Candidates for Substitution*

Healfdene Goguen
hhg@dcs.ed.ac.uk

James McKinna
jhm@dcs.ed.ac.uk

Laboratory for Foundations of Computer Science
Department of Computer Science
The King's Buildings, University of Edinburgh, EH9 3JZ, Scotland

1 Introduction

Context morphisms, that is to say parallel substitutions with additional typing and well-formedness information, have emerged as an important tool in the semantics and metatheory of type theories: they are the basis for categorical semantics of type theory, they yield an appealing syntax for explaining proofs under hypotheses in the informal meaning theory for type theory, and most proofs of strong normalization use context morphisms to strengthen the inductive hypothesis.

Kripke term models [CG90, Gog94] are closed under weakening or context extension, and this is used to show the admissibility of parallel substitution. In this paper we explore a uniform proof of thinning and substitution which isolates several simple candidate-style conditions on context morphisms arising from the Kripke construction. The proof exploits the evident similarity of the properties of thinning and substitution, when considered more generally as a property of context morphisms.

We study this result in the context of Pure Type Systems (PTS), introduced as a general syntactic characterisation of type theories [Bar92]. PTS gives a framework for showing syntactic results, such as strengthening and subject reduction, for a broad class of systems [vBJ93, Geu93].

The second author gave an informal proof of closure under substitution and thinning for Geuvers' systems (which extend the rules

*Submitted to Journal of Functional Programming

for PTS with both η-conversion, and strengthening as a rule of inference, complicating their meta-theory a good deal) using context morphisms [Geu93, p. 104]. This observation was made in the course of a long collaboration with Randy Pollack on machine-checking the meta-theory of PTS [MP97]. Indeed, their first proof of thinning for PTS [MP93] used a particular class of context morphisms, the renamings (see Lemma 1.2 below), motivated by the use of context morphisms in normalization proofs.

A distinctive feature of their approach to PTS, first introduced by Pollack, and motivated by considerations of *syntax-directed systems* [vBJMP94], is the use of an *atomic* weakening rule. This tightens the meta-theory, by making induction over derivations treat fewer cases. One consequence of this presentation is that it then requires some work to prove full weakening as an admissible rule (as a consequence of thinning).

The proof mentioned in Geuvers' thesis required full weakening as a rule. The second author gave a strengthening of the hypotheses, given by the candidate closure conditions described below, which allows the proof to go through in the system with atomic weakening.

Postponing the substantive definitions until Section 2, we summarise our results as follows, with proofs in Section 3:

Lemma 1.1 (Closure under a candidate) *Suppose $\mathcal{S}^{\Delta}_{\Gamma}$ is a candidate for substitution. Then*

$$\gamma \in \mathcal{S}^{\Delta}_{\Gamma}, \Gamma \vdash M : A \;\Rightarrow\; \Delta \vdash M[\gamma] : A[\gamma]$$

Lemma 1.2 *The family of* renamings*,*

$$\mathbf{P}^{\Delta}_{\Gamma} =_{\mathrm{def}} \{\rho : \mathsf{Dom}\,(\Gamma) \longrightarrow_{\mathit{fin}} \mathcal{V} \mid \rho : \Delta \longrightarrow \Gamma\}$$

is a candidate for substitution. Moreover, the least such containing the thinnings*,*

$$\mathbf{T}^{\Delta}_{\Gamma} =_{\mathrm{def}} \{\tau : \mathsf{Dom}\,(\Gamma) \longrightarrow_{\mathit{fin}} \mathcal{V} \mid \forall x \in \mathsf{Dom}\,(\Gamma).\tau(x) = x\}.$$

Lemma 1.3 *If the typing judgement is closed under thinnings,*

$$\tau \in \mathbf{T}^{\Delta}_{\Gamma}, \Gamma \vdash M : A \;\Rightarrow\; \Delta \vdash M[\tau] : A[\tau]$$

then the family of all *substitutions (context morphisms),*

$$\mathbf{S}^{\Delta}_{\Gamma} =_{\mathrm{def}} \{\gamma : \mathsf{Dom}\,(\Gamma) \longrightarrow_{\mathit{fin}} \mathcal{T} \mid \gamma : \Delta \longrightarrow \Gamma\}$$

is a candidate for substitution.

Corollary 1.4 *The typing judgment is closed under parallel substitution.*

Proof. Combine the above three lemmas, appealing twice to the candidate closure lemma.

