Technical Report No. 725

Department of Computer Science

A note on representation and semantics in logical frameworks

David Pym

Queen Mary and Westfield College

UNIVERSITY OF LONDON

1996

To appear: Proc. CADE-13 Workshop on Proof-search in Type-theoretic Languages, D. Galmiche (ed.), 1996.

A note on representation and semantics in logical frameworks*

D.J. Pym

Queen Mary and Westfield College
University of London

1 Introduction

We begin with a brief remark on the nature of this document. We emphasize that it is not intended to be a complete discussion of all the issues it raises and that we elide many details. Also, it does not contain a complete collection of references. It is intended to provide a sketch of our point of view which, we believe, constitutes a research contribution.

Type-theoretic languages are often described as 'logical frameworks'. We contend that such claims are severely deficient. Specifically, we claim that in order to describe a framework, one must:

1. Characterize the class of (object-)logics to be represented;
2. Give a metalanguage, preferably including an account of its (meta-)logical status with respect to the class of object-logics;
3. Characterize the mechanism by which object-logics are represented.

This point of view can be conveniently summarized by the slogan,

$$Framework = Language + Representation.$$

We claim that each of (1), (2) and (3), which are interdependent, has an impact on the following uses of frameworks:

4. As formal theories of logics;
5. As bases for formal theories of computation in specified logics — in our setting, search-based computation (logic programming).

In § 2, we consider the *judgements-as-types* notion of representation [6], illustrating its significance by considering frameworks whose languages are (i) the $\lambda\Pi$-calculus, *i.e.*, LF [6], and (ii) the $\lambda\Lambda$-calculus, *i.e.*, RLF [8]. In § 3, we consider briefly the consequences of our point of view for the semantics of logical frameworks. In § 4, we review briefly the notion of search-induced computation for the framework (= language + representation), considering how the requirements of a semantics of computation can be satisfied within our model theory.

In § 5, we mention some recent related ideas of the present author, his collaborators and others.

*This research was supported in part by UK EPSRC grant GR/K41687, 'Search Modules I: Representation and Combination of Proof Procedures'.

2 Representation

2.1 The LF logical framework

The *judgements-as-types* notion of representation, described informally for LF in [6], begins with Kant's formulation of logic [10], as developed by Martin-Löf [11]. We contend that it is important to formulate this idea in two steps — identifiable *formally* for LF in [6] only for the particular cases of (classical) first- and higher-order natural deduction — as follows:

1. Consider object-logics as systems for deriving not propositions but rather *judged propositions*;
2. Consider a correspondence between judged propositions and types in the language of the framework constructed over a signature containing type-constructors corresponding to each judgement form of the object-logic.

With this formulation, LF's representation of object-logics now goes as follows:

An object-consequence, in logic L, is written

$$\delta : (X, J_1(\phi_1), \ldots, J_m(\phi_m) \vdash_L J(\phi)),$$

where the J_i and J are judgements, X is the set of variables that occur in the formulae and δ is a proof-object. This object-consequence corresponds, in the language of the framework, to a meta-consequence

$$\Gamma_X, y_1 : J_1(\phi_1), \ldots, y_m : J_m(\phi_m) \vdash_{\Sigma_L} M_\delta : J(\phi),$$

where Γ_X corresponds to the set X of variables, each y_i corresponds to a place-holder for the interpretation of a proof of each $J_i(\phi_i)$, and M_δ is a $\lambda\Pi$-term corresponding to the proof-object δ. Roughly speaking, the propositions-as-types correspondence for the $\{\supset, \forall\}$-fragment of minimal first-order logic is the special case in which each $J_i (= J) = \mathsf{proof}$.

