

Combinatorics on Words formalized  
Binary codes that do not preserve primitivity

Štěpán Holub  
Martin Raška

May 26, 2024

Funded by the Czech Science Foundation grant GAČR 20-20621S.

# Contents

0.1	Lemmas for covered x square . . . . .	2
0.1.1	Two particular cases . . . . .	2
0.1.2	Main cases . . . . .	2
0.2	Square interpretation . . . . .	3
0.2.1	Locale: interpretation . . . . .	4
0.2.2	Locale with additional parameters . . . . .	5
0.2.3	Back to the main locale . . . . .	6
0.2.4	Locale: Extendable interpretation . . . . .	7
0.3	General primitivity not preserving codes . . . . .	8
0.4	Covered uniform square . . . . .	9
0.4.1	Primitivity (non)preserving uniform binary codes . . . . .	10
0.5	The main theorem . . . . .	11
0.5.1	Imprimitive words with single y . . . . .	11
0.5.2	Conjugate words . . . . .	11
0.5.3	Square factor of the longer word and both words primitive (was all <sub>assms</sub> ) . . . . .	11
0.5.4	Obtaining primitivity with two squares (refining) . . . . .	11
0.5.5	Obtaining the square of the longer word (gluing) . . . . .	12
0.6	Examples . . . . .	12
0.7	Primitivity non-preserving binary code . . . . .	13
0.7.1	The target theorem . . . . .	13
0.8	Upper bound of the power exponent in the canonical imprimitivity witness . . . . .	14
0.8.1	Optimality of the exponent upper bound . . . . .	15
0.9	Characterization of binary primitivity preserving morphisms given by a pair of words . . . . .	15
0.9.1	Code equation for <i>bin-prim</i> predicate . . . . .	16
0.10	Characterization of binary imprimitivity codes . . . . .	17
	<b>References</b>	<b>18</b>

**theory** *Binary-Square-Interpretation*

**imports**

*Combinatorics-Words.Submonoids*

*Combinatorics-Words.Equations-Basic*

**begin**

## 0.1 Lemmas for covered x square

This section explores various variants of the situation when  $x \cdot x$  is covered with  $x \cdot y^{\textcircled{a}} k \cdot u \cdot v \cdot y^{\textcircled{a}} l \cdot x$ , with  $y = u \cdot v$ , and the displayed dots being synchronized.

### 0.1.1 Two particular cases

**lemma** *pref-suf-pers-short*: **assumes**  $x \leq_p v \cdot x$  **and**  $|v \cdot u| < |x|$  **and**  $x \leq_s r \cdot u \cdot v \cdot u$  **and**  $r \in \langle \{u, v\} \rangle$

—  $x \cdot x$  is covered by  $(p \cdot u \cdot v \cdot u) \cdot v \cdot x$ , the displayed dots being synchronized

— That is, the condition on the first  $x$  in  $x \cdot y^{\textcircled{a}} k \cdot u \cdot v \cdot y^{\textcircled{a}} l \cdot x$  is relaxed

**shows**  $u \cdot v = v \cdot u$

*<proof>*

**lemma** *pref-suf-pers-large-overlap*:

**assumes**

$p \leq_p x$  **and**  $s \leq_s x$  **and**  $p \leq_p r \cdot p$  **and**  $s \leq_s s \cdot r$  **and**  $|x| + |r| \leq |p| + |s|$

**shows**  $x \cdot r = r \cdot x$

*<proof>*

### 0.1.2 Main cases

**locale** *pref-suf-pers* =

**fixes**  $x u v k m$

**assumes**

*x-pref*:  $x \leq_p (v \cdot (u \cdot v)^{\textcircled{a}} k) \cdot x$  —  $\leq_p x (p \cdot x)$  **and**  $\leq_p p (q \cdot p)$  where  $q = v \cdot u$

**and**

*x-suf*:  $x \leq_s x \cdot (u \cdot v)^{\textcircled{a}} m \cdot u$  —  $\leq_s x (s \cdot x)$  **and**  $\leq_s s (q' \cdot s)$  where  $q' = u \cdot v$

**and** *k-pos*:  $0 < k$  **and** *m-pos*:  $0 < m$

**begin**

**lemma** *pref-suf-commute-all-commutes*:

**assumes**  $|u \cdot v| \leq |x|$  **and**  $u \cdot v = v \cdot u$

**shows** *commutes*  $\{u, v, x\}$

*<proof>*

**lemma** *no-overlap*:

**assumes**

*len*:  $|v \cdot (u \cdot v)^{\textcircled{a}} k| + |(u \cdot v)^{\textcircled{a}} m \cdot u| \leq |x|$  (**is**  $|?p| + |?s| \leq |x|$ ) **and**

$0 < k$   $0 < m$

**shows** *commutes*  $\{u, v, x\}$

*<proof>*

**lemma** *no-overlap'*:

**assumes**

*len*:  $|v \cdot (u \cdot v)^{\textcircled{k}}| + |(u \cdot v)^{\textcircled{m}} \cdot u| \leq |x|$  (**is**  $|?p| + |?s| \leq |x|$ )

**and**  $0 < k \ 0 < m$

**shows**  $u \cdot v = v \cdot u$

*<proof>*

**lemma** *short-overlap*:

**assumes**

*len1*:  $|x| < |v \cdot (u \cdot v)^{\textcircled{k}}| + |(u \cdot v)^{\textcircled{m}} \cdot u|$  (**is**  $|x| < |?p| + |?s|$ ) **and**

*len2*:  $|v \cdot (u \cdot v)^{\textcircled{k}}| + |(u \cdot v)^{\textcircled{m}} \cdot u| \leq |x| + |u|$  (**is**  $|?p| + |?s| \leq |x| + |u|$ )

