Confluence by Higher-Order Multi–One Critical pairs with an application to the Functional Machine Calculus

Willem Heijltjes and Vincent van Oostrom*

Department of Computer Science, University of Bath, United Kingdom
wbh22@bath.ac.uk, vvo21@bath.ac.uk

Abstract

The functional machine calculus (FMC) is a model of higher-order computation with effects, and is known to be confluent. Here we re-prove confluence of the FMC via higher-order term rewriting, embedding the FMC in a 3$^{\text{rd}}$-order PRS. Our main contribution is a higher-order version of the critical-pair-criterion that was developed by Okui for first-order TRSs, requiring all multi–one critical peaks to be many–multi joinable.

1 The Functional Machine Calculus

The *Functional Machine Calculus* (FMC) is a model of higher-order computation with effects [1]. It generalizes the λ-calculus and is known to preserve its main properties of confluence and simply-typed termination, while it encodes *reader/writer effects* (state, I/O, probabilities, nondeterminism) and strategies including call–by–name, call–by–value, and call–by–push–value [3]. In this section we recapitulate the FMC in its traditional presentation. In Section 2 we show how it can be embedded in a 3$^{\text{rd}}$-order positional pattern rewrite system. Via this embedding confluence of the FMC is then regained as an *instance* of a critical pair criterion for positional PRSs (Definition 2), generalising Okui's criterion for TRSs [5], as shown in Section 3.

The intuition for the FMC is of λ-terms as instruction sequences for a simple stack machine. Application $M\,N$, written $[N].\,M$, pushes N to the stack and continues with M; abstraction $\lambda x.\,M$, written $\langle x\rangle.\,M$, pops a term N and continues with $\{N/x\}M$ (the substitution of N for x in M). The FMC then consists of two generalizations. One, to multiple stacks, indexed by *locations* a, b, $c, \ldots$ in which application and abstraction are parameterized, $[N]a.\,M$ and $a\langle x\rangle.\,M$. As well as the main stack, these model input and output streams, memory cells, and random generators. Two, with the empty sequence $\star$ and *sequential composition*, implemented by making the variable construct a prefix $x.\,M$; this gives control over evaluation behaviour and models strategies. Both generalizations have interesting consequences for reduction. First, a redex consists of an application and abstraction at the same location, $[N]a\ldots a\langle x\rangle.\,M$, possibly with operations on other locations in between. Second, to substitute N for x in $x.\,M$ involves sequential composition $N\,;M$.

Definition 1. FMC-terms *are given by the following grammar, where* $a\langle x\rangle.\,M$ *binds* x *in* M, *and considered modulo* α*-equivalence. (Trailing* $.\star$ *may be omitted.)*

$$M,\,N,\,P \quad ::= \quad \star \;\mid\; x.\,M \;\mid\; [N]a.\,M \;\mid\; a\langle x\rangle.\,M$$

We define β-reduction *by the rewrite rule schema below (closed under all contexts)*

$$[N]a.\,H.\,a\langle x\rangle.\,M \;\rightarrow\; H.\,\{N/x\}M \qquad (a \notin \mathsf{loc}(H),\;\mathsf{bv}(H)\cap\mathsf{fv}(N)=\varnothing)$$

*Supported by EPSRC Project EP/R029121/1 Typed lambda-calculi with sharing and unsharing.

where H *is a* head context *with* binding variables $\mathsf{bv}(H)$ *and* locations $\mathsf{loc}(H)$ *as defined below, writing* $H.\,M$ *for* $H\{M\}$ *(*H *with the hole* $\{\}$ *replaced by* M*).*

$$H \;::=\; \{\} \mid [N]a.\,H \mid a\langle x\rangle.\,H \qquad \begin{aligned} \mathsf{bv}(\{\}) &= \varnothing \\ \mathsf{bv}([M]a.\,H) &= \mathsf{bv}(H) \\ \mathsf{bv}(a\langle x\rangle.\,H) &= \mathsf{bv}(H)\cup\{x\} \end{aligned} \qquad \begin{aligned} \mathsf{loc}(\{\}) &= \varnothing \\ \mathsf{loc}([M]a.\,H) &= \mathsf{loc}(H)\cup\{a\} \\ \mathsf{loc}(a\langle x\rangle.\,H) &= \mathsf{loc}(H)\cup\{a\} \end{aligned}$$