Roughly speaking, LF is concerned with those Hilbert and natural deduction systems for which the correspondence is *uniform* [7].[1] The basic idea is that an encoding Σ_L of a logic L is uniform if there is a surjection from consequences

$$\delta : (X, J_1(\phi_1), \ldots, J_m(\phi_m) \vdash_L J(\phi)),$$

in L, to consequences

$$\Gamma_X, y_1 : J_1(\phi_1), \ldots, y_m : J_m(\phi_m) \vdash_{\Sigma_L} M_\delta : J(\phi),$$

in Σ_L.

One property of this form of representation is that the *encoded* version of an object-logic inherits the structural properties, such as weakening and/or contraction, of the language of the framework. For example, suppose that Σ_L is a uniform encoding of L, and that $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_L} M_\delta : J(\phi)$ is the image of the

[1] This notion appears to require some adaptation for our formulation.

object-consequence $\delta : (X, \Delta \vdash_T J(\phi))$. In $\lambda\Pi$, weakening is admissible, so that if $\Gamma_X, \Gamma_\Delta \vdash_{\Sigma_L} M_\delta : J(\phi)$ is provable, so is $\Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_L} M_\delta : J(\phi)$ (provided $\Gamma_X, \Gamma_\Delta, \Gamma_\Theta$ is well-formed). By uniformity of Σ_L, $\Gamma_X, \Gamma_\Delta, \Gamma_\Theta \vdash_{\Sigma_L} M_\delta : J(\phi)$ is then the image of an object-consequence $\delta' : (X, \Delta, \Theta \vdash_T J(\phi))$.

2.2 The RLF logical framework

It should now be clear that linear and other relevance or substructural logics cannot be uniformly encoded in LF.

In recent work with S. Ishtiaq [8], we have developed a dependent type theory, the $\lambda\Lambda$-calculus, *with full linear dependent function types.* It is in propositions-as-types correspondence with an intuitionistic linear logic with a linear universal quantifier. This type theory, and some *relevant* variants, can be used as languages for frameworks. Using the judgements-as-types notion of representation, they are able to support uniform encodings of linear and other relevance logics.[2]

We will not develop here the theories of representation, semantics and logic programming for frameworks with linear dependent types. Indeed, these are substantial research topics. However, consideration of the following two illustrative inference rules will indicate some of the problems that must be solved:

$$\Lambda\mathrm{E}\ \frac{\Gamma \vdash_\Sigma M : \Lambda x{:}A\,.\,B \quad \Delta \vdash_\Sigma N{:}A}{\Xi \vdash_\Sigma MN : B[N/x]}\,[\Xi;\Gamma;\Delta] \qquad \Pi\mathrm{E}\ \frac{\Gamma \vdash_\Sigma M : \Lambda x!A\,.\,B \quad !\Delta \vdash_\Sigma N{:}A}{\Xi \vdash_\Sigma MN : B[N/x]}\,[\Xi;\Gamma;!\Delta],$$

where $[\Xi; \Gamma; \Delta]$ should be read as 'Ξ splits into Γ and Δ'.

3 Semantics

In consideration of a semantics for a logical framework — here we restrict our attention to LF, RLF being beyond our present scope — we can see from the discussion above that there are three *requirements* that it must satisfy:

(i) It must provide a semantics for the *type theory*;
(ii) It must provide an account of the *propositions-as-types correspondence* between the type theory and its internal logic, and the *judgements-as-types correspondence* for 'uniform' encodings;
(iii) It must provide a semantics for the notion of *logic programming* induced by search in the language of the framework.

For the first requirement (i) we must have a semantics for dependent types with dependent function spaces that, for example, properly extends the ideas of Mitchell and Moggi [13]. We must also have (Kripke/Beth/Joyal) models of the $\{\supset, \forall\}$-fragment of minimal first-order logic. For the second requirement (ii), we must consider the correspondence between Kripke models of an object-logic and Kripke models of the encoding of that logic in the meta-logic. For the third

[2] This work builds on earlier joint work with D. Miller and G. Plotkin, first described by the present author in [18].

