Combinatorics on Words formalized
Binary codes that do not preserve primitivity

Štěpán Holub
Martin Raška

June 6, 2026

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

Contents

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 $x \cdot y$ [@] $k \cdot u \cdot v \cdot y$ [@] $l \cdot x$, 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 $x \cdot y$ @ $k \cdot u \cdot v \cdot y$ @ $l \cdot x$ 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: $x \leq p\ (v \cdot (u \cdot v)^{@}k) \cdot x$ — $x \leq p\ p \cdot x$ and $p \leq p\ q \cdot p$ where $q = v \cdot u$
and
x-suf: $x \leq s\ x \cdot (u \cdot v)^{@}m \cdot u$ — $\leq_s x\ (s \cdot x)$ 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: $|v \cdot (u \cdot v)^{@}k| + |(u \cdot v)^{@}m \cdot u| \leq |x|$ (**is** $|?p| + |?s| \leq |x|$) **and**
$0 < k$ $0 < m$
shows *commutes* $\{u,v,x\}$
⟨*proof*⟩

lemma *no-overlap'*:
assumes
len: $|v \cdot (u \cdot v)^{@}k| + |(u \cdot v)^{@}m \cdot u| \leq |x|$ (**is** $|?p| + |?s| \leq |x|$)
and $0 < k$ $0 < m$
shows $u \cdot v = v \cdot u$
⟨*proof*⟩

lemma *short-overlap*:
assumes
len1: $|x| < |v \cdot (u \cdot v)^{@}k| + |(u \cdot v)^{@}m \cdot u|$ (**is** $|x| < |?p| + |?s|$) **and**
len2: $|v \cdot (u \cdot v)^{@}k| + |(u \cdot v)^{@}m \cdot u| \leq |x| + |u|$ (**is** $|?p| + |?s| \leq |x| + |u|$)
shows *commutes* $\{u,v,x\}$
⟨*proof*⟩

lemma *medium-overlap*:
assumes
len1: $|x| + |u| < |v \cdot (u \cdot v)^{@}k| + |(u \cdot v)^{@}m \cdot u|$ (**is** $|x| + |u| < |?p| + |?s|$)
and
len2: $|v \cdot (u \cdot v)^{@}k| + |(u \cdot v)^{@}m \cdot u| < |x| + |u \cdot v|$ (**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 Considering the primitive root instead

0.3 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* $(\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a})$ (**is** *primitive ?x*) **and**
primitive $(\mathfrak{a}\cdot\mathfrak{b})$ (**is** *primitive ?y*) **and**
$(\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}) \cdot (\mathfrak{a}\cdot\mathfrak{b}) \neq (\mathfrak{a}\cdot\mathfrak{b}) \cdot (\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a})$
(**is** $?x \cdot ?y \neq ?y \cdot ?x$) **and**
$\varepsilon\ (\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}) \cdot (\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a})\ (\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}) \sim_{\mathcal{I}} [(\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}),(\mathfrak{a}\cdot\mathfrak{b}),(\mathfrak{a}\cdot\mathfrak{b}),(\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a})]$
(**is** $\varepsilon\ ?x \cdot ?x\ ?s \sim_{\mathcal{I}} [?x,?y,?y,?x]$)
⟨*proof*⟩

0.3.1 Locale: interpretation

term *refine*
locale *square-interp* =
— The basic set of assumptions
— The goal is to arrive at $ws = [x] \cdot [y]\ ^{@}\ k \cdot [x]$ 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**
y-le-x: $|y| \leq |x|$ **and**
ws-lists: $ws \in lists\ \{x,y\}$ **and**
nconjug: $\neg\ \varrho\ x \sim \varrho\ y$ **and**
disj-root: $p\ (Ref\ \{\varrho\ x,\ y\}[x,x])\ s \sim_{\mathcal{D}} ws$

begin

interpretation *xy-code*: *binary-code x y*
⟨*proof*⟩

interpretation *yx-code*: *binary-code y x*
⟨*proof*⟩

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

lemma *nconjug'*: $\neg\ x \sim y$
⟨*proof*⟩

lemma *pref-xx-root-expE*: **assumes** $us \leq p\ [x,\ x]$

obtains e **where** $concat\ us = (\varrho\ x)^{@}e$ **and** $e \le e_\varrho\ x * 2$
⟨*proof*⟩

lemma *pref-xx-exp-le*: **assumes** $(\varrho\ x)^{@}e \le p\ x \cdot x$
shows $e \le e_\varrho\ x*2$
⟨*proof*⟩

lemma *prim-disj-interp*: **assumes** *primitive x* **shows** $p\ [x,x]\ s \sim_{\mathcal{D}} ws$
⟨*proof*⟩

lemma *disjoint*: **assumes** $p1 \le p\ [x,x]\ p2 \le p\ ws$ **shows** $p \cdot concat\ p1 \ne concat\ p2$
⟨*proof*⟩