Composition $N\,;M$ *and substitution* $\{M/x\}N$ *are capture-avoiding, and are as follows.*

$$\begin{aligned} \star\,;M &= M & [P]a.\,N\,;M &= [P]a.\,(N\,;M) \\ x.\,N\,;M &= x.\,(N\,;M) & a\langle y\rangle.\,N\,;M &= a\langle y\rangle.\,(N\,;M) \quad (y\notin\mathsf{fv}(M)) \end{aligned}$$

$$\begin{aligned} \{P/x\}\star &= \star & \{P/x\}[N]a.\,M &= [\{P/x\}N]a.\,\{P/x\}M \\ \{P/x\}x.\,M &= P\,;\{P/x\}M & \{P/x\}a\langle x\rangle.\,M &= a\langle x\rangle.\,M \\ \{P/x\}y.\,M &= y.\,\{P/x\}M \quad (x\neq y) & \{P/x\}a\langle y\rangle.\,M &= a\langle y\rangle.\,\{P/x\}M \quad (y\notin\mathsf{fv}(P)) \end{aligned}$$

The pure λ-calculus may be embedded in the FMC by choosing a *main* location λ, omitted from terms for compactness, and defining $\lambda x.M = \langle x\rangle.\,M$ and $M\,N = [N].\,M$.

Example 1. *To model global store, a cell is a dedicated location* a *with* lookup $!a$ *encoded by* $a\langle x\rangle.\,[x]a.\,x$ *and* update $N := a\,;M$ *by* $a\langle\_\rangle.\,[N]a.\,M$ *(where* $\_$ *is a non-binding variable). The following example term stores* $\lambda f.\,f\,(f\,3)$ *to the cell* a*, and then retrieves it to call it on* $\lambda y.\,y+1$*. Overall, it should update* a *and return* 5*, which FMC reduction indeed exposes. (Underlining indicates a redex, and colours trace subterms through translations and reductions.)*

$$\begin{aligned} a := (\lambda f.\,f\,(f\,3))\,;!a\,(\lambda y.\,y+1) \;&=\; a\langle\_\rangle.\,\underline{[\langle f\rangle.\,[[3].\,f].\,f]a}.\,[\langle y\rangle.\,[y].\,[1].\,+].\,\underline{a\langle x\rangle}.\,[x]a.\,x \\ &\to\; a\langle\_\rangle.\,\underline{[\langle y\rangle.\,[y].\,[1].\,+]}.\,[\langle f\rangle.\,[[3].\,f].\,f]a.\,\underline{\langle f\rangle}.\,[[3].\,f].\,f \\ &\to\; a\langle\_\rangle.\,[\langle f\rangle.\,[[3].\,f].\,f]a.\,[[3].\,\langle y\rangle.\,[y].\,[1].\,+].\,\langle y\rangle.\,[y].\,[1].\,+ \\ &\twoheadrightarrow\; a\langle\_\rangle.\,[\langle f\rangle.\,[[3].\,f].\,f]a.\,5 \\ &=\; a := (\lambda f.\,f\,(f\,3))\,;5 \end{aligned}$$

2 Embedding the FMC in a PRS

We show the FMC can be embedded in a 3$^{\text{rd}}$-order *pattern rewrite system* (PRS), with which we assume familiarity [4, 9]. Since we will build on it below, we revisit the standard embedding of the pure λ-calculus in a 2$^{\text{nd}}$-order PRS ([4, Example 3.4],[9, Examples 11.2.6(i),11.2.22(ii)]).