2 Definitions

We now give the relevant definitions, which motivate the main results of the paper.

2.1 Pure Type Systems

PTS is a class of theories defining a typing judgement for explicitly labelled λ-terms, $\mathcal{T}$, built in the usual way out of variables $\mathcal{V}$ and sorts $\mathcal{S}$. The typing judgement is given by a set of derivation rules, presented in table 1, parameterized by two relations:

- *axioms*, $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$, written $(s_1{:}s_2) \in \mathcal{A}$
- *rules*, $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$, written $(s_1, s_2, s_3) \in \mathcal{R}$.

We write $\Gamma \vdash A$ for $\Gamma \vdash A : s$ for some s, and $\Delta \vdash$ if $\Delta \vdash s_1 : s_2$ for some s_1, s_2 (equivalently, if $\Delta \vdash M : A$ for some M, A).

Remark 2.1 *In this paper we shall treat free and bound variables and associated questions of α-conversion informally. As we shall observe in the proof of Lemma 1.1 below, context morphisms provide a useful mechanism for negotiating the switch between bound and free variables when dealing with binders under substitution.*

Two elementary results about atomic subjects M in the typing judgement $\Gamma \vdash M : A$ are required:

start lemma Every axiom is derivable in every valid context

$$(s_1{:}s_2) \in \mathcal{A}, \Delta \vdash \;\Rightarrow\; \Delta \vdash s_1 : s_2$$

which follows by an easy induction on $\Delta \vdash$.

occurrence lemma Every free variable in a judgement is bound in the context

$$\Gamma \vdash M : A \;\Rightarrow\; \mathsf{FV}(M), \mathsf{FV}(A) \subseteq \mathsf{Dom}(\Gamma)$$

which follows by an easy induction on $\Gamma \vdash M : A$.

Ax	$\bullet \vdash s_1 : s_2$	$(s_1{:}s_2) \in \mathcal{A}$
Start	$\dfrac{\Gamma \vdash A}{\Gamma[x{:}A] \vdash x : A}$	$x \notin \Gamma$
Weak	$\dfrac{\Gamma \vdash \alpha : C \qquad \Gamma \vdash A}{\Gamma[x{:}A] \vdash \alpha : C}$	$\alpha \in \mathcal{P} \cup \mathcal{S},\ x \notin \Gamma$
Pi	$\dfrac{\Gamma \vdash A : s_1 \qquad \Gamma[x{:}A] \vdash B : s_2}{\Gamma \vdash \Pi x{:}A.B : s_3}$	$(s_1, s_2, s_3) \in \mathcal{R}$
Lda	$\dfrac{\Gamma[x{:}A] \vdash M : B \qquad \Gamma \vdash \Pi x{:}A.B}{\Gamma \vdash \lambda x{:}A.M : \Pi x{:}A.B}$	
App	$\dfrac{\Gamma \vdash M : \Pi x{:}A.B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : [N/x]B}$	
Conv	$\dfrac{\Gamma \vdash M : A \qquad \Gamma \vdash B}{\Gamma \vdash M : B}$	$A \simeq_\beta B$

Table 1: The Typing Rules of PTS

2.2 Context morphisms and parallel substitutions

The admissible rule of closure under substitution is usually stated for PTS via the notion of substitution for a single free variable. The proof of this, aside from the delicacies of α-conversion to which we have alluded above, requires an induction loading to go through. It is more convenient for our treatment to distinguish a *parallel* notion of substitution, based on finite mappings. The empty mapping, like the empty context, is denoted by $\bullet$, and mapping extension by $\gamma[x = M]$.

Definition 2.2 *Suppose $\gamma : \mathcal{W} \longrightarrow_{fin} \mathcal{T}$ is a finite mapping with domain $\mathcal{W} \subseteq_{fin} \mathcal{V}$. Then for $M \in \mathcal{T}$ such that $\mathsf{FV}(M) \subseteq \mathcal{W}$, we define the* action of the substitution γ on M*, written $M[\gamma]$, by recursion on M:*

- $s[\gamma] =_{\text{def}} s$;
- $x[\gamma] =_{\text{def}} \gamma(x)$;