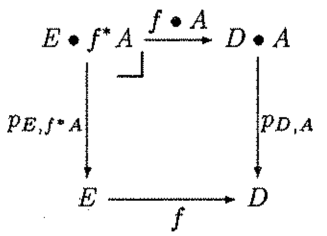
requirement (iii), we must at least be able to identify a class of Herbrand models and be able to provide a least fixed point construction corresponding, as usual, to resolution. We return to the third requirement in more detail in § 4.

We can satisfy requirements (i), (ii) and (iii) within the setting of indexed, or fibred, category theory [14, 9]. These ideas owe much to work of Cartmell [3], Pitts [16], Seely [26], Ehrhard [4], Streicher [27] and others. We take a Kripke *prestructure* to be a functor

$$\mathcal{J} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{V}]],$$

where $\mathcal{W}$ is a small category (of 'worlds'), $\mathcal{D}^{op} = \coprod_W \mathcal{D}_W^{op}$, where W ranges over the objects of $\mathcal{W}$, and each $\mathcal{D}_W$ (the base at W) is small; $\mathcal{V}$, a subcategory of $\mathcal{C}$, is a category of values, such that:

1. Each $\mathcal{D}_W$ has a terminal object, $1_{\mathcal{D}_W}$;
2. Each $\mathcal{J}(W)(D)$ has a terminal object, $1_{\mathcal{J}(W)(D)}$, preserved on the nose by each $f^*(= \mathcal{J}(W)(f))$, where $E \xrightarrow{f} D \in \mathcal{D}_W$;
3. For each $W \in \mathcal{W}$, $D \in \mathcal{D}_W$ and $A \in \mathcal{J}(W)(D)$, there is a $D \bullet A \in \mathcal{D}_W$ together with canonical projections $D \bullet A \xrightarrow{p_{D,A}} D$, $1_{\mathcal{J}(W)(D \bullet A)} \xrightarrow{q_{D,A}} p^*_{D,A}(A)$ and canonical pullbacks



   satisfying the *strictness conditions* that $1_D{}^*(A) = A$ and $1_D \bullet A = 1_{D \bullet A}$, for each A in $\mathcal{J}(W)(D)$, and that $g^*(f^*A) = (g;f)^*A$ and $(g \bullet (f^*A));(f \bullet A) = (g;f) \bullet A$, for each appropriate A, f and g. Moreover, for each W and D, $D \bullet 1_{\mathcal{J}(W)(D)} = D$;
4. At each W, the arrow $p^*_{D,A}(= \mathcal{J}(W)(p_{D,A}))$ has a right adjoint,

$$p^*_{D,A} \dashv \Pi_{D,A} : \mathcal{J}(W)(D \bullet A) \longrightarrow \mathcal{J}(W)(D)$$

   satisfying the following (strict) *Beck-Chevalley* condition: for each $E \xrightarrow{f} D$ in $\mathcal{D}_W$, each A in $\mathcal{J}(W)(D)$ and each B in $\mathcal{J}(W)(D \bullet A)$,

$$f^*(\Pi_{D,A}B) = \Pi_{E,f^*A}((f \bullet A)^*B) \quad \text{and} \quad (f \bullet A)^*(app(A,B)) = app(f^*A, (f \bullet A)^*B),$$

   where *app* is the co-unit of the adjunction.

To get a Kripke structure $\mathcal{K}_\mathcal{J}$ for Σ we must move from the category of values $\mathcal{V}$ to a (weak) arrow construction $\boldsymbol{\mathcal{V}}$ on $\mathcal{V}$ (see below), so that $\mathcal{K}_\mathcal{J} : [\mathcal{W}, [\mathcal{D}^{op}, \boldsymbol{\mathcal{V}}]]$. A Kripke *model* of Σ is then given as a pair $\langle \mathcal{K}_\mathcal{J}, [\![-]\!]_{\mathcal{K}_\mathcal{J}}^{-} \rangle$, where $[\![-]\!]_{\mathcal{K}_\mathcal{J}}^{-}$ is a partial function that interprets the syntax of $\lambda\Pi$ in $\mathcal{K}_{\mathcal{J}}$ (so that we must require that $\mathcal{K}_{\mathcal{J}}$ has 'enough points' to interpret the constants in Σ).