Example 2. *The PRS* $\mathcal{L}am$ *has a single base type* term*, two simply typed constants for abstraction and application:* $\mathsf{lam} : (\mathsf{term}\to\mathsf{term})\to\mathsf{term}$ *and* $\mathsf{app} : \mathsf{term}\to\mathsf{term}\to\mathsf{term}$*, and rules:*

$$\begin{aligned} \mathsf{beta} &: \lambda FS.\mathsf{app}(\mathsf{lam}\lambda x.F(x), S) \to \lambda FS.F(S) \\ \mathsf{eta} &: \lambda S.\mathsf{lam}(\lambda x.\mathsf{app}(S,x)) \to \lambda S.S \end{aligned}$$

with variables $x:\mathsf{term}$*,* $F:\mathsf{term}\to\mathsf{term}$ *and* $S:\mathsf{term}$*, and rules, which are symbols in our setting having the type of their lhs / rhs,* $\mathsf{beta} : (\mathsf{term}\to\mathsf{term})\to\mathsf{term}\to\mathsf{term}$*, and* $\mathsf{eta} : \mathsf{term}\to\mathsf{term}$*.*

Objects The objects of a PRS are simply typed λ-terms modulo $\alpha\beta\eta$ for a collection of *base* types, and a signature of *symbols*. We refer to the simply typed $\lambda\alpha\beta\eta$-calculus as the *substitution* calculus of PRSs as it brings about the standard notions of matching, substitution and occurrence [8, 6]. We assume λ-terms to be in η-expanded form ([4, p. 5],[9, Convention 11.2.12]). *Terms* then are λ-terms also in β-normal form, serving as representatives (unique up to α) of $\alpha\beta\eta$-equivalence classes. The *parameter passing* of rewrite rules is brought about by the substitution calculus, *matching* by β-expansion and *substitution* by β-reduction. To separate the *replacement* aspect of rewrite rules from their parameter passing aspect [9, Definition 11.2.25(iv)], rewrite rules are closed. To facilitate defining *occurrences* below, we overline a subterm of a λ-term to denote the λ-term (recursively) obtained by removing the overlining, and if the subterm is a β-redex then contracting it and overlining the created β-redexes.

In [2, Lemma 2] we established that for first-order term rewriting there is a perfect rapport between the *inductive* and *geometric* views of the notion of *occurrence*. We consider the higher-order case: in the inductive view an occurrence of a *pattern* π in a λ-term t then is a *β-expansion* of t to a λ-term $(\lambda\, x.s)\,\pi$ (cf. [8, Definition 2.9]), and in the geometric view a *pat* P is a certain *subset* of the positions of the tree [4, p. 5] of t (cf. [9, Proposition 8.6.25]). To make the rapport perfect, we restrict ourselves to occurrences of *patterns* [4, Definition 3.1] that are *rule-patterns* [9, Definition 11.2.18(ii)], *local* [7, Footnote 4], and moreover such that the free variables are in pre-order and the parameters in outside-in order; these are *positional* patterns:

Definition 2 (Inductive view). *A* positional pattern π *is a closed λ-term of shape $\lambda\, \boldsymbol{F}.f(\boldsymbol{t})$ such that (*head-defined*) f is a function symbol and $f(\boldsymbol{t})$ is of base type; (*linear*) π is linear in $\boldsymbol{F}$, each F_i occurs once; and (*fully-extended*) each $F \in \boldsymbol{F}$ occurs in π as $F(\boldsymbol{x})$ where x is the list of (η-expansions of) variables that are bound above F in $f(\boldsymbol{t})$, in outside-in order. To avoid clutter we may drop the initial binders $\boldsymbol{F}$ of π. We incongruously refer to such an F as a* free variable *of π and to its arguments $\boldsymbol{x}$ as its* parameters. *A rule / PRS is* positional *if its lhs is / rules are. If for a vector $\boldsymbol{\pi}$ of positional patterns and λ-term t, we have $\overline{(\lambda\, \boldsymbol{F}.s)\,\boldsymbol{\pi}} = t$ with s linear in $\boldsymbol{F}$, we speak of a* multipattern $\boldsymbol{\pi}$ in t. *They are taken up to permutation of $\boldsymbol{\pi}$, $\boldsymbol{F}$.*