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

lemma *p-nemp*: $p \ne \varepsilon$
⟨*proof*⟩

lemma *s-nemp*: $s \ne \varepsilon$
⟨*proof*⟩

lemma *ws-nemp*: $ws \ne \varepsilon$
⟨*proof*⟩

lemma *hd-ws-lists*: $hd\ ws \in \{x,\ y\}$
⟨*proof*⟩

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

lemma *kE*: **obtains** k **where** $[hd\ ws] \cdot [y]^{@}k \cdot [last\ ws] = ws$
⟨*proof*⟩

lemma *l-mE*: **obtains** $m\ u\ v\ l$ **where** $(hd\ ws) \cdot y^{@}m \cdot u = p \cdot x$ **and** $v \cdot y^{@}l \cdot (last\ ws) = x \cdot s$ **and**
$u \cdot v = y\ u \ne \varepsilon\ v \ne \varepsilon$ **and** $x \cdot (v \cdot u) \ne (v \cdot u) \cdot x$
⟨*proof*⟩

lemma *last-ws*: $last\ ws = x$
⟨*proof*⟩

lemma *rev-primroot-exp* [*simp*, *cow-simps*]: $e_\varrho\ (rev\ x) = e_\varrho\ x$
⟨*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

0.3.2 Locale with additional parameters

locale *square-interp-plus* = *square-interp* +
fixes *l m u v*
assumes *fst-x*: $x \cdot y^{@}m \cdot u = p \cdot x$ **and**
snd-x: $v \cdot y^{@}l \cdot x = x \cdot s$ **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 $[\![?x \leq p\ ?v \cdot ?x;\ |?v \cdot ?u| < |?x|;\ \leq_s\ ?x\ (?r \cdot ?u \cdot ?v \cdot ?u);\ ?r \in \langle\{?u,\ ?v\}\rangle]\!] \Longrightarrow ?u \cdot ?v = ?v \cdot ?u$ and exploits the symmetric interpretation.

lemma *fst-exp-zero*: **assumes** $m = 0$ **and** $0 < l$ **shows** *False*
⟨*proof*⟩

lemma *snd-exp-zero*: **assumes** $0 < m$ **and** $l = 0$ **shows** *False*
⟨*proof*⟩

Both exponents positive: impossible

lemma *both-exps-pos*: **assumes** $0 < m$ **and** $0 < l$ **shows** *False*
⟨*proof*⟩

thm *suf-cancel-conv*

end

0.3.3 Back to the main locale

context *square-interp*

begin

definition u **where** $u = x^{-1>}(p \cdot x)$
definition v **where** $v = (x \cdot s)^{<-1}x$

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*: $v \neq \varepsilon$
⟨*proof*⟩

lemma *cover*: $x \cdot y \cdot x = p \cdot x \cdot x \cdot s$
⟨*proof*⟩

lemma *conjug-facs*: $\varrho\ u \sim \varrho\ v$
⟨*proof*⟩

term *square-interp.v*

— We have a detailed information about all words
lemma *bin-sq-interpE*: **obtains** $r\ t\ m\ k\ l$
where $(t \cdot r)^{@}k = u$ **and** $(r \cdot t)^{@}l = v$ **and**
$(r \cdot t)^{@}m \cdot r = x$ **and** $(t \cdot r)^{@}k \cdot (r \cdot t)^{@}\ l = y$
and $(r \cdot t)^{@}k = p$ **and** $(t \cdot r)^{@}\ l = s$ **and** $r \cdot t \neq t \cdot r$ **and**
$0 < k$ **and** $0 < m$ **and** $0 < l$ **and** $k + l \leq m$
⟨*proof*⟩

end

0.3.4 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*))