In fact, *prestructures* would be an adequate basis for defining models that would satisfy requirement (i). We define the following notion of satisfaction of the inhabitation of a type A (intended to be of kind Type, *i.e.*, not of the form $\lambda x{:}A\,.\,B$) by an object M with respect to context Γ at world W: let Σ be a signature, $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{V}]]$ be a Kripke Σ-$\lambda\Pi$-model and let Γ be a context, A be a type and M be an object.[3] In the model $\mathcal{K}_{\mathcal{J}}$, the world W *satisfies* the inhabitation of A by M with respect to Γ *i.e.*,

$$W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M{:}A)[\Gamma],$$

if and only if we have that $[\![\Gamma]\!]^{W}_{\mathcal{K}_{\mathcal{J}}}$, $[\![A_{\Gamma}]\!]^{W}_{\mathcal{K}_{\mathcal{J}}}$ and $[\![M_{\Gamma}]\!]^{W}_{\mathcal{K}_{\mathcal{J}}}$ are defined and that $1_{\mathcal{K}_{\mathcal{J}}(W)([\![\Gamma]\!]^{W}_{\mathcal{K}_{\mathcal{J}}})} \xrightarrow{[\![M_{\Gamma}]\!]^{W}_{\mathcal{K}_{\mathcal{J}}}} [\![A_{\Gamma}]\!]^{W}_{\mathcal{K}_{\mathcal{J}}}$ in $\mathcal{K}_{\mathcal{J}}(W)([\![\Gamma]\!]^{W}_{\mathcal{K}_{\mathcal{J}}})$.

We are then able to obtain, *inter alia*, the following familiar-looking properties (all of which are subject to requirements that interpretations $[\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}}$ be suitably defined at appropriate worlds):

Monotonicity: Let Σ be a signature and let $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{V}]]$, be a Kripke Σ-$\lambda\Pi$-model. If $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M{:}A)[\Gamma]$ and if $W \xrightarrow{\alpha} W'$, then $W' \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M{:}A)^{[\alpha]}[\Gamma]$;[4]

Π-forcing: Let Σ be a signature and let $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{V}]]$, be a Kripke Σ-$\lambda\Pi$-model. $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M : \Pi x{:}A\,.\,B)[\Gamma]$ if and only if, for all $W \xrightarrow{\alpha} W'$ and for all N such that $W' \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (N{:}A)^{[\alpha]}[\Gamma]$, there is a P such that $W' \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (P : B[N/x])^{[\alpha]}[\Gamma]$ and $P =_{\beta\eta} MN$. Similarly for the non-dependent function space, $\rightarrow$;

Soundness: Let Σ be a signature and let $\langle \mathcal{K}_{\mathcal{J}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}}} \rangle$, where $\mathcal{K}_{\mathcal{J}} : [\mathcal{W}, [\mathcal{D}^{op}, \mathcal{V}]]$, be a Kripke Σ-$\lambda\Pi$-model. If $\Gamma \vdash_{\Sigma} M{:}A$ is provable, then $W \models_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (M{:}A)[\Gamma]$, at each W for which all the required interpretations are defined;

Model existence: There is a Kripke-Σ-$\lambda\Pi$-model $\langle \mathcal{K}_{\mathcal{J}\mathcal{T}}, [\![-]\!]^{-}_{\mathcal{K}_{\mathcal{J}\mathcal{T}}} \rangle$ with a world W_0 such that if $\Gamma \nvdash_{\Sigma} M{:}A$, then $W_0 \not\models_{\Sigma}^{\mathcal{K}_{\mathcal{J}\mathcal{T}}} (M{:}A)[\Gamma]$;

Completeness: If $\Gamma \models_{\Sigma} M{:}A$ holds, then $\Gamma \vdash_{\Sigma} M{:}A$ is provable.