Definition 3 (Geometric view). *A* pat *in a λ-term t is a non-empty set P of positions in the tree*[1] *of t such that (*convex*) if $p, q \in P$ then all positions on the path between p and q are in P [2, Footnote 4]; (*rigid*) if $t(p)$ is a variable and $p \in P$, then it is bound by a λ-abstraction at a position in P; (*base-fringe*) $t_{|p}$ is of base type for p the root of P or a child not in P of a position in P; and (*normal*) if $t(p)$ is an application and $p \in P$, then its left child is not the position of a λ-abstraction. A* multipat *is a vector $\boldsymbol{P}$ of pairwise disjoint pats in t.*

Example 3. *For examples of patterns see [9, Example 11.2.19]. The lhs of* beta *is a positional pattern. It would not be so anymore when swapping its initial binders from $\lambda\, FS$ into $\lambda\, SF$ (pre-order violated). The lhs of* eta *is a pattern, but is not positional (full-extendedness violated).*

For π the lhs of beta*, we have $\{11, 111, 1111, 1112, 11121, 11122\}$ is a pat; $11, 111, 1111$ are the positions from its root 11 toward the head symbol* app*, 11121 the position of* abs*, and 11122 that of $\lambda\, x$. This is the* greatest *pat in π, its* internal *pat $\mathring{\pi}$. The only other pat in π is $\{1112, 11121, 11122\}$ corresponding to* lam$\lambda\, x.F(x)$. *For instance, $\{1112, 11121\}$ is not a pat, since the subterm $\lambda\, x.F(x)$ at position 11122 is not of base type violating (base-fringe), and $\{112\}$ is not a pat since (rigid) is violated by S being a free variable. For TRSs, a pat coincides with a non-empty convex set of function symbol positions as in [2].*

Multipatterns and multipats can be ordered by *refinement* $\sqsubseteq$. These orders correspond and will allow us to state the notion of critical peak in lattice-theoretic terms [2].

Definition 4. $(\lambda\, \boldsymbol{G}.(\overline{(\lambda\, \boldsymbol{F}.s)\, \boldsymbol{u}}))\,\boldsymbol{\pi} \sqsubseteq \overline{(\lambda\, \boldsymbol{G}.((\lambda\, \boldsymbol{F}.s)\, \boldsymbol{u}))\,\boldsymbol{\pi}}$ *if both sides are multipatterns and s, $\boldsymbol{u}$ are linear in $\boldsymbol{F}$, $\boldsymbol{G}$. For multipats, $\boldsymbol{Q} \sqsubseteq \boldsymbol{P}$ if each pat $Q \in \boldsymbol{Q}$ is a subset of a pat $P \in \boldsymbol{P}$.*

[1] We employ $t_{|p}$ / $t(p)$ to denote the subterm / symbol *at* position p in t ([4, p. 5] uses t/p for the former).

Example 4. *We have* $\{\{2,21\},\{222,2221\}\} \sqsubseteq \{\{2,21,22,221,222,2221\}\}$ *for multipats in* $f(g(h(i(a))))$*. Likewise* $(\lambda XY.f(X(h(Y(a))))\,(\lambda z.g(z))\,(\lambda z.i(z)) \sqsubseteq (\lambda Z.f(Z(a)))\,\lambda z.g(h(i(z)))$ *for multipatterns as witnessed by* $(\lambda XY.(\lambda Z.f(Z(a)))\,(\lambda z.X(h(Y(z)))))\,(\lambda z.g(z))\,(\lambda z.i(z))$.

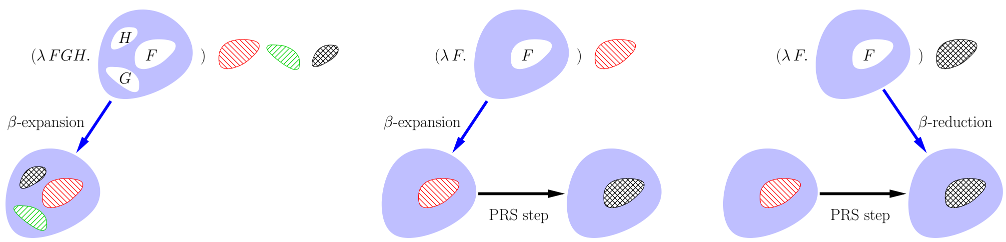

Figure 1: Carving out multipat from term by β-expanding into multipattern (left), and step for PRS rule $\ell \to r$ via matching (β-expansion; middle) and substitution (β-reduction; right)