$\langle proof \rangle$

lemma *p-suf-y*: $p \leq s\ y$
$\langle proof \rangle$

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

lemma *ps-len*: $|p| = |s|$ **and** *p-eq-v*: $p = v$ **and** *s-eq-u*: $s = u$
$\langle proof \rangle$

lemma *v-x-x-u*: $v \cdot x = x \cdot u$
$\langle proof \rangle$

lemma *sp-y*: $s \cdot p = y$
$\langle proof \rangle$

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

lemma *xxy-root*: $x \cdot x \cdot y = (x \cdot p) \cdot (x \cdot p)$
$\langle proof \rangle$

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

end

lemma *prim-sq-interp*:
assumes $x \cdot y \neq y \cdot x$ **and** *primitive* x **and** $|y| \leq |x|$ **and** $ws \in lists\ \{x,\ y\}$ **and** $\neg\ x \sim y$ **and**
$p\ [x,x]\ s \sim_{\mathcal{D}} ws$
shows *square-interp* $x\ y\ p\ s\ ws$
$\langle proof \rangle$

0.4 Global claims

theorem *bin-sq-interpE*:
assumes $x \cdot y \neq y \cdot x$ **and** $|y| \leq |x|$ **and** $ws \in lists\ \{x,\ y\}$ **and** $\neg\ \varrho\ x \sim \varrho\ y$ **and**
$p\ Ref\ \{\varrho\ x,\ y\}\ [x,\ x]\ s \sim_{\mathcal{D}} ws$
obtains $r\ t\ m\ k\ l$ **where** $(r \cdot t)^{@}\ m \cdot r = x$ **and** $(t \cdot r)^{@}\ k \cdot (r \cdot t)^{@}l = y$
$(r \cdot t)^{@}\ k = p$ **and** $(t \cdot r)^{@}\ l = s$ **and** $r \cdot t \neq t \cdot r$ **and** $0 < k\ 0 < m\ 0 < l\ k + l \leq m$
$\langle proof \rangle$

theorem *bin-sq-interp-primE*:
assumes $x \cdot y \neq y \cdot x$ **and** *primitive* x **and** $|y| \leq |x|$ **and** $ws \in$ *lists* $\{x, y\}$ **and** $\neg\ x \sim y$ **and**
$p\ [x,x]\ s \sim_{\mathcal{D}} ws$
obtains $r\ t\ m\ k\ l$ **where** $(r \cdot t)^{@}\ m \cdot r = x$ **and** $(t \cdot r)^{@}\ k \cdot (r \cdot t)^{@} l = y$
$(r \cdot t)^{@}\ k = p$ **and** $(t \cdot r)^{@}\ l = s$ **and** $r \cdot t \neq t \cdot r$ **and** $0 < k\ 0 < m\ 0 < l\ k + l \leq m$
⟨*proof*⟩

theorem *bin-sq-interp*:
assumes $x \cdot y \neq y \cdot x$ **and** $|y| \leq |x|$ **and** $ws \in$ *lists* $\{x, y\}$ **and** $\neg\ \varrho\ x \sim \varrho\ y$ **and**
$p\ Ref\ \{\varrho\ x,\ y\}\ [x,\ x]\ s \sim_{\mathcal{D}} ws$
shows $ws = [x,y,x]$
⟨*proof*⟩

theorem *bin-sq-interp-prim*:
assumes $x \cdot y \neq y \cdot x$ **and** *primitive* x **and** $|y| \leq |x|$ **and** $ws \in$ *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** $|y| \leq |x|$ **and** $ws \in$ *lists* $\{x, y\}$ **and** $\neg\ \varrho\ x \sim \varrho\ y$ **and**
$p\ Ref\ \{\varrho\ x,\ y\}\ [x,\ x]\ s \sim_{\mathcal{D}} ws$ **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** $(r \cdot t)^{@}\ m \cdot r = x$ **and** $(t \cdot r)^{@}\ k \cdot (r \cdot t)^{@}\ k = y$
$(r \cdot t)^{@}\ k = p$ **and** $(t \cdot r)^{@}\ k = s$ **and** $r \cdot t \neq t \cdot r$ **and** $0 < k$ **and** $0 < m$
⟨*proof*⟩

theorem *bin-sq-interp-ext-primE*:
assumes $x \cdot y \neq y \cdot x$ **and** *primitive* x **and** $|y| \leq |x|$ **and** $ws \in$ *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** $(r \cdot t)^{@}\ m \cdot r = x$ **and** $(t \cdot r)^{@}\ k \cdot (r \cdot t)^{@}\ k = y$
$(r \cdot t)^{@}\ k = p$ **and** $(t \cdot r)^{@}\ k = s$ **and** $r \cdot t \neq t \cdot r$ **and** $0 < k$ **and** $0 < m$
⟨*proof*⟩