**shows** *commutes*  $\{u, v, x\}$

*<proof>*

**lemma** *medium-overlap*:

**assumes**

*len1*:  $|x| + |u| < |v \cdot (u \cdot v)^{\textcircled{k}}| + |(u \cdot v)^{\textcircled{m}} \cdot u|$  (**is**  $|x| + |u| < |?p| + |?s|$ )

**and**

*len2*:  $|v \cdot (u \cdot v)^{\textcircled{k}}| + |(u \cdot v)^{\textcircled{m}} \cdot u| < |x| + |u \cdot v|$  (**is**  $|?p| + |?s| < |x| + |u \cdot v|$ )

**shows** *commutes*  $\{u, v, x\}$

*<proof>*

**thm**

*no-overlap*

*short-overlap*

*medium-overlap*

**end**

**thm**

*pref-suf-pers.no-overlap*

*pref-suf-pers.short-overlap*

*pref-suf-pers.medium-overlap*

*pref-suf-pers.large-overlap*

## 0.2 Square interpretation

In this section fundamental description is given of (the only) possible  $\{x, y\}$ -interpretation of the square  $x \cdot x$ , where  $|y| \leq |x|$ . The proof is divided into several locales.

**lemma** *cover-not-disjoint*:

**shows** *primitive*  $(\mathbf{a \cdot b \cdot a \cdot b \cdot a \cdot b \cdot a})$  (**is** *primitive*  $?x$ ) **and**

*primitive*  $(\mathbf{a \cdot b})$  (**is** *primitive*  $?y$ ) **and**

$(\mathbf{a \cdot b \cdot a \cdot b \cdot a \cdot b \cdot a}) \cdot (\mathbf{a \cdot b}) \neq (\mathbf{a \cdot b}) \cdot (\mathbf{a \cdot b \cdot a \cdot b \cdot a \cdot b \cdot a})$

(**is**  $?x \cdot ?y \neq ?y \cdot ?x$ ) **and**

$\varepsilon (\mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a}) \cdot (\mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a}) (\mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a}) \sim_{\mathcal{I}} [(\mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a}), (\mathbf{a} \cdot \mathbf{b}), (\mathbf{a} \cdot \mathbf{b}), (\mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a} \cdot \mathbf{b} \cdot \mathbf{a})]$   
 (is  $\varepsilon ?x \cdot ?x ?s \sim_{\mathcal{I}} [?x, ?y, ?y, ?x]$ )  
 ⟨proof⟩

## 0.2.1 Locale: interpretation

**locale** *square-interp* =

— The basic set of assumptions  
 — The goal is to arrive at  $ws = [x] \cdot [y] \textcircled{a} k \cdot [x]$  including the description of the interpretation in terms of the first and the second occurrence of x in the interpreted square.

**fixes**  $x y p s ws$

**assumes**

*non-comm*:  $x \cdot y \neq y \cdot x$  **and**

*prim-x*: *primitive*  $x$  **and**

*y-le-x*:  $|y| \leq |x|$  **and**

*ws-lists*:  $ws \in \text{lists } \{x, y\}$  **and**

*nconjug*:  $\neg x \sim y$  **and**

*disj-interp*:  $p [x, x] s \sim_{\mathcal{D}} ws$

**begin**

**lemma** *interp*:  $p (x \cdot x) s \sim_{\mathcal{I}} ws$

⟨proof⟩

**lemma** *disjoint*:  $p1 \leq_p [x, x] \implies p2 \leq_p ws \implies p \cdot \text{concat } p1 \neq \text{concat } p2$

⟨proof⟩

**interpretation** *binary-code*  $x y$

⟨proof⟩

**lemmas** *interpret-concat* = *fac-interpD*(3)[*OF interp*]

**lemma** *p-nemp*:  $p \neq \varepsilon$

⟨proof⟩

**lemma** *s-nemp*:  $s \neq \varepsilon$

⟨proof⟩

**lemma** *x-root*:  $\rho x = x$

⟨proof⟩

**lemma** *ws-nemp*:  $ws \neq \varepsilon$

⟨proof⟩

**lemma** *hd-ws-lists*:  $\text{hd } ws \in \{x, y\}$

⟨proof⟩

**lemma** *last-ws-lists*:  $last\ ws \in \{x, y\}$   
*<proof>*

**lemma** *kE*: **obtains**  $k$  **where**  $[hd\ ws] \cdot [y]^{\textcircled{a}} k \cdot [last\ ws] = ws$   
*<proof>*

**lemma** *l-mE*: **obtains**  $m\ u\ v\ l$  **where**  $(hd\ ws) \cdot y^{\textcircled{a}} m \cdot u = p \cdot x$  **and**  $v \cdot y^{\textcircled{a}} l \cdot (last\ ws) = x \cdot s$  **and**  
 $u \cdot v = y\ u \neq \varepsilon\ v \neq \varepsilon$  **and**  $x \cdot (v \cdot u) \neq (v \cdot u) \cdot x$   
*<proof>*

**lemma** *last-ws*:  $last\ ws = x$   
*<proof>*

**lemma** *rev-square-interp*:  
*square-interp* (*rev*  $x$ ) (*rev*  $y$ ) (*rev*  $s$ ) (*rev*  $p$ ) (*rev* (*map* *rev*  $ws$ ))  
*<proof>*