Lemma 1. *Refinement* $\sqsubseteq$ *on multipats / multipatterns of a* λ*-term is a finite distributive lattice. Multipatterns and multipats w.r.t. their respective notions of refinement* $\sqsubseteq$*, are isomorphic.*

Proof idea. By extending the proof of-[2, Lemma 2] to positional PRSs. The isomorphism between multipats and multipatterns is illustrated in Figure 1; for any multipat $\boldsymbol{P}$ in a λ-term t a multipattern $\boldsymbol{\pi}$ may be *carved out* from t in that $(\lambda \boldsymbol{F}.s)\,\boldsymbol{\pi} = t$ for some s linear in $\boldsymbol{F}$ such that the set of *internal* positions of the $\boldsymbol{\pi}$ in it *trace* [9] to the $\boldsymbol{P}$ in t, and vice versa. $\square$

Steps The *steps* of a PRS are *terms* over the signature extended with rules [9, Chapter 8].

Definition 5. *A* multistep *of a PRS* $\mathcal{P}$ *is a term over its signature extended with its rule symbols. This induces a rewrite system* $\multimap_{\mathcal{P}}$ *having terms as objects, multisteps as steps, with source / target maps obtained substituting the lhs / rhs for the rule symbol [8, 6]; cf. Figure 1 (middle,right). Requiring to have* one *rule in a multistep yields* steps $\to_{\mathcal{P}}$.

Example 5. $\mathsf{abs}(\lambda y.\mathsf{beta}(\lambda x.\mathsf{app}(x,x),y)))$ *and* $\mathsf{eta}(\mathsf{abs}(\lambda x.\mathsf{app}(x,x)))$ *are* Lam*-steps. Despite being intensionally distinct, they are extensionally the same as they have the same sources* $\mathsf{abs}(\lambda y.(\lambda FS.\mathsf{app}(\mathsf{lam}\lambda x.F(x),S))\,(\lambda x.\mathsf{app}(x,x),y))) = \mathsf{abs}(\lambda y.\mathsf{app}(\mathsf{abs}(\lambda x.\mathsf{app}(x,x)),y)) = (\lambda S.\mathsf{lam}(\lambda x.\mathsf{app}(S,x)))\,(\mathsf{abs}(\lambda x.\mathsf{app}(x,x)))$ *and targets* $\mathsf{abs}(\lambda y.(\lambda FS.F(S))\,(\lambda x.\mathsf{app}(x,x),y))) = \mathsf{abs}(\lambda y.\mathsf{app}(y,y)) = (\lambda S.S)\,(\mathsf{abs}(\lambda x.\mathsf{app}(x,x)))$.

Multisteps render traditional redex-orthogonality-talk obsolete [2]; redexes are orthogonal *because* there is a multistep contracting them. Note $\to_{\mathcal{P}} \subseteq \multimap_{\mathcal{P}} \subseteq \twoheadrightarrow_{\mathcal{P}}$ [9, Lemma 11.6.24(ii)].

The FMC as fragment of a PRS The untyped λ-calculus is embedded in a *fragment* of the 2nd-order PRS $\mathcal{Lam}$, namely in terms where all variables are of type term. We show the same holds for the FMC: its terms are embedded as a fragment of a 3rd-order PRS $\mathcal{FMC}$. The embedding hinges on that although the FMC (Definition 1) has a non-standard notion of substitution, that may be *represented* by PRS substitution by replacing each $\star$ by a variable χ, so that *composition* with N in the FMC is represented in $\mathcal{FMC}$ as substitution of N for χ.

