A short note explaining the bug in the APAL term assignment formulation of FILL

Harley Eades III

August 2014

In this short note I give the details of Bierman's counterexample [1] to cut elimination of the term assignment formulation of FILL first given in [4]. I first reformulate his counterexample into our definition of FILL, and then comment on the reason for the counterexample. Following this I reformulate the counterexample in the dependency tracking system proposed by Braüner and de Paiva in [2] and revised by the same authors in [3]. In this reformulation we will see that the rule proposed by Bellin but communicated by Bierman in [1] is the proper left rule for par.

Conjecture: Revise if incorrect

1 The Parl Inference Rule

The existing Parl inference rule is as follows:

$$\frac{\Gamma, x : A \vdash d_i : C_i \qquad \Gamma', y : B \vdash f_j : D_j}{\Gamma, \Gamma', z : A \parr B \vdash \mathsf{let}\, z\, \mathsf{be}\, x \parr - \,\mathsf{in}\, d_i : C_i \mid \mathsf{let}\, z\, \mathsf{be}\, - \parr y \,\mathsf{in}\, f_j : D_j}\ \textsc{Parl}$$

In the terms $\mathsf{let}\, t_1\, \mathsf{be}\, x \parr * \,\mathsf{in}\, t_2$ and $\mathsf{let}\, t_1\, \mathsf{be}\, * \parr y \,\mathsf{in}\, t_2$ the variables x and y in the patterns are bound in t_2. So when applying the Parl rule we bind both the free variable x in d_i, and the free variable y in each f_i. Now notice that we do this binding even when the variables are not free in the respective terms. Furthermore, as a result of binding these pattern variables we carry along the newly introduced free variable z. It is this global binding across the entire righthand size context along with introducing the free variable z in each term that results in the counterexample of Bierman.

2 Bierman's Counterexample

First lets recall the cut-elimination commuting conversion that is the locus of the counterexample. The following cut:

$$\frac{\Gamma \vdash d : C \mid g_k : D \qquad \dfrac{x : C, w : A, \Gamma' \vdash e_i : D' \qquad z : B, \Gamma'' \vdash t_i : D''}{x : C, y : A \parr B, \Gamma', \Gamma'' \vdash \mathsf{let}\, y\, \mathsf{be}\, w \parr * \,\mathsf{in}\, e_i : D' \mid \mathsf{let}\, y\, \mathsf{be}\, * \parr z \,\mathsf{in}\, t_i : D''}\ \textsc{Parl}}{\Gamma, y : A \parr B, \Gamma', \Gamma'' \vdash [d/x](\mathsf{let}\, y\, \mathsf{be}\, w \parr * \,\mathsf{in}\, e_i) : D' \mid [d/x](\mathsf{let}\, y\, \mathsf{be}\, * \parr z \,\mathsf{in}\, t_j) : D'' \mid g_k : D}\ \textsc{Cut}$$

Converts into the following:

$$\frac{\dfrac{\Gamma \vdash d : C \mid g_k : D \qquad x : C, w : A, \Gamma' \vdash e_i : D'}{w : A, \Gamma' \vdash [d/x]e_i : D'}\ \textsc{Cut} \qquad z : B, \Gamma'' \vdash t_i : D''}{\Gamma, y : A \parr B, \Gamma', \Gamma'' \vdash \mathsf{let}\, y\, \mathsf{be}\, w \parr * \,\mathsf{in}\, [d/x]e_i : D' \mid \mathsf{let}\, y\, \mathsf{be}\, * \parr z \,\mathsf{in}\, t_j : D'' \mid \mathsf{let}\, y\, \mathsf{be}\, w \parr * \,\mathsf{in}\, g_k : D}\ \textsc{Parl}$$

Notice that in the above cut, the Parl rule commutes with Cut. So again, we bind w as a pattern variable in each e_i, and z in each t_i regardless of whether or not these are actually free in any of the terms. In addition, we introduce z into each of these terms.

