Combinatorics on Words formalized
Two Generated Word Monoids Intersection

Štěpán Holub
Štěpán Starosta

March 17, 2025

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

Contents

theory *Two-Generated-Word-Monoids-Intersection*
imports *Combinatorics-Words.Equations-Basic Combinatorics-Words.Binary-Code-Morphisms Combinatorics-Words-Graph-Lemma.Glued-Codes*
begin

The characterization of intersection of binary languages formalized here is due to [1].

Chapter 1

Binary Intersection Formalized

locale *binary-codes-coincidence-two-generators* = *binary-codes-coincidence* +
  assumes *two-coins*: $\exists\ r\ s\ r'\ s'.\ g\ r =_m h\ s \wedge g\ r' =_m h\ s' \wedge (r,s) \neq (r',s')$

begin

lemma *criticalE'*:
  obtains *p q r1 s1 r2 s2* **where**
    $g\ p \cdot \alpha_g = h\ q \cdot \alpha_h$ **and**
    $g\ (p \cdot r1) = h\ (q \cdot s1)$ **and**
    $g\ (p \cdot r2) = h\ (q \cdot s2)$ **and**
    $r1 \neq \varepsilon$ **and** $r2 \neq \varepsilon$ **and**
    $hd\ r1 \neq hd\ r2$
⟨*proof*⟩

lemma *alphas-suf*: $\alpha_h \leq s\ \alpha_g$
⟨*proof*⟩

lemma *c-def*: $\mathrm{c} \cdot \alpha_h = \alpha_g$
  ⟨*proof*⟩

lemma *marked-version-solution-conv*: $g_m\ r = h_m\ s \longleftrightarrow g\ r \cdot \mathrm{c}\ = \mathrm{c} \cdot\ h\ s$
  ⟨*proof*⟩

lemma *criticalE*:
  obtains *p q r1 s1 r2 s2* **where**
    $\alpha_g \cdot g_m\ p = \alpha_h \cdot h_m\ q$ **and**
    $\bigwedge p'\ q'.\ \alpha_g \cdot g_m\ p' = \alpha_h \cdot h_m\ q' \Longrightarrow p \leq p\ p' \wedge q \leq p\ q'$ **and**
    $g_m\ (r1 \cdot p) = h_m\ (s1 \cdot q)$ **and**
    $g_m\ (r2 \cdot p) = h_m\ (s2 \cdot q)$ **and**
    $r1 \cdot p \neq \varepsilon$ **and** $r2 \cdot p \neq \varepsilon$ **and**
    $hd\ (r1 \cdot p) \neq hd\ (r2 \cdot p)$
⟨*proof*⟩

Defining the beginning block

definition *beginning-block* :: *binA list* $*$ *binA list* **where**
beginning-block = (*SOME pair*. $\alpha_g \cdot g_m$ (*fst pair*) = $\alpha_h \cdot h_m$ (*snd pair*) $\wedge$
($\forall$ p' q'. $\alpha_g \cdot g_m\ p' = \alpha_h \cdot h_m\ q' \longrightarrow$ (*fst pair*) $\leq_p p' \wedge$ (*snd pair*) $\leq_p q'$))

definition *fst-beginning-block* (‹p›) **where**
fst-beginning-block $\equiv$ *fst beginning-block*
definition *snd-beginning-block* (‹q›) **where**
snd-beginning-block $\equiv$ *snd beginning-block*

lemma *begin-block*: $\alpha \cdot g_m\ p = h_m\ q$ **and**
begin-block-min: $\alpha \cdot g_m\ p' = h_m\ q' \Longrightarrow p \leq_p p' \wedge q \leq_p q'$
⟨*proof*⟩

lemma *begin-block-conjug-conv*:
assumes $r \cdot p = p \cdot r'$ **and** $s \cdot q = q \cdot s'$
shows $g\ r = h\ s \longleftrightarrow g_m\ r' = h_m\ s'$
⟨*proof*⟩

lemma *solution-ext-conv*: $g\ r = h\ s \longleftrightarrow \alpha \cdot g_m\ (r \cdot p) = h_m\ (s \cdot q)$
⟨*proof*⟩