**lemma** *hd-ws*:  $hd\ ws = x$   
*<proof>*

**lemma** *p-pref*:  $p <_p x$   
*<proof>*

**lemma** *s-suf*:  $s <_s x$   
*<proof>*

**end**

## 0.2.2 Locale with additional parameters

**locale** *square-interp-plus* = *square-interp* +  
**fixes**  $l\ m\ u\ v$   
**assumes** *fst-x*:  $x \cdot y^{\textcircled{a}} m \cdot u = p \cdot x$  **and**  
*snd-x*:  $v \cdot y^{\textcircled{a}} l \cdot x = x \cdot s$  **and**  
*uv-y*:  $u \cdot v = y$  **and**  
*u-nemp*:  $u \neq \varepsilon$  **and** *v-nemp*:  $v \neq \varepsilon$  **and**  
*vu-x-non-comm*:  $x \cdot (v \cdot u) \neq (v \cdot u) \cdot x$   
**begin**

**interpretation** *binary-code*  $x\ y$   
*<proof>*

**lemma** *rev-square-interp-plus*: *square-interp-plus* (*rev*  $x$ ) (*rev*  $y$ ) (*rev*  $s$ ) (*rev*  $p$ )  
(*rev* (*map* *rev*  $ws$ ))  $m\ l$  (*rev*  $v$ ) (*rev*  $u$ )  
*<proof>*

**Exactly one of the exponents is zero: impossible.**

Uses lemma  $\llbracket \leq_p ?x (?v \cdot ?x); |?v \cdot ?u| < |?x|; \leq_s ?x (?r \cdot ?u \cdot ?v \cdot ?u); ?r \in \langle \{?u, ?v\} \rangle \rrbracket \implies ?u \cdot ?v = ?v \cdot ?u$  and exploits the symmetric interpretation.

**lemma *fst-exp-zero*: assumes  $m = 0$  and  $0 < l$  shows *False***  
 $\langle proof \rangle$

**lemma *snd-exp-zero*: assumes  $0 < m$  and  $l = 0$  shows *False***  
 $\langle proof \rangle$

**Both exponents positive: impossible**

**lemma *both-exps-pos*: assumes  $0 < m$  and  $0 < l$  shows *False***  
 $\langle proof \rangle$

**thm *suf-cancel-conv***

**end**

### 0.2.3 Back to the main locale

**context *square-interp***

**begin**

**definition *u where*  $u = x^{-1} \langle p \cdot x \rangle$**

**definition *v where*  $v = (x \cdot s)^{\langle -1 \rangle} x$**

**lemma *cover-xyx*:  $ws = [x, y, x]$  and *vu-x-non-comm*:  $x \cdot (v \cdot u) \neq (v \cdot u) \cdot x$  and *uv-y*:  $u \cdot v = y$  and**

***px-xu*:  $p \cdot x = x \cdot u$  and *vx-xs*:  $v \cdot x = x \cdot s$  and *u-nemp*:  $u \neq \varepsilon$  and *v-nemp*:  $v \neq \varepsilon$**   
 $\langle proof \rangle$

**lemma *cover*:  $x \cdot y \cdot x = p \cdot x \cdot x \cdot s$**   
 $\langle proof \rangle$

**lemma *conjug-facs*:  $\varrho u \sim \varrho v$**   
 $\langle proof \rangle$

**term *square-interp.v***

— We have a detailed information about all words

**lemma *bin-sq-interpE*: obtains  $r \ t \ m \ k \ l$**

**where  $(t \cdot r)^{\textcircled{k}} = u$  and  $(r \cdot t)^{\textcircled{l}} = v$  and**

**$(r \cdot t)^{\textcircled{m}} \cdot r = x$  and  $(t \cdot r)^{\textcircled{k}} \cdot (r \cdot t)^{\textcircled{l}} = y$**

**and  $(r \cdot t)^{\textcircled{k}} = p$  and  $(t \cdot r)^{\textcircled{l}} = s$  and  $r \cdot t \neq t \cdot r$  and**

**$0 < k$  and  $0 < m$  and  $0 < l$**

$\langle proof \rangle$

end

## 0.2.4 Locale: Extendable interpretation

Further specification follows from the assumption that the interpretation is extendable, that is, the covered  $x \cdot x$  is a factor of a word composed of  $\{x, y\}$ . Namely,  $u$  and  $v$  are then conjugate by  $x$ .

**locale** *square-interp-ext* = *square-interp* +  
**assumes** *p-extend*:  $\exists pe. pe \in \langle \{x, y\} \rangle \wedge p \leq_s pe$  **and**  
*s-extend*:  $\exists se. se \in \langle \{x, y\} \rangle \wedge s \leq_p se$

**begin**

**lemma** *s-pref-y*:  $s \leq_p y$   
*<proof>*

**lemma** *rev-square-interp-ext*: *square-interp-ext* (*rev x*) (*rev y*) (*rev s*) (*rev p*) (*rev*  
(*map rev ws*))  
*<proof>*

**lemma** *p-suf-y*:  $p \leq_s y$   
*<proof>*

**theorem** *bin-sq-interp-extE*: **obtains**  $r t k m$  **where**  $(r \cdot t)^{\textcircled{m}} \cdot r = x$  **and**  $(t \cdot$   
 $r)^{\textcircled{k}} \cdot (r \cdot t)^{\textcircled{k}} = y$   
 $(r \cdot t)^{\textcircled{k}} = p$  **and**  $(t \cdot r)^{\textcircled{k}} = s$  **and**  $r \cdot t \neq t \cdot r$  **and**  $u = s$  **and**  $v = p$  **and**  
 $|p| = |s|$  **and**  
 $0 < k$  **and**  $0 < m$   
*<proof>*