Next we give Bierman's counterexample. The following uses the first rule given above.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$v : A \vdash v : A$}
\RightLabel{$\scriptstyle\mathrm{Pr}$}
\UnaryInfC{$v : A \vdash v : A \mid \circ : \bot$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$x : A \vdash x : A$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$y : B \vdash y : B$}
\RightLabel{$\scriptstyle\mathrm{Tr}$}
\BinaryInfC{$x : A, y : B \vdash x \otimes y : A \otimes B$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$w : C \vdash w : C$}
\RightLabel{$\scriptstyle\mathrm{Parl}$}
\BinaryInfC{$x : A, z : B \parr C \vdash \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y : A \otimes B \mid \mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w : C$}
\RightLabel{$\scriptstyle\mathrm{Cut}$}
\BinaryInfC{$v : A, z : B \parr C \vdash \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y : A \otimes B \mid \mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w : C \mid \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Parr}$}
\UnaryInfC{$v : A, z : B \parr C \vdash (\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w) : ((A \otimes B) \parr C) \mid \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Impr}$}
\UnaryInfC{$v : A \vdash \lambda z.((\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w)) : (B \parr C) \multimap ((A \otimes B) \parr C) \mid \circ : \bot$}
\end{prooftree}
$$

Next we use the second derived rule above to commute the cut in the previous derivation past the PARL rule:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$v : A \vdash v : A$}
\RightLabel{$\scriptstyle\mathrm{Pr}$}
\UnaryInfC{$v : A \vdash v : A \mid \circ : \bot$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$x : A \vdash x : A$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$y : B \vdash y : B$}
\RightLabel{$\scriptstyle\mathrm{Tr}$}
\BinaryInfC{$x : A, y : B \vdash x \otimes y : A \otimes B$}
\RightLabel{$\scriptstyle\mathrm{Cut}$}
\BinaryInfC{$y : B, v : A \vdash v \otimes y : A \otimes B \mid \circ : \bot$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$w : C \vdash w : C$}
\RightLabel{$\scriptstyle\mathrm{Parl}$}
\BinaryInfC{$v : A, z : B \parr C \vdash \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y : A \otimes B \mid \mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w : C \mid \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in} \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Parr}$}
\UnaryInfC{$v : A, z : B \parr C \vdash ((\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w)) : (A \otimes B) \parr C \mid \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in} \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Impr}$}
\UnaryInfC{$v : A \vdash \lambda z.((\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w)) : (B \parr C) \multimap ((A \otimes B) \parr C) \mid \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in} \circ : \bot$}
\end{prooftree}
$$

Now notice that as a result of the rule PARL rule a fresh free variable z – colored blue when it is considered free – is introduced, and then let-bound in every term in the righthand side context. Furthermore, we bind y and w in terms which do not depend on them, for example, we bind y in $\circ$. Furthermore, we introduce z into each of these terms, especially, the rightmost term. Thus, the application of the IMPR rule is in error, because z occurs in the right most term.

3 Bierman's Counterexample in the Dependency-Relation Formalization

Next we give Bierman's counterexample in the dependency-relation formalization. To obtain the derivations we simply erase all the terms:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$v : A \vdash v : A$}
\RightLabel{$\scriptstyle\mathrm{Pr}$}
\UnaryInfC{$v : A \vdash v : A \mid \circ : \bot$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$x : A \vdash x : A$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$y : B \vdash y : B$}
\RightLabel{$\scriptstyle\mathrm{Tr}$}
\BinaryInfC{$x : A, y : B \vdash x \otimes y : A \otimes B$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$w : C \vdash w : C$}
\RightLabel{$\scriptstyle\mathrm{Parl}$}
\BinaryInfC{$x : A, z : B \parr C \vdash \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y : A \otimes B \mid \mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w : C$}
\RightLabel{$\scriptstyle\mathrm{Cut}$}
\BinaryInfC{$v : A, z : B \parr C \vdash \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y : A \otimes B \mid \mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w : C \mid \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Parr}$}
\UnaryInfC{$v : A, z : B \parr C \vdash (\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w) : ((A \otimes B) \parr C) \mid \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Impr}$}
\UnaryInfC{$v : A \vdash \lambda z.((\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w)) : (B \parr C) \multimap ((A \otimes B) \parr C) \mid \circ : \bot$}
\end{prooftree}
$$