Both block exist

lemma *both-blocks*: *marked.blockP c*
⟨*proof*⟩

notation *marked.suc-fst* (‹$\mathfrak{e}$›) **and**
marked.suc-snd (‹$\mathfrak{f}$›)

lemma *sucs-eq*: $g_m\ (\mathfrak{e}\ \tau) = h_m\ (\mathfrak{f}\ \tau)$
⟨*proof*⟩

sublocale *marked*: *two-binary-marked-blocks* $g_m\ h_m$
⟨*proof*⟩

1.1 Blocks and intersection

Every solution has a block decomposition. However, not all block combinations yield a solution. This motivates the following definition.

definition *coin-block* **where** *coin-block* $\tau\ \equiv p \leq_s p \cdot (\mathfrak{e}\ \tau) \wedge q \leq_s q \cdot (\mathfrak{f}\ \tau)$

theorem *char-coincidence*:
$g\ r = h\ s \longleftrightarrow (\exists\ \tau.$ *coin-block* $\tau \wedge r = (p \cdot \mathfrak{e}\ \tau)^{<-1} p \wedge s = (q \cdot \mathfrak{f}\ \tau)^{<-1} q)$ (**is** $g\ r = h\ s \longleftrightarrow$ *?Q*)
⟨*proof*⟩

theorem *char-coincidence'*:

$g\ r = h\ s \longleftrightarrow (g_m\ (p^{-1>}(r \cdot p)) = h_m\ (q^{-1>}(s \cdot q)) \wedge p \leq_p r \cdot p \wedge q \leq_p s \cdot q)$
(**is** $g\ r = h\ s \longleftrightarrow ?Q$)
⟨*proof*⟩

theorem *coincidence-eq-blocks*: $\mathfrak{C}\ g\ h = \{((p \cdot \mathfrak{e}\ \tau)^{<-1}p, (q \cdot \mathfrak{f}\ \tau)^{<-1}q) \mid \tau.\ \textit{coin-block}\ \tau\}$
⟨*proof*⟩

lemma
minblock0: $g_m\ (\mathfrak{e}\ \mathfrak{a}) =_m h_m\ (\mathfrak{f}\ \mathfrak{a})$ **and**
minblock1: $g_m\ (\mathfrak{e}\ \mathfrak{b}) =_m h_m\ (\mathfrak{f}\ \mathfrak{b})$ **and**
hdblock0: $hd\ (\mathfrak{e}\ \mathfrak{a}) = bina$ **and**
hdblock1: $hd\ (\mathfrak{e}\ \mathfrak{b}) = binb$
⟨*proof*⟩

definition $\mathcal{T}$ **where** $\mathcal{T} \equiv \{\tau\ .\ \textit{coin-block}\ \tau\}$

lemma $\mathcal{T}$*-def'*: $\tau \in \mathcal{T} \longleftrightarrow \textit{coin-block}\ \tau$
⟨*proof*⟩

Properties of the set of coincidence blocks

lemma $\mathcal{T}$*-closed*: **assumes** *coin-block* τ_1 **and** *coin-block* τ_2
shows *coin-block* $(\tau_1 \cdot \tau_2)$
⟨*proof*⟩

lemma *emp-block*: *coin-block* ε
⟨*proof*⟩

lemma $\mathcal{T}$*-hull*: $\langle\mathcal{T}\rangle = \mathcal{T}$
⟨*proof*⟩

lemma $\mathcal{T}$*-pref*: *coin-block* $\tau_1 \Longrightarrow$ *coin-block* $(\tau_1 \cdot \tau_2) \Longrightarrow$ *coin-block* τ_2
⟨*proof*⟩