- $(M\,N)[\gamma] =_{\mathrm{def}} M[\gamma]\,N[\gamma]$ *;*
- $(\langle x{:}M\rangle N)[\gamma] =_{\mathrm{def}} \langle y{:}M[\gamma]\rangle N[\gamma[x=y]]$ [1]*, where* y *is chosen fresh with respect to* $\mathsf{FV}\,(N[\gamma])\backslash\{x\}$ *in the usual way;*

Remark 2.3 *In the last of the above clauses we have adopted a Curry-style definition of the action on binders, in accordance with our decision to treat variable names informally. Such matters may be treated formally in a more elaborate fashion, following the lines of previous work of Pollack and the second author [MP93, Pol94, MP97]. The definition of the action changes, while the candidate closure conditions do not.*

We may now encapsulate the above-mentioned induction loading once and for all, and moreover unify this notion of substitution with the notion of thinning, by introducing the following definition of *context morphism*, which both enforces validity of the domain context, and inductively ensures well-typedness of the substitution.

Definition 2.4 (Context morphism) *A* context morphism*, usually written* $\gamma : \Delta \longrightarrow \Gamma$*, is a finite mapping* $\gamma : \mathsf{Dom}\,(\Gamma) \longrightarrow_{fin} \mathcal{T}$ *such that:*

- $\bullet : \Delta \longrightarrow \bullet$ *iff* $\Delta \vdash$ *, and*
- $\gamma[x=M] : \Delta \longrightarrow \Gamma[x{:}A]$ *iff*
  - $\gamma : \Delta \longrightarrow \Gamma$*, and* $\Delta \vdash M : A[\gamma]$ *.*

Example 2.5 *Suppose* $\Gamma \vdash N : A$ *and* $\Gamma[x{:}A]\Delta \vdash M : B$ *. Then*

$$\mathbf{id}_\Gamma[x=N]\mathbf{id}_\Delta : \Gamma,\, \Delta[N] \longrightarrow \Gamma[x{:}A]\Delta$$

is a context morphism. Closure under this morphism yields the substitution lemma in its usual form.

Example 2.6 *Suppose that* Γ, Δ *are valid contexts, with* $\Gamma \subseteq \Delta$*, where inclusion is defined naïvely by set-theoretic inclusion of finite maps (so that we embrace* permutation *of binders, as well as extensions). Then the* thinnings*,*

$$\mathbf{T}^{\Delta}_{\Gamma} =_{\mathrm{def}} \{\tau : \mathsf{Dom}\,(\Gamma) \longrightarrow_{fin} \mathcal{V} \mid \forall x{\in}\mathsf{Dom}\,(\Gamma).\tau(x) = x\}$$

define a family of context morphisms.

[1] We employ the notation $\langle v{:}A\rangle a$ of [MP97] to allow us to combine the cases of the two binders.

2.3 Candidates for substitution

We now turn to the definition of the crucial abstraction which underlies our proof of closure under substitution.

Definition 2.7 (Candidate for substitution) *We say a family* $\mathcal{S}^{\Delta}_{\Gamma}$ *of context morphisms* $\gamma : \Delta \longrightarrow \Gamma$ *is a* candidate for substitution *if and only if the following axioms* S_0 – S_2 *hold:*

S0 *if* $\gamma[x = M] \in \mathcal{S}^{\Delta}_{\Gamma[x:A]}$, *then* $\gamma \in \mathcal{S}^{\Delta}_{\Gamma}$;

S1 *if* $\gamma \in \mathcal{S}^{\Delta}_{\Gamma}$, $y \notin \mathsf{Dom}\,(\Delta)$ *and* $\Delta \vdash B$, *then* $\gamma \in \mathcal{S}^{\Delta[y:B]}_{\Gamma}$;

S2 *if* $\gamma \in \mathcal{S}^{\Delta}_{\Gamma}$, $x \notin \mathsf{Dom}\,(\Gamma)$ *and* $\Delta \vdash y : A[\gamma]$, *then* $\gamma[x = y] \in \mathcal{S}^{\Delta}_{\Gamma[x:A]}$.

The motivation for candidate axiom S_0 is as a hygiene condition, whose proof in any given example will usually be immediate from the second clause in the definition of context morphism.

The motivation for candidate axioms S_1 and S_2 is a simple form of closure condition, namely that the candidate is closed under particularly simple extensions of the domain and range of a context morphism. The candidate axiom S_1 arises as the familiar monotonicity condition in Kripke models. It would be trivial to validate if the typing judgement were to admit full weakening, by induction on the proof that γ is a context morphism.