0.4.1 Examples

Basic example of an extendable cover

lemma *example-imprim-sq-cover*:
fixes $x\ y\ p\ s$
defines $x \equiv \mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$ **and** $y \equiv \mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}$ **and**
$p \equiv \mathfrak{a}{\cdot}\mathfrak{b}$ **and** $s \equiv \mathfrak{b}{\cdot}\mathfrak{a}$

shows $x \cdot y \neq y \cdot x$ **and** $|y| \leq |x|$ **and** *primitive* x
$p\ x{\cdot}x\ s \sim_{\mathcal{I}} [x,y,x]$
$\neg$ *primitive* $(x{\cdot}x{\cdot}y)$
⟨*proof*⟩

Example of a non-extendable cover

lemma *example-prim-sq-cover*:
fixes $x\ y\ p\ s$
defines $x \equiv \mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$ **and** $y \equiv \mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}$ **and**
$p \equiv \mathfrak{a}{\cdot}\mathfrak{b}$ **and** $s \equiv \mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$
shows $x \cdot y \neq y \cdot x$ **and** $|y| \leq |x|$ **and** *primitive* x
$p\ x{\cdot}x\ s \sim_{\mathcal{I}} [x,y,x]$
primitive $(x{\cdot}x{\cdot}y)$
⟨*proof*⟩

Cube cover with a long y

lemma *example-cube-cover*:
fixes $x\ y\ p\ s$
defines $x \equiv \mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$ **and** $y \equiv \mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}$ **and**
$p \equiv \mathfrak{a}{\cdot}\mathfrak{b}$ **and** $s \equiv \mathfrak{b}{\cdot}\mathfrak{a}$
shows $x \cdot y \neq y \cdot x$ **and** $|x| + |y| < |x{\cdot}x{\cdot}x|$ **and** *primitive* x
$p\ x{\cdot}x{\cdot}x\ s \sim_{\mathcal{I}} [x,y,x]$
primitive $(x{\cdot}x{\cdot}y)$
⟨*proof*⟩

lemma *example-pow-cover*:
fixes $x\ y\ p\ s\ n$
assumes $2 \leq n$
defines $x \equiv \mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$ **and** $y \equiv \mathfrak{b}{\cdot}\mathfrak{a}{\cdot}x^{@}(n-2){\cdot}\mathfrak{a}{\cdot}\mathfrak{b}$ **and**
$p \equiv \mathfrak{a}{\cdot}\mathfrak{b}$ **and** $s \equiv \mathfrak{b}{\cdot}\mathfrak{a}$
shows $x \cdot y \neq y \cdot x$
and $|x| + |y| \leq |x^{@}n|$
and *primitive* x
$p\ x^{@}n\ s \sim_{\mathcal{I}} [x,y,x]$
⟨*proof*⟩

Not root-disjoint covers

lemma *example-root-joint-sq-cover*:
fixes $x\ y\ p\ s$
defines $x \equiv \mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$ **and** $y \equiv \mathfrak{a}{\cdot}\mathfrak{b}$ **and**
$p \equiv \mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$ **and** $s \equiv \mathfrak{b}{\cdot}\mathfrak{a}{\cdot}\mathfrak{a}{\cdot}\mathfrak{b}{\cdot}\mathfrak{a}$
shows $x \cdot y \neq y \cdot x$ **and** $|y| \leq |x|$ **and** $\neg$ *primitive* x
$p\ x{\cdot}x\ s \sim_{\mathcal{I}} [x,x,y,x]$
primitive $(x{\cdot}x{\cdot}y)$
⟨*proof*⟩

lemma *example-root-joint-xxyy-cover*:
fixes $x\ y\ p\ s$

defines $x \equiv \mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}$ **and** $y \equiv \mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}$ **and**
$p \equiv \mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}$ **and** $s \equiv \mathfrak{b}\cdot\mathfrak{a}$
shows $x \cdot y \neq y \cdot x$
and $\neg$ *primitive* x
$p\ x\cdot x\cdot y\cdot y\ s \sim_{\mathcal{I}} [x,x,x,y]$
⟨*proof*⟩