Definition 6. *The PRS* $\mathcal{FMC}$ *has a signature comprising for every location* a*, symbols* $\mathsf{lam}_a : ((\mathsf{term}\to\mathsf{term})\to\mathsf{term})\to\mathsf{term}$ *and* $\mathsf{app}_a : \mathsf{term}\to(\mathsf{term}\to\mathsf{term})\to\mathsf{term}$*, and rewrite rule schema:*

$$\mathsf{beta}_H \quad : \quad \lambda M\boldsymbol{P}N.\mathsf{app}_a(H[\mathsf{lam}_a(\lambda x.M(\boldsymbol{x},x))],N) \quad \to \quad \lambda M\boldsymbol{P}N.H[M(\boldsymbol{x},N)]$$

where N, $\boldsymbol{x}$, and x all have type $\mathsf{term} \to \mathsf{term}$ *(not η-expanded to avoid clutter) and H ranges over* contexts*, compositions of basic contexts with the empty context $\Box$, with a* basic *context being of shape either* $\mathsf{app}_b(\Box, P(\boldsymbol{x}))$ *or* $\mathsf{lam}_b(\lambda\, x.\Box)$*, for any location b distinct from a, and each $P \in \boldsymbol{P}$ a fresh free variable having as parameters the variables bound by the contexts above it.*

Terms of the FMC are represented by *spines*, $\mathcal{FMC}$-terms $\lambda\,\chi.S$ of type $\mathsf{term} \to \mathsf{term}$ with:

$$S \quad ::= \quad \chi \mid x\,S \mid \mathsf{app}_a(S, \lambda\,\chi.S) \mid \mathsf{lam}_a(\lambda\,x.S)$$

where χ is the *unique* variable of type term. We embed an FMC term M as $\lambda\,\chi.\langle M\rangle$ and show this fragment of $\mathcal{FMC}$ is well-behaved, where $\langle\,\rangle$ maps the FMC constructs as follows: (i) $\star$ is mapped by $\langle\,\rangle$ to χ, that is, to the coccyx of a spine; (ii) $x.\,M$ is mapped to $x\langle M\rangle$, that is, to the application of x to the embedding of M; (iii) $[N]a.\,M$ is mapped to $\mathsf{app}_a(\langle M\rangle, \lambda\,\chi.\langle N\rangle)$; and (iv) $a\langle x\rangle.\,M$ is mapped to $\mathsf{lam}_a(\lambda\,x.\langle M\rangle)$.

Lemma 2. *Embedding the FMC in the $\lambda\,\chi.S$-fragment yields a bisimulation for $\rightarrow$ and $\rightarrow_{\mathsf{beta}_H}$.*

3 A Multi–One Critical Pair Criterion for the FMC

We generalise the critical pair criterion for confluence introduced in [5] from left-linear TRSs to positional PRSs to obtain confluence of $\mathcal{FMC}$, and hence (Lemma 2) of its $\lambda\,\chi.S$-fragment.

Definition 7. *Multipatterns ς and ζ in term t are* overlapping *if $\varsigma \sqcap \zeta \neq \bot$, where $\sqcap$ denotes the meet w.r.t. refinement $\sqsubseteq$ and $\bot$ the least element (t). The overlap is* critical *if moreover $\varsigma \sqcup \zeta = (\lambda\,F.\hat{F})\,t$ with $\hat{F}$ the η-expansion of F. This extends to peaks ${}_{\Phi}{\leftarrow\!\!\circ}\; t \rightarrow_{\Psi}$ of multisteps $\Phi = (\lambda\,\boldsymbol{F}.s)\,\boldsymbol{\varrho}$ and $\Psi = (\lambda\,\boldsymbol{G}.u)\,\boldsymbol{\theta}$ for rules $\boldsymbol{\varrho} : \boldsymbol{\ell} \to \boldsymbol{r}$ and $\boldsymbol{\theta} : \boldsymbol{g} \to \boldsymbol{d}$, via their multipatterns $(\lambda\,\boldsymbol{F}.s)\,\boldsymbol{\ell}$ and $(\lambda\,\boldsymbol{G}.u)\,\boldsymbol{g}$. If Ψ is a step, we speak of a multi–one (critical) peak.*