Example 2.8 *The thinnings fail to satisfy candidate axiom* S_2, *and so do* not *form a candidate for substitution. The thinning lemma would follow if we could show closure under the thinnings, but this will follow from Lemmas 1.1 and 1.2.*

Remark 2.9 *We may capture the combined effect of* S_1 *and* S_2 *in the following axiom:*

S$_{\text{ext}}$ *if* $\gamma \in \mathcal{S}^{\Delta}_{\Gamma}$, $x \notin \mathsf{Dom}\,(\Gamma)$, $y \notin \mathsf{Dom}\,(\Delta)$ *and* $\Delta \vdash A[\gamma](\dagger)$, *then* $\gamma[x = y] \in \mathcal{S}^{\Delta[y:A[\gamma]]}_{\Gamma[x:A]}$

We may show that $S_0 + S_{\text{ext}} \Rightarrow S_1 + S_2$ (with two uses of the well-typedness premise $(\dagger)$), and that $S_1 + S_2 \Rightarrow S_{\text{ext}}$, by appealing to atomic weakening. We shall see in the proof of Lemma 1.1 that it is perhaps more convenient to take the $S_0 + S_{\text{ext}}$ axiomatisation.

Remark 2.10 *We have chosen a presentation which uses a named syntax for λ-terms, so the reader more familiar with a deBruijn representation may well wonder what all the fuss is about. After all, although the use of deBruijn terms would force us to incorporate explicit uses of the lift operator in all our definitions, this ostensibly achieves a harmony between the definition of the action $\gamma[M]$ and the* Lda *rule. But we pay the price that the representation imposes an* ordering *on the bindings. This is essentially the temporal order on binding creation (instances of the* Start *rule) time. This not only makes the corresponding definition of the family $\mathbf{T}^{\Delta}_{\Gamma}$ much more complicated (it cannot be given simply in terms of iterated composition of liftings, and makes testing the condition $\Gamma \subseteq \Delta$ a non-trivial computation), but it also obscures the rôle of candidate axiom S_2. This is where permutation may take place, since the premise $\Delta \vdash y : A[\gamma]$ does not specify the creation time of y (nor even that it is well-typed by a sequence of instances of the* Start *and* Wk *rules: the* Conv *rule allows us to exploit possibly non-trivial computation in arriving at the type $A[\gamma]$ for y).*

3 Proofs of the main results

Proof of Lemma 1.1. The proof is by induction on $\Gamma \vdash M : A$. The interesting cases are those of the Start and Lda (Pi is similar) rules. The App and Conv cases require elementary commutation of our parallel substitution notion with simple substitution and conversion respectively, and are not treated here.

Case Start: by assumption, we have $\gamma \in \mathcal{S}^{\Delta}_{\Gamma[x:A]}$, that is to say $\gamma = \delta[x = M]$ where $\Delta \vdash M : A[\delta]$ and $\delta \in \mathcal{S}^{\Delta}_{\Gamma}$, by candidate axiom S_0. We are required to show that $\Delta \vdash x[\gamma] : A[\gamma]$, and this follows immediately, since $x[\gamma] = M$ and $A[\gamma] = A[\delta]$, because $\mathsf{FV}\,(A) \subseteq \mathsf{Dom}\,(\Gamma) = \mathsf{Dom}\,(\delta)$, by the occurrence lemma.

Case Lda: by assumption, we have $\gamma \in \mathcal{S}^{\Delta}_{\Gamma}$, and by induction hypothesis, we have $\forall \gamma, \Delta.\, \gamma \in \mathcal{S}^{\Delta}_{\Gamma} \Rightarrow \Delta \vdash (\Pi x{:}A.B)[\gamma] : s[\gamma]$ and $\forall \delta, \Xi.\, \delta \in \mathcal{S}^{\Xi}_{\Gamma[x:A]} \Rightarrow \Xi \vdash M[\delta] : B[\delta]$. We are required to show that $\Delta \vdash (\lambda x{:}A.M)[\gamma] : (\Pi x{:}A.B)[\gamma]$. By the Lda rule, it suffices to show, for $y \notin \mathsf{Dom}\,(\Delta)$, that $\Delta[y{:}A[\gamma]] \vdash M[\gamma[x = y]] : B[\gamma[x = y]]$. Accordingly, we take $\delta =_{\mathrm{def}} \gamma[x = y]$, and $\Xi =_{\mathrm{def}} \Delta[y{:}A[\gamma]]$, and appeal to the second induction hypothesis. It is at this point alone (and at the corresponding point in the Pi case), that we make appeal to candidate axiom S_{ext}, in order to ensure that $\delta \in \mathcal{S}^{\Xi}_{\Gamma[x:A]}$. It remains to show the well-typedness premise (†) of candidate axiom S_{ext}, namely $\Delta \vdash A[\gamma]$.