Next is the derivation after the commute:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$v : A \vdash v : A$}
\RightLabel{$\scriptstyle\mathrm{Pr}$}
\UnaryInfC{$v : A \vdash v : A \mid \circ : \bot$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$x : A \vdash x : A$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$y : B \vdash y : B$}
\RightLabel{$\scriptstyle\mathrm{Tr}$}
\BinaryInfC{$x : A, y : B \vdash x \otimes y : A \otimes B$}
\RightLabel{$\scriptstyle\mathrm{Cut}$}
\BinaryInfC{$y : B, v : A \vdash v \otimes y : A \otimes B \mid \circ : \bot$}
\AxiomC{}
\RightLabel{$\scriptstyle\mathrm{Ax}$}
\UnaryInfC{$w : C \vdash w : C$}
\RightLabel{$\scriptstyle\mathrm{Parl}$}
\BinaryInfC{$v : A, z : B \parr C \vdash \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y : A \otimes B \mid \mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w : C \mid \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in} \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Parr}$}
\UnaryInfC{$v : A, z : B \parr C \vdash ((\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w)) : (A \otimes B) \parr C \mid \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in} \circ : \bot$}
\RightLabel{$\scriptstyle\mathrm{Impr}$}
\UnaryInfC{$v : A \vdash \lambda z.((\mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in}\ v \otimes y) \parr (\mathsf{let}\ z\ \mathsf{be}\ * \parr w\ \mathsf{in}\ w)) : (B \parr C) \multimap ((A \otimes B) \parr C) \mid \mathsf{let}\ z\ \mathsf{be}\ y \parr * \ \mathsf{in} \circ : \bot$}
\end{prooftree}
$$

References

[1] G.M. Bierman. A note on full intuitionistic linear logic. *Annals of Pure and Applied Logic*, 79(3):281 – 287, 1996.

[2] Torben Brauner and Valeria Paiva. Cut-elimation for full intuitionistic linear logic. BRICS 395, Computer Laboratory, University of Cambridge, 1996.

[3] Torben Brauner and Valeria Paiva. A formulation of linear logic based on dependency-relations. In Mogens Nielsen and Wolfgang Thomas, editors, *Computer Science Logic*, volume 1414 of *Lecture Notes in Computer Science*, pages 129–148. Springer Berlin Heidelberg, 1998.

[4] Martin Hyland and Valeria de Paiva. Full intuitionistic linear logic (extended abstract). *Annals of Pure and Applied Logic*, 64(3):273 – 291, 1993.

A The full specification of FILL

$term\_var,\ w,\ x,\ y,\ z,\ v$			
$index\_var,\ i,\ j,\ k$			
$form,\ A,\ B,\ C,\ D,\ E$	$::=$		
	$\mid$	I	
	$\mid$	$\perp$	
	$\mid$	$A \multimap B$	
	$\mid$	$A \otimes B$	
	$\mid$	$A \parr B$	
	$\mid$	(A)	S
$patterns,\ p$	$::=$		
	$\mid$	$*$	
	$\mid$	x	
	$\mid$	$p_1 \otimes p_2$	
	$\mid$	$p_1 \parr p_2$	
	$\mid$	$-$	
$term,\ t,\ e,\ d,\ f,\ g,\ u$	$::=$		
	$\mid$	x	
	$\mid$	$*$	
	$\mid$	$\circ$	
	$\mid$	$e_1 \otimes e_2$	
	$\mid$	$e_1 \parr e_2$	
	$\mid$	$\lambda x.t$	
	$\mid$	$\mathsf{let}\ t\ \mathsf{be}\ p\ \mathsf{in}\ e$	
	$\mid$	$f\ e$	
	$\mid$	$[t/x]t'$	M
	$\mid$	$[t/x, e/y]t'$	M
	$\mid$	(t)	S
	$\mid$	t	M
	$\mid$	t	M
Γ	$::=$		
	$\mid$	$x : A$	
	$\mid$	$\cdot$	
	$\mid$	Γ, Γ'	
	$\mid$	$x : A$	