However, for requirements (ii) and (iii), *structures* are exploited. The basic idea is as follows: a structure is obtained from a prestructure by replacing the category of 'types' over a world and 'context' by a *chosen* category of arrows from the the base. Corresponding to structures, we can define the following satisfaction predicate, which is a generalization of the one above:

$$W \models\!\Rightarrow_{\Sigma}^{\mathcal{K}_{\mathcal{J}}} (\Delta \xrightarrow{\langle M_1,\ldots,M_n\rangle} \Theta)[\Gamma],$$

[3] The types, objects and contexts considered here are required only to be members of the raw syntactic categories.

[4] The superscript $-^{[\alpha]}$ should be read as 'after α'.

i.e., in the Kripke Σ-$\lambda\Pi$-model $\mathcal{K}_{\mathcal{J}}$, W forces $\Delta \stackrel{\langle M_1,\ldots,M_n\rangle}{\longrightarrow} \Theta$ with respect to Γ.

The following can then be obtained:

- A semantic account of propositions-as-types;
- A semantic account of judgements-as-types for uniform LF encodings;
- A least fixed point semantics of logic programming with the $\lambda\Pi$-calculus, *i.e.*, requirement (iii).

We defer discussion of the semantics of the propositions- and judgements-as-types correspondences to another occasion. We discuss how our models satisfy requirement (iii) in § 4.

The details of the ideas described in this section can be found in [21].[5]

4 Proof-search and logic programming

Our formulation of representation provides a convenient setting for considering a variety of issues related to proof-search and logic programming for encoded logics. The details are beyond the scope of this note and have been discussed elsewhere [17, 23, 19, 20, 2]. Here, we concentrate on the semantics of logic programming for the language of the LF framework, $\lambda\Pi$.

In [17, 22], it was shown that the language $\lambda\Pi$ admits a natural interpretation as a logic programming language, based on a calculus **U** of sequents of the form $\Gamma \Rightarrow_\Sigma A(\alpha)$, where α is an indeterminate. Such sequents are interpreted as requests to calculate terms M and N such that $\Gamma \vdash_\Sigma M : A[N/\alpha]$ is provable. Here N corresponds to the usual notion of *answer substitution*, intended to be calculated by unification.[6]

From the semantic point of view, it is useful to formulate a *resolution calculus* for $\lambda\Pi$-realizations, $\Gamma \stackrel{\sigma}{\longrightarrow} \Delta$. The following resolution calculus relies, for its completeness property with respect to the usual calculi for $\lambda\Pi$, on a certain *clausal form* for types [17, 20]:

$$\text{Axiom} \quad \frac{}{\vdash_\Sigma \Gamma \stackrel{\langle @_1,\ldots,@_n\rangle}{\longrightarrow} \Theta} \quad \textit{each } @_i \in \Sigma \cup \Gamma; \tag{1}$$

$$\text{Resolution} \quad \frac{\vdash_\Sigma \Gamma \stackrel{\langle M_1,\ldots,M_i,\ldots,M_n\rangle}{\longrightarrow} \Theta}{\vdash_\Sigma \Gamma \stackrel{\langle M_1,\ldots,M_i',\ldots,M_n\rangle}{\longrightarrow} \Theta} \quad \Gamma \vdash_\Sigma M_i' : D_i[M_j/y_j]_{j=1}^{i-1}, \tag{2}$$

where the clause $@_i : \Pi z_{i1}{:}B_{i1} \ldots \Pi z_{ip}{:}B_{im} . (C_{i1} \to (C_{i2} \to (\ldots(C_{iq} \to D_i)\ldots))) \in \Sigma \cup \Gamma$, $@_i P_1 \ldots P_p Q_1 \ldots Q_q \to_{\beta\eta} M_i'$, for some $1 \leq i \leq n$ and p, q possibly 0;

$$\text{Introduction} \quad \frac{\vdash_\Sigma \Gamma, x{:}A \stackrel{\langle M_1,\ldots,M_n,x,M\rangle}{\longrightarrow} \Gamma, x{:}A, y{:}B}{\vdash_\Sigma \Gamma \stackrel{\langle M_1,\ldots,M_n,\lambda x:A\,.\,M\rangle}{\longrightarrow} \Gamma, y : \Pi x{:}A\,.\,B} \tag{3}$$

and also a rule for $\beta\eta$-equalities.