But this follows by inversion, since by the first induction hypothesis, we may show that $\Delta \vdash (\Pi x{:}A.B)[\gamma] : s[\gamma]$.

Proof of Lemma 1.2. The verification of candidate axioms S_0 and S_2 is immediate from the definitions of context morphism and the family $\mathbf{P}^{\Delta}_{\Gamma}$, while candidate axiom S_1 requires a little work, namely induction on the proof that $\rho \in \mathbf{P}^{\Delta}_{\Gamma}$, together with an appeal to atomic weakening in both the base and step cases. The inclusion, and the fact that it is strict, and minimal, is again immediate.

Proof of Lemma 1.3. Immediate from previous considerations: the thinnings yield the instances of full weakening required to validate candidate axiom S_1.

4 Conclusions

We have given a simple and uniform proof of two meta-theoretic properties of PTS, thinning and substitution. We began by noticing that the notion of context morphism can be used to express the two properties. We identified several candidate-style conditions on sets of context morphisms, and showed that the judgements of PTS are closed under sets which satisfy these conditions. Finally, we showed that these conditions are satisfied by the class of renamings, and that the class of all context morphisms also satisfies them, knowing that the judgements of PTS are closed under renamings.

We analysed an originally blocked proof, that the judgements are closed under thinnings, and can now observe the rôle of renamings in proofs of this kind: the renamings are exactly the least set of context morphisms containing the thinnings, and closed under the candidate conditions.

Acknowledgements Our work has throughout been influenced by the ideas of Thierry Coquand. We are also grateful to our other colleagues in the EU Esprit Working Group "TYPES", especially Randy Pollack, Zhaohui Luo, and Herman Geuvers.

References

[Bar92] Henk Barendregt. Lambda calculi with types. In Abramsky, Gabbai, and Maibaum, editors, *Handbook of Logic in Computer Science*, volume II. Oxford University Press, 1992.

[CG90] Thierry Coquand and Jean Gallier. A proof of strong normalization for the theory of constructions using a Kripke-like interpretation. In *Workshop on Logical Frameworks–Preliminary Proceedings*, 1990.

[Geu93] Herman Geuvers. *Logics and Type Systems*. PhD thesis, Department of Mathematics and Computer Science, University of Nijmegen, 1993.

[Gog94] Healfdene Goguen. *A Typed Operational Semantics for Type Theory*. PhD thesis, University of Edinburgh, August 1994.

[MP93] James McKinna and Robert Pollack. Pure Type Systems formalized. In M.Bezem and J.F.Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications, TLCA'93, Utrecht*, number 664 in LNCS, pages 289–305. Springer-Verlag, March 1993.

[MP97] James McKinna and Robert Pollack. Some λ-calculus and type theory formalized. *Journal of Automated Reasoning*, 1997. Submitted for publication in the special issue on formal proof. Available by anonymous ftp from `ftp://ftp.dcs.ed.ac.uk/pub/lego/McKinnaPollack97.ps.gz`.

[Pol94] Robert Pollack. *The Theory of LEGO: A Proof Checker for the Extended Calculus of Constructions*. PhD thesis, University of Edinburgh, 1994. `ftp://ftp.dcs.ed.ac.uk/pub/lego/thesis-pollack.ps.Z`.

[vBJ93] L.S. van Benthem Jutting. Typing in Pure Type Systems. *Information and Computation*, 105(1):30–41, July 1993.

[vBJMP94] L.S. van Benthem Jutting, James McKinna, and Robert Pollack. Checking algorithms for Pure Type Systems. In Henk Barendregt and Tobias Nipkow, editors, *Types for Proofs and Programs: International Workshop TYPES'93, Nijmegen, May 1993, Selected Papers*, volume 806 of *LNCS*, pages 19–61. Springer-Verlag, 1994.