Linear metatheory via linear algebra

Robert Atkey[1] and James Wood[1*]

University of Strathclyde, Glasgow, United Kingdom
{robert.atkey,james.wood.100}@strath.ac.uk

Introduction. We introduce a simply typed calculus $\lambda\mathcal{R}$ that allows the use of variables to be constrained by usage annotations in the context which binds them. $\lambda\mathcal{R}$ is a generalisation of existing core type theories for sensitivity analysis [RP10], dependency and confidentiality [ABHR99], linearity [Bar96], and modal validity [PD99]. It is related to quantitative type theory [Atk18], and various coeffect calculi [POM14, BGMZ14, GS14].

One of our insights is that because our usage annotations form a semiring, we have just enough structure to talk about vectors and matrices in the metatheory. We find useful some constructs of linear algebra, culminating in substitution phrased as application of a linear map.

An earlier version of this work was presented at TyDe 2018 [AW18]. The syntax and semantics are formalised in Agda, with the code at https://github.com/laMudri/quantitative/.

Syntax. Our syntax is that of a simply typed λ-calculus modified to let us reason about how variables are used. We assume a partially ordered semiring (posemiring) $(\mathscr{R}, \trianglelefteq, 0, +, 1, *)$ of usage annotations, with elements coloured in green for emphasis. The types are base types (ι_k), functions $(\multimap)$, tensor products $(1, \otimes)$, with products $(\top, \&)$, sums $(0, \oplus)$, and graded bangs $(!_\rho)$. A context $\Gamma^{\mathcal{R}}$ is the combination of a typing context Γ and a usage context $\mathcal{R}$. We see a usage context as a vector over $\mathscr{R}$ generated by the basis made of the variables it contains.

$$\rho, \pi \in \mathscr{R} \qquad A, B, C ::= \iota_k \mid A \multimap B \mid 1 \mid A \otimes B \mid \top \mid A \mathbin{\&} B \mid 0 \mid A \oplus B \mid !_\rho A$$

$$\Gamma, \Delta ::= \cdot \mid \Gamma, x : A \qquad \mathcal{P}, \mathcal{Q}, \mathcal{R} ::= \cdot \mid \mathcal{R}, x^\rho \qquad \Gamma^{\mathcal{R}} ::= \cdot \mid \Gamma^{\mathcal{R}}, x \overset{\rho}{:} A$$

Tensor products are eliminated by pattern matching (each side bound with annotation 1), whereas with products are eliminated by projections. The difference is correspondingly seen in the introduction rules, where the two halves of a tensor product have separate usage, and the two halves of a with product have shared usage (illustrated below).

The rule $\otimes$-I is the archetypal use of $+$. The constraint $\mathcal{P} + \mathcal{Q} \trianglelefteq \mathcal{R}$ says that $\mathcal{R}$ must be at least as permissive as the accumulation of usages in $\mathcal{P}$ and $\mathcal{Q}$. If the addition of the semiring is a join of the order (as in the modality example below), these two types of product are equivalent.

$$\frac{\Gamma^{\mathcal{P}} \vdash M : A \qquad \Gamma^{\mathcal{Q}} \vdash N : B \qquad \mathcal{P} + \mathcal{Q} \trianglelefteq \mathcal{R}}{\Gamma^{\mathcal{R}} \vdash (M, N)_\otimes : A \otimes B}\ \otimes\text{-I} \qquad \frac{\Gamma^{\mathcal{R}} \vdash M : A \qquad \Gamma^{\mathcal{R}} \vdash N : B}{\Gamma^{\mathcal{R}} \vdash (M, N)_\& : A \mathbin{\&} B}\ \&\text{-I}$$

With the graded bang, we see use of $*$ from the annotation posemiring. We read $\rho * \pi$ as applying the action ρ to π. Introduction can be seen as scaling usage. Elimination is by pattern matching, where we bind a new variable with whatever usage annotation the type gave us.

$$\frac{\Gamma^{\mathcal{P}} \vdash M : A \qquad \rho * \mathcal{P} \trianglelefteq \mathcal{R}}{\Gamma^{\mathcal{R}} \vdash [M] : !_\rho A}\ !_\rho\text{-I} \qquad \frac{\Gamma^{\mathcal{P}} \vdash M : !_\rho A \qquad \Gamma^{\mathcal{Q}}, x \overset{\rho}{:} A \vdash N : B \qquad \mathcal{P} + \mathcal{Q} \trianglelefteq \mathcal{R}}{\Gamma^{\mathcal{R}} \vdash \text{let } [x] = M \text{ in } N : B}\ !_\rho\text{-E}$$

*James Wood is supported by an EPSRC Studentship.

The VAR rule (not pictured) at x can only be used in a usage context $\mathcal{R}$ when x has a usage annotation as permissive as 1, and all other variables have annotation as permissive as 0. In other words, x can be used plainly, and all other variables can be discarded. This can be succinctly stated as the constraint $x^1 \trianglelefteq \mathcal{R}$, where x^1 is the xth basis vector.

Substitution. We have two admissible rules leading up to the substitution lemma — subusaging (SUBUSE) and weakening (WEAK) — stated below. In the language of linear algebra, weakening is embedding into a space of higher dimension.