	$\mid$	A
Δ	$::=$	
	$\mid$	$t : A$
	$\mid$	$\cdot$
	$\mid$	$\Delta \mid \Delta'$
	$\mid$	Δ
	$\mid$	A
	$\mid$	Δ, Δ'
formula	$::=$	
	$\mid$	*judgement*
	$\mid$	$formula_1 \quad formula_2$
	$\mid$	(*formula*)
	$\mid$	$x \notin \mathsf{FV}(\Delta)$
InferRules	$::=$	
	$\mid$	$\Gamma \vdash \Delta$
	$\mid$	$f = e$
judgement	$::=$	
	$\mid$	*InferRules*
user_syntax	$::=$	
	$\mid$	*term_var*
	$\mid$	*index_var*
	$\mid$	*form*
	$\mid$	*patterns*
	$\mid$	*term*
	$\mid$	Γ
	$\mid$	Δ
	$\mid$	*formula*

$\boxed{\Gamma \vdash \Delta}$

$$\frac{}{x : A \vdash x : A} \quad \textsc{Ax}$$

$$\frac{\Gamma \vdash t : A \mid \Delta \quad y : A, \Gamma' \vdash f_i : B_i}{\Gamma, \Gamma' \vdash \Delta \mid [t/y]f_i : B_i} \quad \textsc{Cut}$$

$$\frac{\Gamma \vdash e_i : A_i}{\Gamma, x : I \vdash \mathsf{let}\, x\, \mathsf{be}\, * \,\mathsf{in}\, e_i : A_i} \quad \textsc{Il}$$

$$\frac{}{\cdot \vdash * : I} \quad \textsc{Ir}$$

$$\frac{\Gamma, x : A, y : B \vdash f_i : C_i}{\Gamma, z : A \otimes B \vdash \mathsf{let}\, x\, \mathsf{be}\, x \otimes y\, \mathsf{in}\, f_i : C_i} \quad \textsc{Tl}$$

$$\frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma' \vdash f : B \mid \Delta'}{\Gamma, \Gamma' \vdash e \otimes f : A \otimes B \mid \Delta \mid \Delta'} \quad \textsc{Tr}$$

$$\frac{}{x : \perp \vdash \cdot} \quad \textsc{Pl}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \circ : \perp \mid \Delta} \quad \textsc{Pr}$$

$$\frac{\Gamma, x : A \vdash d_i : C_i \quad \Gamma', y : B \vdash f_j : D_j}{\Gamma, \Gamma', z : A \parr B \vdash \mathsf{let}\, z\, \mathsf{be}\, x \parr - \,\mathsf{in}\, d_i : C_i \mid \mathsf{let}\, z\, \mathsf{be}\, - \parr y\, \mathsf{in}\, f_j : D_j} \quad \textsc{Parl}$$

$$\frac{\Gamma \vdash \Delta \mid e : A \mid f : B \mid \Delta'}{\Gamma \vdash \Delta \mid e \parr f : A \parr B \mid \Delta'} \quad \textsc{Parr}$$

$$\frac{\Gamma \vdash e : A \mid \Delta \quad \Gamma', x : B \vdash f_i : C_i}{\Gamma, y : A \multimap B, \Gamma' \vdash [y\, e/x]f_i : C_i \mid \Delta} \quad \textsc{Impl}$$