**lemma** *ps-len*:  $|p| = |s|$  **and** *p-eq-v*:  $p = v$  **and** *s-eq-u*:  $s = u$   
*<proof>*

**lemma** *v-x-x-u*:  $v \cdot x = x \cdot u$   
*<proof>*

**lemma** *sp-y*:  $s \cdot p = y$   
*<proof>*

**lemma** *p-x-x-s*:  $p \cdot x = x \cdot s$   
*<proof>*

**lemma** *xy-root*:  $x \cdot x \cdot y = (x \cdot p) \cdot (x \cdot p)$   
*<proof>*

**theorem** *sq-ext-interp*:  $ws = [x, y, x] s \cdot p = y p \cdot x = x \cdot s$   
*<proof>*



**end**

**theorem** *bin-sq-interpE*:

**assumes**  $x \cdot y \neq y \cdot x$  **and** *primitive*  $x$  **and**  $|y| \leq |x|$  **and**  $ws \in \text{lists } \{x, y\}$  **and**  
 $\neg x \sim y$  **and**  
 $p [x,x] s \sim_{\mathcal{D}} ws$   
**obtains**  $r t m k l$  **where**  $(r \cdot t)^{\textcircled{a}} m \cdot r = x$  **and**  $(t \cdot r)^{\textcircled{a}} k \cdot (r \cdot t)^{\textcircled{a}} l = y$   
 $(r \cdot t)^{\textcircled{a}} k = p$  **and**  $(t \cdot r)^{\textcircled{a}} l = s$  **and**  $r \cdot t \neq t \cdot r$  **and**  $0 < k \ 0 < m \ 0 < l$   
*<proof>*

**theorem** *bin-sq-interp*:

**assumes**  $x \cdot y \neq y \cdot x$  **and** *primitive*  $x$  **and**  $|y| \leq |x|$  **and**  $ws \in \text{lists } \{x, y\}$  **and**  
 $\neg x \sim y$  **and**  
 $p [x,x] s \sim_{\mathcal{D}} ws$   
**shows**  $ws = [x,y,x]$   
*<proof>*

**theorem** *bin-sq-interp-extE*:

**assumes**  $x \cdot y \neq y \cdot x$  **and** *primitive*  $x$  **and**  $|y| \leq |x|$  **and**  $ws \in \text{lists } \{x, y\}$  **and**  
 $\neg x \sim y$  **and**  
 $p [x,x] s \sim_{\mathcal{D}} ws$  **and**  
*p-extend*:  $\exists pe. pe \in \langle \{x,y\} \rangle \wedge p \leq s pe$  **and**  
*s-extend*:  $\exists se. se \in \langle \{x,y\} \rangle \wedge s \leq p se$   
**obtains**  $r t m k$  **where**  $(r \cdot t)^{\textcircled{a}} m \cdot r = x$  **and**  $(t \cdot r)^{\textcircled{a}} k \cdot (r \cdot t)^{\textcircled{a}} k = y$   
 $(r \cdot t)^{\textcircled{a}} k = p$  **and**  $(t \cdot r)^{\textcircled{a}} k = s$  **and**  $r \cdot t \neq t \cdot r$  **and**  $0 < k$  **and**  $0 < m$   
*<proof>*

**end**

**theory** *Binary-Code-Imprimitive*

**imports**

*Combinatorics-Words-Graph-Lemma.Glued-Codes*

*Binary-Square-Interpretation*

**begin**

This theory focuses on the characterization of imprimitive words which are concatenations of copies of two words (forming a binary code). We follow the article [1] (mainly Théorème 2.1 and Lemme 3.1), while substantially optimizing the proof. See also [3] for an earlier result on this question, and [2] for another proof.

### 0.3 General primitivity not preserving codes

**context** *code*

**begin**

Two nontrivially conjugate elements generated by a code induce a disjoint

interpretation.

**lemma** *shift-disjoint*:

**assumes**  $ws \in \text{lists } \mathcal{C}$  **and**  $ws' \in \text{lists } \mathcal{C}$  **and**  $z \notin \langle \mathcal{C} \rangle$  **and**  $z \cdot \text{concat } ws = \text{concat } ws' \cdot z$   
 $us \leq_p ws^{\textcircled{n}}$  **and**  $vs \leq_p ws'^{\textcircled{n}}$   
**shows**  $z \cdot \text{concat } us \neq \text{concat } vs$   
*<proof>*

This in particular yields a disjoint extendable interpretation of any prefix

**lemma** *shift-interp*:

**assumes**  $ws \in \text{lists } \mathcal{C}$  **and**  $ws' \in \text{lists } \mathcal{C}$  **and**  $z \notin \langle \mathcal{C} \rangle$  **and**  
 $\text{conjug: } z \cdot \text{concat } ws = \text{concat } ws' \cdot z$  **and**  $|z| \leq |\text{concat } ws'|$   
**and**  $us \leq_p ws$  **and**  $us \neq \varepsilon$   
**obtains**  $p \ s \ vs \ ps$  **where**  
 $p \ us \ s \sim_{\mathcal{D}} \ vs$  **and**  $vs \in \text{lists } \mathcal{C}$   
**and**  $s \leq_p \text{concat } (us^{-1} \triangleright (ws \cdot ws))$  **and**  $p \leq_s \text{concat } ws$  — extendable  
**and**  $ps \cdot vs \leq_p ws' \cdot ws'$  **and**  $\text{concat } ps \cdot p = z$   
*<proof>*

The conditions are in particular met by imprimitivity witnesses

**lemma** *imprim-witness-shift*:

**assumes**  $ws \in \text{lists } \mathcal{C}$  **and**  $\text{primitive } ws$  **and**  $\neg \text{primitive } (\text{concat } ws)$   
**obtains**  $z \ n$  **where**  $\text{concat } ws = z^{\textcircled{n}}$   $z \notin \langle \mathcal{C} \rangle$  **and**  
 $z \cdot \text{concat } ws = \text{concat } ws \cdot z$  **and**  $|z| < |\text{concat } ws|$  **and**  $2 \leq n$   
*<proof>*