The key step is to identify a semantic counterpart to the resolution rule (2). The basic idea goes as follows:

[5] We believe our our class of models to be the least prescriptive yet described.

[6] A alternative formulation of logic programming for $\lambda\Pi$, the language of LF, has been presented by Pfenning [15].

- Define a class of *Herbrand* models by defining Herbrand prestructures and structures together with a suitable standard Herbrand interpretation. The prestructure for Herbrand models is identical to that required for model existence in requirement (i) of § (3), the salient feature being the construction of the category of worlds as the full subcategory of the base category of contexts in which each arrow is of the form $\Gamma \xrightarrow{\sigma} \Gamma, \Gamma'$. Herbrand structures, however, are a more delicate matter. An Herbrand structure $\mathcal{H}$ at world Δ and base Γ is a subset of the homset $\mathcal{C}_\Sigma(\Gamma, \Delta)$ of realizations between Γ and Δ.[7] It follows that such Herbrand structures form a complete lattice, with the least structure $\perp$ being that which assigns the empty set of arrows at each world and base;
- Define an operator T between Herbrand structures that can be considered to be a semantic counterpart to the resolution rule (2). Given an Herbrand structure $\mathcal{H}$, $\mathsf{T}(\mathcal{H})$ is the Herbrand structure built as follows: at each world Δ and each base Γ, add to $\mathcal{H}(\Delta)(\Gamma)$ all of those arrows that can be constructed by one resolution step from arrows that are already forced by $\mathcal{H}$ according to the predicate $\models\!\Rightarrow^{\mathcal{H}}_{\Sigma}$.

It can be shown that such a T is monotone and continuous and so by the Knaster-Tarski theorem has least fixed point $\mathsf{T}^\omega(\perp)$. We obtain, *inter alia*, completeness for the model so obtained. Pleasingly, the least fixed point model is characterized by the Yoneda functor.

5 Remarks

We briefly draw attention to some recent related work: (i) D'Agostino and Gabbay's fibred semantics [1]; (ii) the papers by Ritter, Pym and Wallen that are in Tableaux 96 [24] and CADE-13 [25] consider exploiting the $\lambda\mu$-calculus as a basis for studying classical and intuitionistic search. It is interesting to ask what is the status of $\lambda\mu$ as the language of a logical framework ? This question appears to require careful consideration of how one should represent systems in $\lambda\mu$.

Semantically, it appears that recent work in Axiomatic Domain Theory, *e.g.*, [5], may help to understand the computational structure of models.

Some of the ideas discussed herein were first considered in [17].

References

1. M. D'Agostino and D. Gabbay. Fibred tableaux for multi-implication logics. In: *Proc. Tableaux 96*, P. Miglioli, U. Moscato, D. Mundici and M. Ornagni (eds.), LNCS 1071, 16-35, Springer, 1996.
2. J. Brown and L. Wallen. In preparation, 1996.
3. J. Cartmell. Generalised Algebraic Theories and Contextual Categories. *Ann. Pure Appl. Logic* 32, 209-243, 1986.
4. T. Ehrhard. Thèse, Paris VII, 1988.

[7] This situation can be considered to be a variation on the 'programs-as-worlds' view of the semantics of logic programs [12, 17].