lemma *example-root-joint-xxy-cover*:
fixes $x\ y\ p\ s$
defines $x \equiv \mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}$ **and** $y \equiv \mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}$ **and**
$p \equiv \mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}$ **and** $s \equiv \mathfrak{a}$
shows $x \cdot y \neq y \cdot x$
and $\neg$ *primitive* x
$p\ x\cdot x\cdot y\ s \sim_{\mathcal{I}} [x,x,x,x,x]$
⟨*proof*⟩

lemma *example-root-joint-no-overlap*:
fixes $x\ y\ p\ s$
defines $x \equiv \mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}$ **and** $y \equiv \mathfrak{a}$ **and**
$p \equiv \mathfrak{a}\cdot\mathfrak{b}$ **and** $s \equiv \mathfrak{b}\cdot\mathfrak{a}$
shows $x \cdot y \neq y \cdot x$
and $\neg$ *primitive* x
$p\ y\cdot y\ s \sim_{\mathcal{I}} [x]$
⟨*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.

0.5 General primitivity not preserving codes

context *code*

begin

Two nontrivially conjugate elements generated by a code induce a disjoint interpretation.

lemma *shift-disjoint*:
assumes $ws \in lists\ \mathcal{C}$ **and** $ws' \in lists\ \mathcal{C}$ **and** $z \notin \langle\mathcal{C}\rangle$ **and** $z \cdot concat\ ws = concat\ ws' \cdot z$
$us \le p\ ws^{@}n$ **and** $vs \le p\ ws'^{@}n$
shows $z \cdot concat\ us \neq concat\ vs$
$\langle proof \rangle$

This in particular yields a disjoint extendable interpretation of any prefix

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

The conditions are in particular met by imprimitivity witnesses

lemma *imprim-witness-shift*:
assumes $ws \in lists\ \mathcal{C}$ **and** $primitive\ ws$ **and** $\neg\ primitive\ (concat\ ws)$
obtains $z\ n$ **where** $concat\ ws = z^{@}n\ z \notin \langle\mathcal{C}\rangle$ **and**
$z \cdot concat\ ws = concat\ ws \cdot z$ **and** $|z| < |concat\ ws|$ **and** $2 \le n$
$\langle proof \rangle$

end

0.6 Covered uniform square

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

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

lemma *cover-xy-xxy*: **assumes** $|x| = |y|$ **and** $s \neq \varepsilon$ **and** *eq*: $p \cdot x \cdot y \cdot s = x \cdot x \cdot y$
shows $x = y$
$\langle proof \rangle$

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 $x = y$
$\langle proof \rangle$

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 ¬ *primitive* $(x \cdot y)$
⟨*proof*⟩

lemma *cover-xy-yxy*: **assumes** $|x| = |y|$ **and** $p \neq \varepsilon$ **and** ‹$s \neq \varepsilon$› **and** *eq*: $p \cdot x \cdot y \cdot s = y \cdot x \cdot y$
shows ¬ *primitive* $(x \cdot y)$
⟨*proof*⟩

lemma *cover-xy-three*: **assumes** $|ws| = 3$ $ws \in$ *lists* $\{x,y\}$ $|x| = |y|$
$p \cdot (x \cdot y) \cdot s =$ *concat ws* $p \neq \varepsilon$ $s \neq \varepsilon$
shows ¬ *primitive* $(x{\cdot}y) \wedge (ws = [x,y,x] \vee ws = [y,x,y])$
⟨*proof*⟩

lemma *bin-uniform-len*: **assumes** $ws \in$ *lists* $\{x,y\}$ $|x| = |y|$
shows $|$*concat ws*$| = |ws| * |x|$
⟨*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 ¬ *primitive* $(x{\cdot}y)$ **and** $vs = [x,y,x] \vee vs = [y,x,y]$
⟨*proof*⟩