**end**

## 0.4 Covered uniform square

**lemma** *cover-xy-xxx*: **assumes**  $|x| = |y|$  **and**  $p \cdot x \cdot y \cdot s = x \cdot x \cdot x$   
**shows**  $x = y$   
*<proof>*

**lemma** *cover-xy-yyy*: **assumes**  $|x| = |y|$  **and**  $eq: p \cdot x \cdot y \cdot s = y \cdot y \cdot y$   
**shows**  $x = y$   
*<proof>*

**lemma** *cover-xy-xyx*: **assumes**  $|x| = |y|$  **and**  $s \neq \varepsilon$  **and**  $eq: p \cdot x \cdot y \cdot s = x \cdot x \cdot y$   
**shows**  $x = y$   
*<proof>*

**lemma** *cover-xy-xyy*: **assumes**  $|x| = |y|$  **and**  $p \neq \varepsilon$  **and**  $eq: p \cdot x \cdot y \cdot s = x \cdot y \cdot y$   
**shows**  $x = y$   
*<proof>*

**lemma** *cover-xy-yyx*: **assumes**  $|x| = |y|$  **and**  $eq: p \cdot x \cdot y \cdot s = y \cdot y \cdot x$

**shows**  $x = y$   
*<proof>*

**lemma** *cover-xy-yxx*: **assumes**  $|x| = |y|$  **and**  $eq: p \cdot x \cdot y \cdot s = y \cdot x \cdot x$   
**shows**  $x = y$   
*<proof>*

**lemma** *cover-xy-xyx*: **assumes**  $|x| = |y|$  **and**  $p \neq \varepsilon$  **and**  $s \neq \varepsilon$  **and**  $eq: p \cdot x \cdot y \cdot s = x \cdot y \cdot x$   
**shows**  $\neg primitive(x \cdot y)$   
*<proof>*

**lemma** *cover-xy-yxy*: **assumes**  $|x| = |y|$  **and**  $p \neq \varepsilon$  **and**  $\langle s \neq \varepsilon \rangle$  **and**  $eq: p \cdot x \cdot y \cdot s = y \cdot x \cdot y$   
**shows**  $\neg primitive(x \cdot y)$   
*<proof>*

**theorem** *uniform-square-interp*: **assumes**  $x \cdot y \neq y \cdot x$  **and**  $|x| = |y|$  **and**  $vs \in lists\ \{x,y\}$   
**and**  $p(x \cdot y)\ s \sim_{\mathcal{I}} vs$  **and**  $p \neq \varepsilon$   
**shows**  $\neg primitive(x \cdot y)$  **and**  $vs = [x,y,x] \vee vs = [y,x,y]$   
*<proof>*

#### 0.4.1 Primitivity (non)preserving uniform binary codes

**theorem** *bin-uniform-prim-morph*:  
**assumes**  $x \cdot y \neq y \cdot x$  **and**  $|x| = |y|$  **and**  $primitive(x \cdot y)$   
**and**  $ws \in lists\ \{x,y\}$  **and**  $2 \leq |ws|$   
**shows**  $primitive\ ws \iff primitive(\text{concat}\ ws)$   
*<proof>*

**lemma** *bin-uniform-imprim*: **assumes**  $x \cdot y \neq y \cdot x$  **and**  $|x| = |y|$  **and**  $\neg primitive(x \cdot y)$   
**shows**  $primitive\ x$   
*<proof>*

**theorem** *bin-uniform-prim-morph'*:  
**assumes**  $x \cdot y \neq y \cdot x$  **and**  $|x| = |y|$  **and**  $primitive(x \cdot y) \vee \neg primitive\ x \vee \neg primitive\ y$   
**and**  $ws \in lists\ \{x,y\}$  **and**  $2 \leq |ws|$   
**shows**  $primitive\ ws \iff primitive(\text{concat}\ ws)$   
*<proof>*

## 0.5 The main theorem

### 0.5.1 Imprimitve words with single y

If the shorter word occurs only once, the result is straightforward from the parametric solution of the Lyndon-Schutzenberger equation.

**lemma** *bin-imprim-single-y*:

**assumes** *non-comm*:  $x \cdot y \neq y \cdot x$  **and**

$ws \in \text{lists } \{x, y\}$  **and**

$|y| \leq |x|$  **and**

$2 \leq \text{count-list } ws \ x$  **and**

$\text{count-list } ws \ y < 2$  **and**

*primitive*  $ws$  **and**

$\neg \text{primitive } (\text{concat } ws)$

**shows**  $ws \sim [x, x, y]$  **and** *primitive*  $x$  **and** *primitive*  $y$

*<proof>*

### 0.5.2 Conjugate words

**lemma** *bin-imprim-not-conjug*:

**assumes**  $ws \in \text{lists } \{x, y\}$  **and**

$x \cdot y \neq y \cdot x$  **and**

$2 \leq |ws|$  **and**

*primitive*  $ws$  **and**

$\neg \text{primitive } (\text{concat } ws)$

**shows**  $\neg x \sim y$

*<proof>*

### 0.5.3 Square factor of the longer word and both words primitive (was all\_assms)

The main idea of the proof is as follows: Imprimitivity of the concatenation yields (at least) two overlapping factorizations into  $\{x, y\}$ . Due to the presence of the square  $x \cdot x$ , these two can be synchronized, which yields that the situation coincides with the canonical form.