Example 6. *We give two multi–one critical peaks for the following TRS [5, Example 1], with our multi–one critical peaks corresponding to the critical pairs numbered (4) and (5) there:*

$$\begin{aligned} \alpha &: \lambda\,xyz.x + (y+z) \to \lambda\,xyz.(x+y)+z \\ \gamma &: \qquad\quad \lambda\,xy.x+y \to \lambda\,xy.y+x \end{aligned}$$

$$\lambda\,xyz.(z+y)+x \;{}_{(\lambda\,FGxyz.F(x,G(y,z)))\,\gamma\gamma}{\leftarrow\!\!\circ}\; \lambda\,xyz.x+(y+z) \rightarrow_{(\lambda\,Hxyz.H(x,y,z))\,\alpha} \lambda\,xyz.(x+y)+z$$

$$\lambda\,\boldsymbol{w}.((x+y)+z)+w \;{}_{(\lambda\,FG\boldsymbol{w}.F(w,G(x,y,z)))\,\gamma\alpha}{\leftarrow\!\!\circ}\; \lambda\,\boldsymbol{w}.w+(x+(y+z)) \rightarrow_{(\lambda\,H\boldsymbol{w}.H(w,x,y+z)\,\alpha} \lambda\,\boldsymbol{w}.(w+x)+(y+z)$$

where $\boldsymbol{x} = wxyz$. The first multi–one peak has $\{111\cdot\{\varepsilon,1,11\}, 111\cdot\{2,21,211\}\}$ as multipat for the left multistep and $\{111\cdot\{\varepsilon,1,11,2,21,211\}\}$ for the right.

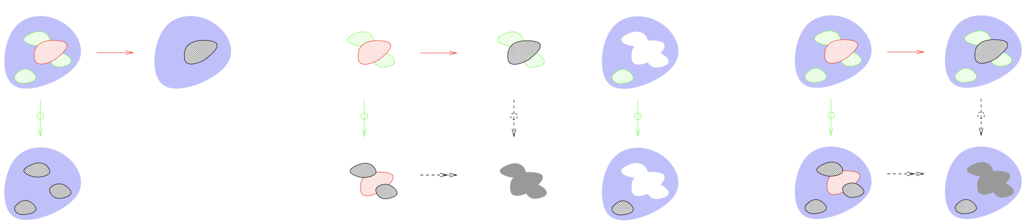

Figure 2: Illustration of proof of Lemma 3 by *splitting-off* critical multi–one peak

.

Lemma 3. *If for a positional PRS $\mathscr{P}$ every critical multi–one peak is many–multi joinable, i.e. if ${}_{\Phi}\!\leftarrow\!\!\circ\!\!- \cdot \to_{\Psi} \subseteq \twoheadrightarrow_{\mathscr{P}} \cdot {}_{\mathscr{P}}\!\leftarrow\!\!\circ\!\!-$ for Φ, Ψ critical, then multi–one peaks are many–multi joinable.*

Proof idea. Let $s \;{}_{\Phi}\!\leftarrow\!\!\circ\!\!-\; t \to_{\Psi} u$ be a multi–one peak. The geometric view, justified by Lemma 1, for the following construction is illustrated in Figure 2 where the blue blob denotes t, the green blobs the multipat of the multistep $-\!\!\circ\!\!\to_{\Phi}$, and the red blob that of the step $\to_{\Psi}$.

We may write the multipattern ς of Φ as $(\lambda\, \boldsymbol{G'G}.s')\,\boldsymbol{\ell'\ell}$, and the multipattern ζ of Ψ as $(\lambda\, F.u')\,g$, with $\boldsymbol{\ell}$ those patterns in ς overlapping the pattern g (2 green blobs in the figure overlapping the red one), and $\boldsymbol{\ell'}$ (1 green blob) the non-overlapping ones;