0.6.1 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* ⟷ *primitive* (*concat ws*)
⟨*proof*⟩

lemma *bin-uniform-imprim*: **assumes** $x \cdot y \neq y \cdot x$ **and** $|x| = |y|$ **and** ¬ *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* $(x \cdot y) \vee$ ¬ *primitive* $x \vee$ ¬ *primitive* y
and $ws \in$ *lists* $\{x,y\}$ **and** $2 \leq |ws|$

shows *primitive ws* $\longleftrightarrow$ *primitive* (*concat ws*)
⟨*proof*⟩

0.7 The main theorem

0.7.1 Imprimitive 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 $\in$ *lists* $\{x,y\}$ **and**
$|y| \leq |x|$ **and**
2 $\leq$ *count-list ws x* **and**
count-list ws y $<$ *2* **and**
primitive ws **and**
$\neg$ *primitive* (*concat ws*)
shows *ws* $\sim$ $[x,x,y]$ **and** *primitive x* **and** *primitive y*
⟨*proof*⟩

0.7.2 Conjugate words

lemma *bin-imprim-not-conjug*:
assumes *ws* $\in$ *lists* $\{x,y\}$ **and**
$x \cdot y \neq y \cdot x$ **and**
2 $\leq |ws|$ **and**
primitive ws **and**
$\neg$ *primitive* (*concat ws*)
shows $\neg$ $x \sim y$
⟨*proof*⟩

0.7.3 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* $\in$ *lists* $\{x, y\}$
and *primitive ws* **and** $\neg$ *primitive* (*concat ws*)
and $[x, x] \leq f$ $ws \cdot ws$
shows *ws* $\sim$ $[x, x, y]$
⟨*proof*⟩

0.7.4 Obtaining primitivity with two squares (refining)

lemma *bin-imprim-both-squares-prim*:
assumes $x \cdot y \neq y \cdot x$
and $ws \in lists\ \{x,\ y\}$
and *primitive ws* **and** $\neg$ *primitive* (*concat ws*)
and $[x,\ x] \leq f\ ws \cdot ws$
and $[y,\ y] \leq f\ ws \cdot ws$
and *primitive x* **and** *primitive y*
shows *False*
⟨*proof*⟩

lemma *bin-imprim-both-squares*:
assumes $x \cdot y \neq y \cdot x$
and $ws \in lists\ \{x,\ y\}$
and *primitive ws* **and** $\neg$ *primitive* (*concat ws*)
and $[x,\ x] \leq f\ ws \cdot ws$
and $[y,\ y] \leq f\ ws \cdot ws$
shows *False*
⟨*proof*⟩

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

lemma *bin-imprim-both-twice*:
assumes $x \cdot y \neq y \cdot x$
and $ws \in lists\ \{x,\ y\}$
and *count-list ws x* $\geq$ *2*
and *count-list ws y* $\geq$ *2*
and *primitive ws* **and** $\neg$ *primitive* (*concat ws*)
shows *False*
⟨*proof*⟩

0.8 Examples

lemma $x \neq \varepsilon \Longrightarrow \varepsilon\ (x \cdot x)\ \varepsilon \sim_{\mathcal{I}} [x,x]$

$\langle proof \rangle$

lemma assumes $x = [(0::nat),1,0,1,0]$ **and** $y = [1,0,0,1]$
shows $[0,1]$ $(x \cdot x)$ $[1,0]$ $\sim_{\mathcal{I}}$ $[x,y,x]$
$\langle proof \rangle$

0.9 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 @ $j \cdot y$ @ k.

lemma *bin-imprim-expsE-y*: **assumes** $x \cdot y \neq y \cdot x$ **and**
$ws \in lists\ \{x,y\}$ **and**
$2 \leq |ws|$ **and**
primitive ws **and**
$\neg$ *primitive* (*concat ws*) **and**
count-list ws $y = 1$
obtains j k **where** $1 \leq j$ $1 \leq k$ $j = 1 \vee k = 1$
$ws \sim [x]^{@}j \cdot [y]^{@}k$
$\langle proof \rangle$

lemma *bin-imprim-expsE*: **assumes** $x \cdot y \neq y \cdot x$ **and**
$ws \in lists\ \{x,y\}$ **and**
$2 \leq |ws|$ **and**
primitive ws **and**
$\neg$ *primitive* (*concat ws*)
obtains j k **where** $1 \leq j$ $1 \leq k$ $j = 1 \vee k = 1$
$ws \sim [x]^{@}j \cdot [y]^{@}k$
$\langle proof \rangle$

0.9.1 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 $[x]$ @ $j \cdot [y]$ @ k.