Translation from blocks to the intersection

lemma *translate-coin-blocks-to-intersection*:
$(h \circ (\lambda\ x.\ (q \cdot x)^{<-1}q) \circ \mathfrak{f})\ `\ \mathcal{T} = \textit{range}\ g \cap \textit{range}\ h$
⟨*proof*⟩

lemma *translation-blocks-inj*:
inj-on $(h \circ (\lambda\ x.\ (q \cdot x)^{<-1}q) \circ \mathfrak{f})\ \langle\mathcal{T}\rangle$
⟨*proof*⟩

lemma *translation-blocks-morph-on*: *morphism-on* $(h \circ (\lambda\ x.\ (q \cdot x)^{<-1}q) \circ \mathfrak{f})\ \mathcal{T}$
⟨*proof*⟩

interpretation *morphism-on* $(h \circ (\lambda\ x.\ (q \cdot x)^{<-1}q) \circ \mathfrak{f})\ \mathcal{T}$
⟨*proof*⟩

theorem *inter-basis*: $\mathfrak{B}\ (range\ g \cap range\ h) = (h \circ (\lambda\ x.\ (q \cdot x)^{<-1} q) \circ \mathfrak{f})\ `\ (\mathfrak{B}\ \mathcal{T})$
⟨*proof*⟩

1.2 Simple blocks

If both letters are blocks, the situation is easy

theorem *simple-blocks*: **assumes** $\bigwedge$ *a. coin-block* [*a*] **shows**
coin-block τ
⟨*proof*⟩

theorem *simple-blocks-UNIV*: $(\bigwedge\ a.\ coin\text{-}block\ [a]) \Longrightarrow \mathcal{T} = UNIV$
⟨*proof*⟩

theorem *simple-blocks-basis*: **assumes** $\bigwedge a.\ coin\text{-}block\ [a]$
shows $\mathfrak{B}\ \mathcal{T} = \{\mathfrak{a}, \mathfrak{b}\}$
⟨*proof*⟩

1.3 At least one block

At least one letter – the last one – is a block

lemma *last-letter-fst-suf*: **assumes** *coin-block* $(z \cdot [c])$
shows $p <s \mathfrak{e}\ [c]$
⟨*proof*⟩

lemma *rich-block-suf-fst′*:
assumes *coin-block* $(z \cdot [1-c] \cdot [c]^{@} Suc\ i)$
shows $gm.bin\text{-}code\text{-}lcs \cdot g_m\ p \leq s\ g_m\ (\mathfrak{e}\ ([1-c] \cdot [c]^{@} Suc\ i))$
⟨*proof*⟩

lemma *rich-block-suf-fst*:
assumes *coin-block* $(z \cdot [1-c] \cdot [c]^{@} Suc\ i)$
shows $\alpha \cdot g_m\ (p) \leq s\ g_m\ (\mathfrak{e}\ ([1-c] \cdot [c]^{@} Suc\ i))$
⟨*proof*⟩

lemma *rich-block-suf-snd′*:
assumes *coin-block* $(z \cdot [1-c] \cdot [c]^{@} Suc\ i)$
shows $\alpha_h \cdot h_m\ q \leq s\ h_m\ (\mathfrak{f}\ ([1-c] \cdot [c]^{@} Suc\ i))$
⟨*proof*⟩

lemma *rich-block-suf-snd*:
assumes *coin-block* $(z \cdot [1-c] \cdot [c]^{@} Suc\ i)$
shows $q \leq s\ \mathfrak{f}\ ([1-c] \cdot [c]^{@} Suc\ i)$
⟨*proof*⟩

lemma *last-letter-block*: **assumes** *coin-block* $(z \cdot [c])$
shows *coin-block* $[c]$