$$\frac{\Gamma, x : A \vdash e : B \mid \Delta \quad x \notin \mathsf{FV}(\Delta)}{\Gamma \vdash \lambda x.e : A \multimap B \mid \Delta} \quad \textsc{Impr}$$

$$\frac{}{A' \vdash A''} \quad \textsc{dax}$$

$$\frac{\begin{array}{c}\Gamma_1 \vdash B', \Delta_1 \\ \Gamma_2, B'' \vdash \Delta_2\end{array}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \quad \textsc{dcut}$$

$$\frac{\Gamma, A, B \vdash D}{\Gamma, A \otimes B \vdash D} \quad \textsc{Dtl}$$

$$\frac{\begin{array}{c}\Gamma_1 \vdash A, D_1 \\ \Gamma_2 \vdash B, D_2\end{array}}{\Gamma_1, \Gamma_2 \vdash A \otimes B, D_1, D_2} \quad \textsc{Dtr}$$

$$\frac{\Gamma \vdash D}{\Gamma, I \vdash D} \quad \textsc{Dil}$$

$$\frac{}{\cdot \vdash I} \quad \textsc{Dir}$$

$$\frac{\begin{array}{c}\Gamma_1, A \vdash D_1 \\ \Gamma_3, B \vdash D_2\end{array}}{\Gamma_1, \Gamma_3, A \parr B \vdash D_1, D_2} \quad \textsc{Dparl}$$

$$\frac{\Gamma \vdash A, B, D}{\Gamma \vdash A \parr B, D} \quad \textsc{Dparr}$$

$$\frac{}{\perp \vdash \cdot} \quad \textsc{Dpl}$$

$$\frac{\Gamma \vdash D}{\Gamma \vdash \perp, D} \quad \textsc{Dpr}$$

$$\frac{\begin{array}{c}\Gamma_1 \vdash A, D_1 \\ \Gamma_2, B \vdash D_2\end{array}}{\Gamma_1, \Gamma_2, A \multimap B \vdash D_1, D_2} \quad \textsc{Dimpl}$$

$$\frac{\Gamma, A \vdash B, D}{\Gamma \vdash A \multimap B, D} \quad \textsc{Dimpr}$$

$\boxed{f = e}$

$$\frac{}{(\lambda x.e)\; e' = [e'/x]e} \quad \textsc{Eq\_Beta}$$

$$\frac{}{\lambda x.f\; x = f} \quad \textsc{Eq\_Eta}$$

$$\frac{}{\mathsf{let}\ *\ \mathsf{be}\ *\ \mathsf{in}\ e = e}\ \textsc{Eq\_I}$$

$$\frac{}{\mathsf{let}\ u\ \mathsf{be}\ *\ \mathsf{in}\ [*/z]f = [u/z]f}\ \textsc{Eq\_Stp}$$

$$\frac{}{\mathsf{let}\ e \otimes t\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ u = [e/x, t/y]u}\ \textsc{Eq\_T1}$$

$$\frac{}{\mathsf{let}\ u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ [x \otimes y/z]f = [u/z]f}\ \textsc{Eq\_T2}$$

$$\frac{}{\mathsf{let}\ u \parr t\ \mathsf{be}\ x \parr -\ \mathsf{in}\ e = [u/x]e}\ \textsc{Eq\_P1}$$

$$\frac{}{\mathsf{let}\ u \parr t\ \mathsf{be}\ - \parr y\ \mathsf{in}\ e = [t/y]e}\ \textsc{Eq\_P2}$$

$$\frac{}{(\mathsf{let}\ x\ \mathsf{be}\ x \parr -\ \mathsf{in}\ x) \parr (\mathsf{let}\ u\ \mathsf{be}\ - \parr y\ \mathsf{in}\ y) = u}\ \textsc{Eq\_P3}$$