The join $\varsigma \sqcup \zeta$ is then of shape $(\lambda\, \boldsymbol{G'}F'.v)\,\boldsymbol{\ell'}\pi$ with π being the minimal pattern refinable into both $\boldsymbol{\ell}$ and g. Thus, $\varsigma = (\lambda\, \boldsymbol{G'G}.\overline{(\lambda\, \boldsymbol{G'}F'.v)\,\boldsymbol{G'}s''})\,\boldsymbol{\ell'\ell}$ and $\zeta = (\lambda\, F.\overline{(\lambda\, \boldsymbol{G'}F'.v)\,\boldsymbol{\ell'}u''})\,g$ for some s'' and u'', with the multisteps Φ and Ψ obtained by replacing the left-hand sides $\boldsymbol{\ell'\ell}$ in ς and g in ζ by rule symbols, and with $\overline{(\lambda\, \boldsymbol{G'G}.(\lambda\, \boldsymbol{G'}F'.v)\,\boldsymbol{G'}s'')\,\boldsymbol{\ell'\ell}} = \varsigma \sqcup \zeta = \overline{(\lambda\, F.(\lambda\, \boldsymbol{G'}F'.v)\,\boldsymbol{\ell'}u'')\,g}$. By minimality, π is the source of the *critical* multi-one peak for multistep $\hat{\Phi}$ and step $\hat{\Psi}$ having multipatterns $(\lambda\, \boldsymbol{G}.s'')\,\boldsymbol{\ell}$ and $(\lambda\, F.u'')\,g$, which is many–multi joinable by assumption, say by valley $\twoheadrightarrow_{\hat{\Psi}'} \cdot {}_{\hat{\Phi}'}\!\leftarrow\!\!\circ\!\!-$ for reduction $\hat{\Psi}'$ and multistep $\hat{\Phi}'$. We conclude by *plugging these into context* as in Figure 2 (right), yielding the reduction $(\lambda\, \boldsymbol{G'}F'.v)\,\boldsymbol{r'}\hat{\Psi}'$ and the multistep $(\lambda\, \boldsymbol{G'}F'.v)\,\boldsymbol{\varrho'}\hat{\Phi}'$, for $\boldsymbol{r'}$ and $\boldsymbol{\varrho'}$ the right-hand sides respectively the rules, corresponding to $\boldsymbol{\ell'}$. $\square$

Theorem 1. *A positional PRS is confluent if multi–one critical peaks are many–multi joinable.*

Proof. By Lemma 3 using $\to_{\mathscr{P}} \subseteq -\!\!\circ\!\!\to_{\mathscr{P}} \subseteq \twoheadrightarrow_{\mathscr{P}}$ for any positional PRS $\mathscr{P}$. $\square$

Theorem 2. *FMC reduction is confluent.*

Proof. By Theorem 1 and Lemma 2 it suffices that all multi–one critical peaks of $\mathscr{FMC}$ are many–multi joinable. There are still infinitely many such peaks, but these are uniformly shown to be many–multi joinable: since in the FMC all patterns in a critical peak are on the same spine, and patterns on the spine are not replicated, the peaks are even one–multi joinable. $\square$

References

[1] C. Barrett, W. Heijltjes, and G. McCusker. The functional machine calculus, 2022. To appear in Mathematical Foundations of Programming Semantics (MFPS 2022). Available at http://people.bath.ac.uk/wbh22/index.html#FMC2022.

[2] N. Hirokawa, J. Nagele, V. van Oostrom, and M. Oyamaguchi. Confluence by critical pair analysis revisited. In *CADE 27*, volume 11716 of *LNCS*, pages 319–336. Springer, 2019.

[3] P.B. Levy. *Call-by-push-value: A functional/imperative synthesis*, volume 2 of *Semantic Structures in Computation*. Springer Netherlands, 2003.

[4] R. Mayr and T. Nipkow. Higher-order rewrite systems and their confluence. *TCS*, 192(1):3–29, 1998.

[5] S. Okui. Simultaneous critical pairs and Church–Rosser property. In T. Nipkow, editor, *RTA-98*, volume 1379 of *LNCS*, pages 2–16. Springer, 1998.

[6] V. van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, March 1994.

[7] V. van Oostrom. Finite family developments. In H. Comon, editor, *RTA-97*, volume 1232 of *LNCS*, pages 308–322. Springer, 1997.

[8] V. van Oostrom and F. van Raamsdonk. Weak orthogonality implies confluence: The higher order case. In *LFCS'94*, volume 813 of *LNCS*, pages 379–392. Springer, 1994.

[9] Terese. *Term Rewriting Systems*, volume 55 of *CTTCS*. CUP, 2003.