theorem *bin-imprim-code*: **assumes** $x \cdot y \neq y \cdot x$ **and** $ws \in lists\ \{x,y\}$ **and**
$2 \leq |ws|$ **and** *primitive ws* **and** $\neg$ *primitive* (*concat ws*)
obtains j k **where** $1 \leq j$ **and** $1 \leq k$ **and** $j = 1 \vee k = 1$
$\bigwedge ws.\ ws \in lists\ \{x,y\} \Longrightarrow 2 \leq |ws| \Longrightarrow$
(*primitive ws* $\wedge\ \neg$ *primitive* (*concat ws*) $\longleftrightarrow ws \sim [x]^{@}j \cdot [y]^{@}k$) **and**
$|y| \leq |x| \Longrightarrow 2 \leq j \Longrightarrow j = 2 \wedge$ *primitive x* $\wedge$ *primitive y* **and**
$|y| \leq |x| \Longrightarrow 2 \leq k \Longrightarrow j = 1 \wedge$ *primitive x*
$\langle proof \rangle$

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** $1 \le j$ **and** $1 \le k$ **and** $j = 1 \lor k = 1$
$\bigwedge ws.\ ws \in lists\ \{x,y\} \Longrightarrow\ 2 \le |ws| \Longrightarrow$
$(primitive\ ws \land \neg\ primitive\ (concat\ ws) \longleftrightarrow ws \sim [x]^{@}j \cdot [y]^{@}k)$ **and**
$|y| \le |x| \Longrightarrow 2 \le j \Longrightarrow j = 2 \land primitive\ x \land primitive\ y$ **and**
$|y| \le |x| \Longrightarrow 2 \le k \Longrightarrow j = 1 \land primitive\ x$
⟨*proof*⟩

end

theory *Binary-Imprimitive-Decision*
imports
Binary-Code-Imprimitive.Binary-Code-Imprimitive

begin

0.10 Upper bound of the power exponent in the canonical imprimitivity witness

lemma *LS-power-len-ge*:
assumes $y\ ^{@}\ k \cdot x = z\ ^{@}\ l$
and $k * |y| \ge |z| + |y| - 1$
shows $x \cdot y = y \cdot x$
⟨*proof*⟩

lemma *LS-root-len-ge*:
assumes $y\ ^{@}\ k \cdot x = z\ ^{@}\ l$
and $1 \le k$ **and** $2 \le l$
and $x \cdot y \ne y \cdot x$
shows $(k - 1) * |y| + 2 \le |z|$
⟨*proof*⟩

lemma *LS-root-len-le*:
assumes $y\ ^{@}\ k \cdot x = z\ ^{@}\ l$
and $1 \le k$ **and** $2 \le l$
and $x \cdot y \ne y \cdot x$
shows $|z| \le |x| + |y| - 2$
⟨*proof*⟩

lemma *LS-exp-le'*:
assumes $y\ ^{@}\ k \cdot x = z\ ^{@}\ l$
and $2 \le l$
and $x \cdot y \ne y \cdot x$
shows $k \le (|x| - 4)\ div\ |y| + 2$
⟨*proof*⟩

lemma *LS-exp-le*:
assumes $x \cdot y^{@} k = z^{@} l$
and $2 \leq l$
and $x \cdot y \neq y \cdot x$
shows $k \leq (|x| - 4)$ *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 $ws \in$ *lists* $\{x,y\}$ **and** $2 \leq |ws|$
and *primitive ws* **and** ¬ *primitive* (*concat ws*)
obtains $ws \sim [x, x, y]$
| k **where** $1 \leq k$ **and** $k \leq (|x| - 4)$ *div* $|y| + 2$
and $ws \sim [x] \cdot [y]^{@} k$
⟨*proof*⟩

0.10.1 Optimality of the exponent upper bound

lemma *examples-bound-optimality*:
fixes m k **and** x y z :: *binA list*
assumes $1 \leq m$ **and** $k' = 0 \Longrightarrow m = 1$
defines $x \equiv \mathfrak{a} \cdot \mathfrak{b} \cdot (\mathfrak{b} \cdot (\mathfrak{a} \cdot \mathfrak{b})^{@} m)^{@} k' \cdot \mathfrak{b} \cdot \mathfrak{a}$
and $y \equiv \mathfrak{b} \cdot (\mathfrak{a} \cdot \mathfrak{b})^{@} m$
and $z \equiv \mathfrak{a} \cdot \mathfrak{b} \cdot (\mathfrak{b} \cdot (\mathfrak{a} \cdot \mathfrak{b})^{@} m)^{@} (k' + 1)$
and $k \equiv k' + 2$
shows $|y| \leq |x|$ **and** $x \cdot y^{@} k = z \cdot z$ **and** $k = (|x| - 4)$ *div* $|y| + 2$
⟨*proof*⟩