**lemma** *bin-imprim-primitive*:

**assumes**  $x \cdot y \neq y \cdot x$

**and** *primitive*  $x$  **and** *primitive*  $y$

**and**  $|y| \leq |x|$

**and**  $ws \in \text{lists } \{x, y\}$

**and** *primitive*  $ws$  **and**  $\neg \text{primitive } (\text{concat } ws)$

**and**  $[x, x] \leq_f ws \cdot ws$

**shows**  $ws \sim [x, x, y]$

*<proof>*

### 0.5.4 Obtaining primitivity with two squares (refining)

**lemma** *bin-imprim-both-squares-prim*:

**assumes**  $x \cdot y \neq y \cdot x$   
**and**  $ws \in \text{lists } \{x, y\}$   
**and**  $\text{primitive } ws$  **and**  $\neg \text{primitive } (\text{concat } ws)$   
**and**  $[x, x] \leq_f ws \cdot ws$   
**and**  $[y, y] \leq_f ws \cdot ws$   
**and**  $\text{primitive } x$  **and**  $\text{primitive } y$   
**shows** *False*  
*<proof>*

**lemma** *bin-imprim-both-squares:*

**assumes**  $x \cdot y \neq y \cdot x$   
**and**  $ws \in \text{lists } \{x, y\}$   
**and**  $\text{primitive } ws$  **and**  $\neg \text{primitive } (\text{concat } ws)$   
**and**  $[x, x] \leq_f ws \cdot ws$   
**and**  $[y, y] \leq_f ws \cdot ws$   
**shows** *False*  
*<proof>*

### 0.5.5 Obtaining the square of the longer word (gluing)

**lemma** *bin-imprim-longer-twice:*

— 1. If there are both squares, then contradiction; 2. If a square is missing: a) if  $y$  appears once: the positive conclusion b) if  $y$  appears twice, then gluing preserves presence of the longer word at least twice (because both appear twice) and induction yields  $[x', x', y']$  where  $y'$  is a suffix of  $x'$ , a contradiction with primitivity of words of the form  $xyxy$ ;

**assumes**  $x \cdot y \neq y \cdot x$   
**and**  $ws \in \text{lists } \{x, y\}$   
**and**  $|y| \leq |x|$   
**and**  $\text{count-list } ws \ x \geq 2$   
**and**  $\text{primitive } ws$  **and**  $\neg \text{primitive } (\text{concat } ws)$   
**shows**  $ws \sim [x, x, y] \wedge \text{primitive } x \wedge \text{primitive } y$   
*<proof>*

**lemma** *bin-imprim-both-twice:*

**assumes**  $x \cdot y \neq y \cdot x$   
**and**  $ws \in \text{lists } \{x, y\}$   
**and**  $\text{count-list } ws \ x \geq 2$   
**and**  $\text{count-list } ws \ y \geq 2$   
**and**  $\text{primitive } ws$  **and**  $\neg \text{primitive } (\text{concat } ws)$   
**shows** *False*  
*<proof>*

## 0.6 Examples

**lemma**  $x \neq \varepsilon \implies \varepsilon (x \cdot x) \varepsilon \sim_{\mathcal{I}} [x, x]$   
*<proof>*

**lemma** **assumes**  $x = [(0::\text{nat}), 1, 0, 1, 0]$  **and**  $y = [1, 0, 0, 1]$

**shows**  $[0,1] (x \cdot x) [1,0] \sim_{\mathcal{I}} [x,y,x]$   
 ⟨*proof*⟩

## 0.7 Primitivity non-preserving binary code

In this section, we give the final form of imprimitive words over a given binary code  $\{x, y\}$ . We start with a lemma, then we show that the only possibility is that such word is conjugate with  $x^{\textcircled{a} j} \cdot y^{\textcircled{a} k}$ .

**lemma** *bin-imprim-expsE-y*: **assumes**  $x \cdot y \neq y \cdot x$  **and**

$ws \in \text{lists } \{x,y\}$  **and**

$2 \leq |ws|$  **and**

*primitive*  $ws$  **and**

$\neg$  *primitive* (*concat*  $ws$ ) **and**

*count-list*  $ws$   $y = 1$

**obtains**  $j k$  **where**  $1 \leq j$   $1 \leq k$   $j = 1 \vee k = 1$

$ws \sim [x]^{\textcircled{a} j} \cdot [y]^{\textcircled{a} k}$

⟨*proof*⟩

**lemma** *bin-imprim-expsE*: **assumes**  $x \cdot y \neq y \cdot x$  **and**

$ws \in \text{lists } \{x,y\}$  **and**

$2 \leq |ws|$  **and**

*primitive*  $ws$  **and**

$\neg$  *primitive* (*concat*  $ws$ )

**obtains**  $j k$  **where**  $1 \leq j$   $1 \leq k$   $j = 1 \vee k = 1$

$ws \sim [x]^{\textcircled{a} j} \cdot [y]^{\textcircled{a} k}$

⟨*proof*⟩

### 0.7.1 The target theorem

Given a binary code  $\{x, y\}$  such that there is a primitive factorisation  $ws$  over it whose concatenation is imprimitive, we finally show that there are integers  $j$  and  $k$  (depending only on  $\{x, y\}$ ) such that any other such factorisation  $ws'$  is conjugate to  $[x]^{\textcircled{a} j} \cdot [y]^{\textcircled{a} k}$ .