Let $|-|$ denote the length of a context. Usage contexts are taken to be row vectors. A *substitution* σ from $\Gamma^{\mathcal{P}}$ to $\Delta^{\mathcal{Q}}$ comprises a $|\mathcal{Q}| \times |\mathcal{P}|$ matrix $\mathbf{\Sigma}$ such that $\mathcal{Q}\mathbf{\Sigma} \trianglelefteq \mathcal{P}$, and for each $(x : A) \in \Delta$, a term M_x such that $\Gamma^{x^1\mathbf{\Sigma}} \vdash M_x : A$. Then, the simultaneous substitution lemma is proven via the linearity of vector-matrix multiplication.

$$\text{SUBUSE}\ \frac{\Gamma^{\mathcal{P}} \vdash M : A \qquad \mathcal{P} \trianglelefteq \mathcal{Q}}{\Gamma^{\mathcal{Q}} \vdash M : A} \qquad \text{WEAK}\ \frac{\Gamma^{\mathcal{P}} \vdash M : A}{\Gamma^{\mathcal{P}}, \Delta^{\mathbf{0}} \vdash M : A} \qquad \frac{\Delta^{\mathcal{Q}} \vdash N : A \qquad \sigma : \Gamma^{\mathcal{P}} \Rightarrow \Delta^{\mathcal{Q}}}{\Gamma^{\mathcal{P}} \vdash N[\sigma] : A}\ \text{SUBST}$$

The identity substitution, where each variable x is substituted by the term x, is witnessed by the identity matrix. We expect composition to be witnessed by matrix multiplication.

Specialisations. To demonstrate the applicability of $\lambda\mathcal{R}$, and give examples of usage posemirings, we show that certain instances are translatable to DILL [Bar96] and the modal type theory of Pfenning and Davies [PD99]. Future work is to give an equational theory for $\lambda\mathcal{R}$, and show that these translations form an isomorphism.

DILL is a linear type theory where contexts are split between unrestricted and linear variables. To model linearity, we introduce the $\{0, 1, \omega\}$ posemiring. Annotation 0 denotes non-use, 1 linear use, and ω unrestricted use. Addition and multiplication are like the corresponding natural number operations, with ω acting as an infinite element and $1 + 1 = \omega$ in lieu of a 2 element. The order is generated by $0 \trianglelefteq \omega$ and $1 \trianglelefteq \omega$, with no relation between 0 and 1. This says that unrestricted variables can be both discarded and used. We translate a DILL derivation of $\Gamma; \Delta \vdash t : A$ into a $\lambda\mathcal{R}$ derivation of $\Gamma^{\omega}, \Delta^{\mathbf{1}} \vdash M_t : A$. We translate DILL's unannotated ! into $!_{\omega}$. In the translation, we make use of WEAK to ignore 0-use variables introduced by usage separation. When translating the other way, we require that $!_0$ and $!_1$ do not occur in the derivation we are translating. We translate a $\lambda\mathcal{R}$ derivation of $\Gamma^{\omega}, \Delta^{\mathbf{1}}, \Theta^{\mathbf{0}} \vdash M : A$ into a DILL derivation of $\Gamma; \Delta \vdash t_M : A$. This makes use of DILL's Environment Weakening lemma to correct cases where a $\lambda\mathcal{R}$ subderivation was too precise about usage.

Pfenning and Davies' modal type theory is already stated in the form of usage annotations. A variable is annotated either *true* or *valid*. Furthermore, conclusions are only ever of *true* things. This suggests that *true* is the 1 of the posemiring, and we introduce an *unused* annotation to be the 0. The PD variable rule says that both *true* and *valid* assumptions are *true*, so we have *true* $\trianglelefteq$ *valid*. Furthermore, all assumptions can be discarded, so *unused* is the bottom of the order. Addition is the join of this order. The modality $\Box$ is translated to $!_{valid}$, which tells us that *valid* $* \pi =$ *valid* for $\pi \neq$ *unused*. *unused* and *true* are the annihilator and unit of $*$, respectively. Having these definitions in place, the translations are similar to those for DILL.

Semantics. We also have a semantics that captures the intensional properties of programs via families of Kripke indexed relations that refine a simple set-theoretic semantics. This allows us to reconstruct the semantic properties of calculi in prior work for sensitivity analysis [RP10], and dependency and confidentiality [ABHR99], as well as a new calculus for monotonicity analysis.

References

[ABHR99] M. Abadi, A. Banerjee, N. Heintze, and J. G. Riecke. A Core Calculus of Dependency. In *POPL '99*, pages 147–160, 1999.

[Atk18] Robert Atkey. The syntax and semantics of quantitative type theory. In *LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, July 9–12, 2018, Oxford, United Kingdom*, 2018.

[AW18] Robert Atkey and James Wood. Context constrained computation. In *3rd Workshop on Type-Driven Development (TyDe '18), Extended Abstract*, 2018.

[Bar96] Andrew Barber. Dual intuitionistic linear logic. Technical report, The University of Edinburgh, 1996.

[BGMZ14] A. Brunel, M. Gaboardi, D. Mazza, and S. Zdancewic. A Core Quantitative Coeffect Calculus. In *ESOP 2014*, pages 351–370, 2014.

[GS14] Dan R. Ghica and Alex I. Smith. Bounded linear types in a resource semiring. In *ESOP 2014*, pages 331–350, 2014.

[PD99] Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. In *Mathematical Structures in Computer Science*, page 2001, 1999.

[POM14] Tomas Petricek, Dominic A. Orchard, and Alan Mycroft. Coeffects: a calculus of context-dependent computation. In *ICFP 2014*, pages 123–135, 2014.

[RP10] J. Reed and B. C. Pierce. Distance makes the types grow stronger. In P. Hudak and S. Weirich, editors, *ICFP 2010*, pages 157–168, 2010.