL, which is a subsystem of **SPar** $(3)^-$

DMBEL	**SPar** $(2/3)$
Type	Object of kind **Type**
Abstraction Type	Kind of form $(\Delta)\mathrm{El}\,(M)$
Term	Object of kind $\mathrm{El}\,(M)$
Abstraction Term	Abstraction of kind $(\Delta)\mathrm{El}\,(M)$

$$\textbf{SPar}\,(2)^- \longrightarrow \mathit{DMBEL} \longrightarrow \textbf{SPar}\,(3)^-$$


PAL
PAL^+
LF
$\mathbf{LPar}(1)$
$\mathbf{LPar}(2)$
$\cdots$
$\mathbf{LPar}(\omega)$
$\mathbf{LPar}(1)^-$
$\mathbf{LPar}(2)^-$
$\cdots$
$\mathbf{LPar}(\omega)^-$
$\mathbf{SPar}(1)$
$\mathbf{SPar}(2)$
$\cdots$
$\mathbf{SPar}(\omega)$
$\mathbf{SPar}(1)^-$
$\mathbf{SPar}(2)^-$
$\cdots$
$\mathbf{SPar}(\omega)^-$
$AUT-68$
ELF

Summary

These are the questions that determine a framework's expressive power:

	Maximum order of constants?	**Type** occurs negatively?	Computation rules?
LF	ω	Yes	Yes
ELF	ω	No	No
PAL$^+$	ω	Yes	Yes
TF	ω	Yes	Yes
Canonical LF	ω	No	No
DMBEL	3	No	No

Conclusion

- Canonical frameworks bring some order to the world of logical frameworks.
- A framework is a canonical framework plus some (theoretically redundant) features.
- The questions that determine a framework's expressive power are:
  - What order may the variables and constants be?
  - To what depth may **Type** occur negatively?
  - May the user declare computation rules?
- Different answers to these questions determine different canonical frameworks . . .
- . . . which extend one another conservatively in a modular hierarchy