**theorem** *bin-imprim-code*: **assumes**  $x \cdot y \neq y \cdot x$  **and**  $ws \in \text{lists } \{x,y\}$  **and**

$2 \leq |ws|$  **and** *primitive*  $ws$  **and**  $\neg$  *primitive* (*concat*  $ws$ )

**obtains**  $j k$  **where**  $1 \leq j$  **and**  $1 \leq k$  **and**  $j = 1 \vee k = 1$

$\bigwedge ws. ws \in \text{lists } \{x,y\} \implies 2 \leq |ws| \implies$

$(\text{primitive } ws \wedge \neg \text{primitive } (\text{concat } ws)) \iff ws \sim [x]^{\textcircled{a} j} \cdot [y]^{\textcircled{a} k}$  **and**

$|y| \leq |x| \implies 2 \leq j \implies j = 2 \wedge \text{primitive } x \wedge \text{primitive } y$  **and**

$|y| \leq |x| \implies 2 \leq k \implies j = 1 \wedge \text{primitive } x$

⟨*proof*⟩

**definition** *bin-imprim-code* **where** *bin-imprim-code*  $x y \equiv x \cdot y \neq y \cdot x \wedge (\neg$   
*bin-prim*  $x y)$

**theorem** *bin-imprim-code'*: **assumes** *bin-imprim-code*  $x y$

**obtains**  $j k$  **where**  $1 \leq j$  **and**  $1 \leq k$  **and**  $j = 1 \vee k = 1$

$\wedge ws. ws \in lists \{x,y\} \implies 2 \leq |ws| \implies$   
 $(primitive\ ws \wedge \neg primitive\ (concat\ ws)) \longleftrightarrow ws \sim [x]^{\textcircled{2}j} \cdot [y]^{\textcircled{2}k}$  **and**  
 $|y| \leq |x| \implies 2 \leq j \implies j = 2 \wedge primitive\ x \wedge primitive\ y$  **and**  
 $|y| \leq |x| \implies 2 \leq k \implies j = 1 \wedge primitive\ x$   
 <proof>

**end**

**theory** *Binary-Imprimitive-Decision*

**imports**

*Binary-Code-Imprimitive.Binary-Code-Imprimitive*

**begin**

## 0.8 Upper bound of the power exponent in the canonical imprimitivity witness

**lemma** *LS-power-len-ge*:

**assumes**  $y^{\textcircled{2}k} \cdot x = z^{\textcircled{2}l}$   
**and**  $k * |y| \geq |z| + |y| - 1$   
**shows**  $x \cdot y = y \cdot x$

<proof>

**lemma** *LS-root-len-ge*:

**assumes**  $y^{\textcircled{2}k} \cdot x = z^{\textcircled{2}l}$   
**and**  $1 \leq k$  **and**  $2 \leq l$   
**and**  $x \cdot y \neq y \cdot x$   
**shows**  $(k - 1) * |y| + 2 \leq |z|$

<proof>

**lemma** *LS-root-len-le*:

**assumes**  $y^{\textcircled{2}k} \cdot x = z^{\textcircled{2}l}$   
**and**  $1 \leq k$  **and**  $2 \leq l$   
**and**  $x \cdot y \neq y \cdot x$   
**shows**  $|z| \leq |x| + |y| - 2$

<proof>

**lemma** *LS-exp-le'*:

**assumes**  $y^{\textcircled{2}k} \cdot x = z^{\textcircled{2}l}$   
**and**  $2 \leq l$   
**and**  $x \cdot y \neq y \cdot x$   
**shows**  $k \leq (|x| - 4) \text{ div } |y| + 2$

<proof>

**lemma** *LS-exp-le*:

**assumes**  $x \cdot y^{\textcircled{2}k} = z^{\textcircled{2}l}$

**and**  $2 \leq l$   
**and**  $x \cdot y \neq y \cdot x$   
**shows**  $k \leq (|x| - 4) \text{ div } |y| + 2$   
 ⟨*proof*⟩

**thm** *bin-imprim-expsE*

**lemma** *bin-imprim-code-witnessE*:

**assumes**  $x \cdot y \neq y \cdot x$  **and**  $|y| \leq |x|$   
**and**  $ws \in \text{lists } \{x, y\}$  **and**  $2 \leq |ws|$   
**and** *primitive*  $ws$  **and**  $\neg \text{primitive } (\text{concat } ws)$   
**obtains**  $ws \sim [x, x, y]$   
 $|k$  **where**  $1 \leq k$  **and**  $k \leq (|x| - 4) \text{ div } |y| + 2$   
**and**  $ws \sim [x] \cdot [y]^{\textcircled{a} k}$   
 ⟨*proof*⟩

### 0.8.1 Optimality of the exponent upper bound

**lemma** *examples-bound-optimality*:

**fixes**  $m k$  **and**  $x y z :: \text{binA list}$   
**assumes**  $1 \leq m$  **and**  $k' = 0 \implies m = 1$   
**defines**  $x \equiv a \cdot b \cdot (b \cdot (a \cdot b)^{\textcircled{a} m})^{\textcircled{a} k'} \cdot b \cdot a$   
**and**  $y \equiv b \cdot (a \cdot b)^{\textcircled{a} m}$   
**and**  $z \equiv a \cdot b \cdot (b \cdot (a \cdot b)^{\textcircled{a} m})^{\textcircled{a} (k' + 1)}$   
**and**  $k \equiv k' + 2$   
**shows**  $|y| \leq |x|$  **and**  $x \cdot y^{\textcircled{a} k} = z \cdot z$  **and**  $k = (|x| - 4) \text{ div } |y| + 2$   
 ⟨*proof*⟩

## 0.9 Characterization of binary primitivity preserving morphisms given by a pair of words

**lemma** *len-le-not-bin-primE*:

**assumes**  $|y| \leq |x|$   
**and**  $\neg \text{bin-prim } x y$   
**obtains**  $\neg \text{primitive } (x \cdot x \cdot y)$   
 $|k$  **where**  $1 \leq k$  **and**  $k \leq (|x| - 4) \text{ div } |y| + 2$   
**and**  $\neg \text{primitive } (x \cdot y^{\textcircled{a} k})$   
 ⟨*proof*⟩