0.11 Characterization of binary primitivity preserving morphisms given by a pair of words

lemma *len-le-not-bin-primE*:
assumes $|y| \leq |x|$
and ¬ *bin-prim x y*
obtains ¬ *primitive* $(x \cdot x \cdot y)$
| k **where** $1 \leq k$ **and** $k \leq (|x| - 4)$ *div* $|y| + 2$
and ¬ *primitive* $(x \cdot y^{@} k)$
⟨*proof*⟩

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

lemma *len-le-bin-prim-iff*:
assumes $|y| \leq |x|$
shows

bin-prim x y ⟷ *primitive* $(x \cdot x \cdot y) \wedge (\forall k.\ 1 \leq k \wedge k \leq (|x| - 4)$ *div* $|y| + 2 \longrightarrow$ *primitive* $(x \cdot y^{@} k))$
(**is** *bin-prim x y* ⟷ (*?xxy* ∧ *?xyk*))
⟨*proof*⟩

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

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

0.11.1 Code equation for *bin-prim* predicate

context
begin

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

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

end
value *bin-prim* (𝔞·𝔟·𝔟·𝔞·𝔞) 𝔟 — True
value *bin-prim* (𝔞·𝔟·𝔟·𝔞) 𝔟 — False
value *bin-prim* (𝔞·𝔟·𝔟·𝔞) (𝔟·𝔞·𝔟·𝔞·𝔟) — False
value *bin-prim* (𝔞·𝔟) (𝔞·𝔟) — False
value *bin-prim* (𝔞·𝔟) (𝔞·𝔟·𝔞·𝔟) — False
value *bin-prim* (𝔞·𝔟·𝔟·𝔞·𝔞) (𝔟·𝔟·𝔟·𝔟·𝔟·𝔟) — True

0.12 Characterization of binary imprimitivity codes

theorem *bin-imprim-code-iff*:
bin-imprim-code x y $\longleftrightarrow$ $x \cdot y \neq y \cdot x$ $\wedge$
(if $|y| < |x|$
then $\neg$ *primitive* $(x \cdot x \cdot y)$ $\vee$ $(\exists k.\ 1 \leq k \wedge k \leq (|x| - 4)\ \mathit{div}\ |y| + 2 \wedge \neg$ *primitive* $(x \cdot y\ ^{@}\ k))$
else if $|x| < |y|$
then $\neg$ *primitive* $(y \cdot y \cdot x)$ $\vee$ $(\exists k.\ 1 \leq k \wedge k \leq (|y| - 4)\ \mathit{div}\ |x| + 2 \wedge \neg$ *primitive* $(y \cdot x\ ^{@}\ k))$
else $\neg$ *primitive* $(x \cdot y)$
)
⟨*proof*⟩

value *bin-imprim-code* ($\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}$) $\mathfrak{b}$ — False
value *bin-imprim-code* ($\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{a}$) $\mathfrak{b}$ — True
value *bin-imprim-code* ($\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{a}$) ($\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}$) — True
value *bin-imprim-code* ($\mathfrak{a}\cdot\mathfrak{b}$) ($\mathfrak{a}\cdot\mathfrak{b}$) — False
value *bin-imprim-code* ($\mathfrak{a}\cdot\mathfrak{b}$) ($\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{b}$) — False
value *bin-imprim-code* ($\mathfrak{a}\cdot\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{a}\cdot\mathfrak{a}$) ($\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{b}\cdot\mathfrak{b}$) — 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. Maňuch. Defect effect of bi-infinite words in the two-element case. *Discret. Math. Theor. Comput. Sci.*, 4(2):273–290, 2001.

[3] J.-P. Spehner. *Quelques problèmes d'extension, de conjugaison et de presentation des sous-monoïdes d'un monoïde libre.* PhD thesis, Université Paris VII, Paris, 1976.