⟨*proof*⟩

end

1.4 Infinite case

locale *binary-codes-coincidence-infinite* = *binary-codes-coincidence-two-generators*
for *a1* +
  assumes *non-block*: ¬ *coin-block* [*a1*]

begin

1.4.1 Description of coincidence blocks

lemma *swap-coin-block*: *coin-block* [*1−a1*]
⟨*proof*⟩

definition *coincidence-exponent* (‹*t*›) **where**
  coincidence-exponent = (*LEAST x*. (*q* ≤*s q* · 𝔣([*a1*] · [*1−a1*]^@*Suc x*)))

lemma *q-nemp*: *q* ≠ *ε*
⟨*proof*⟩

lemma *p-suf*: *p* <*s* 𝔢 [*1−a1*]
  ⟨*proof*⟩

lemma *coin-exp*: *coin-block* ([*a1*] · [*1−a1*]^@*Suc t*) **and**
  coin-exp-min: *j* ≤ *t* ⟹ ¬ *coin-block* ([*a1*] · [*1−a1*]^@*j*)
⟨*proof*⟩

lemma *exp-min*: ¬ *q* ≤*s* 𝔣 [*1−a1*]^@*t*
⟨*proof*⟩

lemma *q-suf-conv*: *q* ≤*s* 𝔣 ([*a1*]·[*1−a1*]^@*Suc k*) ⟷ *t* ≤ *k*
⟨*proof*⟩

lemma *coin-block-with-bad-letter*: **assumes** *a1* ∈ *set w*
  shows *coin-block w* ⟷ [*1−a1*]^@*Suc t* ≤*s w*
⟨*proof*⟩

1.5 Description of the basis

The infinite part of the basis

inductive-set 𝒲 :: *binA list set* **where**
  [*a1*] · [*1−a1*]^@*Suc t* ∈ 𝒲
| *τ* ∈ 𝒲 ⟹ *i* ≤ *t* ⟹ [*a1*] · [*1−a1*]^@*i* · *τ* ∈ 𝒲

lemma 𝒲*-nemp*: *x* ∈ 𝒲 ⟹ *x* ≠ *ε*

⟨*proof*⟩

lemma $\mathcal{W}$*-nemp'*: $x \in (\{[1 - a1]\} \cup \mathcal{W}) \Longrightarrow x \neq \varepsilon$
⟨*proof*⟩

lemma $\mathcal{W}$*-hd*: $x \in \mathcal{W} \Longrightarrow hd\ x = a1$
⟨*proof*⟩

lemma $\mathcal{W}$*-set*: $x \in \mathcal{W} \Longrightarrow a1 \in set\ x$
⟨*proof*⟩

lemma $\mathcal{W}$*-butlast-hd-tl*: $x \in \mathcal{W} \Longrightarrow butlast\ x = [a1] \cdot butlast\ (tl\ x)$
⟨*proof*⟩

lemma $\mathcal{W}$*-suf*: $x \in \mathcal{W} \Longrightarrow [a1] \cdot [1{-}a1]^{@} Suc\ t \leq s\ x$
⟨*proof*⟩

lemma $\mathcal{W}$*-fac*: $x \in \mathcal{W} \Longrightarrow \neg\ [1{-}a1]^{@} Suc\ t \leq f\ butlast\ x$
⟨*proof*⟩

lemma *pref-code-*$\mathcal{W}$: *pref-code* $(\{[1{-}a1]\} \cup \mathcal{W})$
⟨*proof*⟩

lemma $\mathcal{W}$*-coin-blocks*:
assumes $x \in \{[1 - a1]\} \cup \mathcal{W}$ **shows** $x \in \mathcal{T}$
⟨*proof*⟩

lemma $\mathcal{W}$*-gen-T*: $\langle\{[1{-}a1]\} \cup \mathcal{W}\rangle = \mathcal{T}$
⟨*proof*⟩