5. M. Fiore. Ph.D. thesis, University of Edinburgh, 1994.
6. R. Harper, F. Honsell and G. Plotkin. A framework for defining logics. *J. Assoc. Comp. Mach.* 40(1), 1993, 143-184.
7. R. Harper, D. Sannella and A. Tarlecki. Structured theory presentations and logic representations. *Ann. Pure Appl. Logic* 67(1994) 113-160.
8. S. Ishtiaq and D. Pym. A relevant analysis of natural deduction. Manuscript, Queen Mary and Westfield College, University of London, 1996.
9. B. Jacobs. Ph.D. thesis, University of Nijmegan, 1991.
10. I. Kant. *Immanual Kants Logik: Ein Handbuch zu Vorlesungen.* G.B. Jäsche, editor, Friedrich Nicolovius, Königsberg, 1800. In translation: R.S. Hartman and W. Schwartz. *Immanual Kant, Logic.* Dover, 1974.
11. P. Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic* 1(1), 1996, 11-60. (Also: Technical Report 2, Scuola di Specializiazzione in Logica Matematica, Dipartimento di Matematica, Università di Siena, 1982.)
12. D. Miller. A logical analysis of modules in logic programming. *J. Logic Programming* 1 & 2 (1989) 79-108.
13. J. Mitchell and E. Moggi. Kripke-style models for typed lambda-calculus. *Ann. Pure Appl. Logic* 51:99-124 1991.
14. R. Paré and D. Schumacher. Abstract Families and the Adjoint Functor Theorem. In: *Lecture Notes in Mathematics 661*, Indexed Categories and their Applications, P.T. Johnstone and R. Paré (editors), 1-125, Springer-Verlag, 1978.
15. F. Pfenning. Logic programming in the LF logical framework. In: G. Huet and G. Plotkin (editors), *Logical Frameworks*, Cambridge University Press, 1991, 149-181.
16. A. Pitts. Categorical Logic. In: *Handbook of Logic in Computer Science, Vol. VI*, A. Abramsky, D. Gabbay and T. Maibaum (editors), Oxford University Press.
17. D. Pym. Ph.D. thesis, University of Edinburgh, 1990.
18. D. Pym. A relevant analysis of natural deduction. Lecture at Workshop, EU Esprit Basic Research Action 3245, Logical Frameworks: Design, Implementation and Experiment, Båstad, Sweden, 1992.
19. D. Pym. A unification algorithm for the $\lambda\Pi$-calculus. Int. J. Foundations of Computer Science 3(3):333-378.
20. D. Pym. A note on the proof theory [of] the $\lambda\Pi$-calculus. *Studia Logica* 54:199-230, 1995.
21. D. Pym. Functorial Kripke models of the $\lambda\Pi$-calculus. Invited Lecture, Isaac Newton Institute for Mathematical Sciences, Semantics Programme, Workshop on Categories and Logic Programming, Cambridge, 1995. Paper(s) in preparation.
22. D. Pym and L. Wallen. Proof-search in the $\lambda\Pi$-calculus. In: G. Huet and G. Plotkin (editors), *Logical Frameworks*, Cambridge University Press, 1991, 309-340.
23. D. Pym and L. Wallen. Logic programming via proof-valued computations. In: ALPUK92, K. Broda (editor), 253-262, Springer-Verlag, WICS, 1992.
24. E. Ritter, D. Pym and L. Wallen. On the intuitionistic force of classical search. In: *Proc. Tableaux 96*, P. Miglioli, U. Moscato, D. Mundici and M. Ornagni (eds.), LNCS 1071, 295-311, Springer, 1996.
25. E. Ritter, D. Pym and L. Wallen. Proof-terms for classical and intuitionistic resolution. To appear: *Proc. CADE-13*, M. MacRobbie (ed.), LNCS, Springer, 1996.
26. R. Seely. Locally cartesian closed categories and type theories. *Math. Proc. Camb. Phil. Soc.*, 95:33-48, 1984.
27. T. Streicher. Ph.D. thesis, Passau, 1988.