**lemma** *bin-prim-xyk*:

**assumes** *bin-prim*  $x y$  **and**  $0 < k$   
**shows** *primitive*  $(x \cdot y^{\textcircled{a} k})$   
 ⟨*proof*⟩

**lemma** *len-le-bin-prim-iff*:

**assumes**  $|y| \leq |x|$   
**shows**  
 $\text{bin-prim } x y \iff \text{primitive } (x \cdot x \cdot y) \wedge (\forall k. 1 \leq k \wedge k \leq (|x| - 4) \text{ div } |y| + 2 \implies \text{primitive } (x \cdot y^{\textcircled{a} k}))$



(is bin-prim  $x y \iff (?xxy \wedge ?xyk)$ )  
 <proof>

**lemma** *len-eq-bin-prim-iff*:  
 assumes  $|x| = |y|$   
 shows *bin-prim*  $x y \iff$  *primitive*  $(x \cdot y)$   
 <proof>

**theorem** *bin-prim-iff*:  
 $bin-prim\ x\ y \iff$   
 (if  $|y| < |x|$   
 then *primitive*  $(x \cdot x \cdot y) \wedge (\forall k. 1 \leq k \wedge k \leq (|x| - 4) \text{ div } |y| + 2 \longrightarrow$   
*primitive*  $(x \cdot y^{\textcircled{a}} k))$   
 else if  $|x| < |y|$   
 then *primitive*  $(y \cdot y \cdot x) \wedge (\forall k. 1 \leq k \wedge k \leq (|y| - 4) \text{ div } |x| + 2 \longrightarrow$   
*primitive*  $(y \cdot x^{\textcircled{a}} k))$   
 else *primitive*  $(x \cdot y)$   
 )  
 <proof>

### 0.9.1 Code equation for *bin-prim* predicate

**context**  
**begin**

**private lemma** *all-less-Suc-conv*:  $(\forall k < n. P (Suc\ k)) \iff (\forall k \leq n. k \geq 1 \longrightarrow P\ k)$   
 <proof>

**lemma** *bin-prim-iff'* [code]:  
 $bin-prim\ x\ y \iff$   
 (if  $|y| < |x|$   
 then *primitive*  $(x \cdot x \cdot y) \wedge (\forall k < (|x| - 4) \text{ div } |y| + 2. \text{primitive } (x \cdot y^{\textcircled{a}} (Suc\ k)))$   
 else if  $|x| < |y|$   
 then *bin-prim*  $y\ x$   
 else *primitive*  $(x \cdot y)$   
 )  
 <proof>

**end**  
**value** *bin-prim*  $(a \cdot b \cdot b \cdot a \cdot a)\ b \text{ --- True}$   
**value** *bin-prim*  $(a \cdot b \cdot b \cdot a)\ b \text{ --- False}$   
**value** *bin-prim*  $(a \cdot b \cdot b \cdot a)\ (b \cdot a \cdot b \cdot a \cdot b) \text{ --- False}$   
**value** *bin-prim*  $(a \cdot b)\ (a \cdot b) \text{ --- False}$   
**value** *bin-prim*  $(a \cdot b)\ (a \cdot b \cdot a \cdot b) \text{ --- False}$   
**value** *bin-prim*  $(a \cdot b \cdot b \cdot a \cdot a)\ (b \cdot b \cdot b \cdot b \cdot b) \text{ --- True}$

## 0.10 Characterization of binary imprimitivity codes

**theorem** *bin-imprim-code-iff*:

$$\begin{aligned}
 & \text{bin-imprim-code } x \ y \longleftrightarrow x \cdot y \neq y \cdot x \wedge \\
 & \quad (\text{if } |y| < |x| \\
 & \quad \quad \text{then } \neg \text{primitive } (x \cdot x \cdot y) \vee (\exists k. 1 \leq k \wedge k \leq (|x| - 4) \text{ div } |y| + 2 \wedge \neg \\
 & \quad \quad \text{primitive } (x \cdot y^{\textcircled{a}} k)) \\
 & \quad \quad \text{else if } |x| < |y| \\
 & \quad \quad \quad \text{then } \neg \text{primitive } (y \cdot y \cdot x) \vee (\exists k. 1 \leq k \wedge k \leq (|y| - 4) \text{ div } |x| + 2 \wedge \neg \\
 & \quad \quad \quad \text{primitive } (y \cdot x^{\textcircled{a}} k)) \\
 & \quad \quad \quad \text{else } \neg \text{primitive } (x \cdot y) \\
 & \quad \quad ) \\
 & \langle \text{proof} \rangle
 \end{aligned}$$

**value** *bin-imprim-code* (a·b·b·a·a) b — False  
**value** *bin-imprim-code* (a·b·b·a) b — True  
**value** *bin-imprim-code* (a·b·b·a) (b·a·b·a·b) — True  
**value** *bin-imprim-code* (a·b) (a·b) — False  
**value** *bin-imprim-code* (a·b) (a·b·a·b) — False  
**value** *bin-imprim-code* (a·b·b·a·a) (b·b·b·b·b) — False

**end**

# References

- [1] E. Barbin-Le Rest and M. Le Rest. Sur la combinatoire des codes à deux mots. *Theor. Comput. Sci.*, 41:61–80, 1985.
- [2] J. Mañuch. Defect effect of bi-infinite words in the two-element case. *Discret. Math. Theor. Comput. Sci.*, 4(2):273–290, 2001.
- [3] J.-C. Spehner. *Quelques problèmes d'extension, de conjugaison et de présentation des sous-monoïdes d'un monoïde libre*. PhD thesis, Université Paris VII, Paris, 1976.