lemma $\mathcal{W}$*-explicit*: $\mathcal{W} = \{w \cdot [a1] \cdot [1{-}a1]^{@} Suc\ t \mid w.\ w \in \langle\{[a1] \cdot [1{-}a1]^{@} i \mid i.\ i \leq t\}\rangle\}$
⟨*proof*⟩

theorem *infinite-basis*: $\mathfrak{B}\ \mathcal{T} = (\{[1{-}a1]\} \cup \mathcal{W})$
⟨*proof*⟩

end

1.6 Intersection

lemma *bin-inter-coin-set-fst*: $\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle = ((bin\text{-}morph\text{-}of\ x\ y) \circ fst)\ `\ \mathfrak{C}\ (bin\text{-}morph\text{-}of\ x\ y)\ (bin\text{-}morph\text{-}of\ u\ v)$
⟨*proof*⟩

lemma *bin-inter-coin-set-snd*: $\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle = ((bin\text{-}morph\text{-}of\ u\ v) \circ snd)\ `\ \mathfrak{C}\ (bin\text{-}morph\text{-}of\ x\ y)\ (bin\text{-}morph\text{-}of\ u\ v)$
⟨*proof*⟩

theorem *bin-inter-basis*: **assumes** *binary-code x y* **and** *binary-code u v*
shows $\mathfrak{B}\ (\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle) = ((\textit{bin-morph-of}\ u\ v) \circ \textit{snd})\ `\ \mathfrak{C}_m\ (\textit{bin-morph-of}\ x\ y)\ (\textit{bin-morph-of}\ u\ v)$
⟨*proof*⟩

theorem *binary-intersection-code*:
assumes *binary-code x y* **and** *binary-code u v*
shows *code* $\mathfrak{B}\ (\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle)$
⟨*proof*⟩

theorem *binary-intersection*:
assumes *binary-code x y* **and** *binary-code u v*
obtains
$\mathfrak{B}\ (\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle) = \{\}$
|
β **where** $\mathfrak{B}\ (\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle) = \{\beta\}$
|
$\beta\ \gamma$ **where** $\mathfrak{B}\ (\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle) = \{\beta,\gamma\}$
|
$\beta\ \gamma\ \delta\ \mathfrak{t}$ **where** $\delta \neq \varepsilon$ **and** $\gamma \cdot \beta \neq \varepsilon$ **and** $hd\ \delta \neq hd\ (\gamma \cdot \beta)$
$\mathfrak{B}\ (\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle) = \{\beta \cdot \gamma\} \cup \{\beta \cdot (\gamma \cdot \beta)^{@}\mathfrak{t} \cdot w \cdot \delta \cdot \gamma \mid w.\ w \in \langle\{\delta \cdot (\gamma \cdot \beta)^{@}i \mid i.\ i \leq \mathfrak{t}\}\rangle\}$
|
$\beta\ \gamma\ \delta\ \mathfrak{t}\ \mathfrak{q}$ **where** $\delta \neq \varepsilon$**and** $\gamma \cdot \beta \neq \varepsilon$ **and** $hd\ \delta \neq hd\ (\gamma \cdot \beta)$ **and**
$1 \leq \mathfrak{q} \wedge \mathfrak{q} \leq \mathfrak{t}$ **and**
$\mathfrak{B}\ (\langle\{x,y\}\rangle \cap \langle\{u,v\}\rangle) = \{\beta \cdot \gamma\} \cup \{\beta \cdot (\gamma \cdot \beta)^{@}\mathfrak{t} \cdot w \cdot \delta^{<-1}(\beta \cdot (\gamma \cdot \beta)^{@}(\mathfrak{t}-\mathfrak{q}))$
|
$w.\ w \in \langle\{\delta \cdot (\gamma \cdot \beta)^{@}i \mid i.\ i \leq \mathfrak{q} - 1\}\rangle\}$
⟨*proof*⟩

end

References

[1] J. Karhumäki. A note on intersections of free submonoids of a free monoid. *Semigroup forum*, 29:183–206, 1984.