Combinatorics on Words formalized
Graph Lemma

Štěpán Holub
Štěpán Starosta

December 14, 2021

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

Contents

theory *Graph-Lemma*
imports *Combinatorics-Words.Submonoids*

begin

Chapter 1

Graph Lemma

The Graph Lemma is an important tool for gaining information about systems of word equations. It yields an upper bound on the rank of the solution, that is, on the number of factors into all images of unknowns can be factorized. The most straightforward application is showing that a system of equations admits periodic solutions only, which in particular holds for any nontrivial equation over two words.

The name refers to a graph whose vertices are the unknowns of the system, and edges connect front letters of the left- and right- hand sides of equations. The bound mentioned above is then the number of connected components of the graph.

We formalize the algebraic proof from [1]

Let C be a set of generators, and b its distinguished element. We define the set of all products that do not start with b.

inductive-set *no-head* :: $'a$ *list set* $\Rightarrow$ $'a$ *list* $\Rightarrow$ $'a$ *list set*
for C b **where**
emp-in-no-head[*simp*]: $\varepsilon \in$ *no-head* C b
| $u \in C \Longrightarrow u \neq b \Longrightarrow u \in$ *no-head* C b
| $u \neq \varepsilon \Longrightarrow u \in$ *no-head* C $b \Longrightarrow v \in \langle C \rangle \Longrightarrow u \cdot v \in$ *no-head* C b

The set is a submonoid of $\langle C \rangle$

lemma *no-head-sub*: *no-head* C $b \subseteq \langle C \rangle$
$\langle$*proof*$\rangle$

lemma *no-head-closed*: $\langle$*no-head* C $b\rangle$ = *no-head* C b
$\langle$*proof*$\rangle$

We are interested mainly in the situation when C is a code.

context *code*
begin

We characterize the set *no-head* in terms of the decomposition of its (nonempty) elements: the first factor is not b

lemma *no-head-hd*: **assumes** $u \in$ *no-head* $\mathcal{C}$ b **and** $u \neq \varepsilon$ **shows** *hd* (*Dec* $\mathcal{C}$ u) $\neq$ b
⟨*proof*⟩

lemma *b-not-in-no-head*: **assumes** $b \in \mathcal{C}$ **shows** $b \notin$ *no-head* $\mathcal{C}$ b
⟨*proof*⟩

lemma *hd-no-head*: **assumes** $u \in \langle\mathcal{C}\rangle$ **and** *hd* (*Dec* $\mathcal{C}$ u) $\neq b$ **shows** $u \in$ *no-head* $\mathcal{C}$ b
⟨*proof*⟩

corollary *no-head* $\mathcal{C}$ $b = \{u \in \langle\mathcal{C}\rangle.\ u = \varepsilon \vee hd\ (Dec\ \mathcal{C}\ u) \neq b\}$
⟨*proof*⟩

end

The set *no-head* is generated by the following set.

inductive-set *no-head-gen* :: *'a list set* ⇒ *'a list* ⇒ *'a list set*
for C b **where**
$u \in C \Longrightarrow u \neq b \Longrightarrow u \in$ *no-head-gen* C b
| $u \in$ *no-head-gen* C $b \Longrightarrow$ $u \cdot b \in$ *no-head-gen* C b

lemma *no-head-gen-set*: *no-head-gen* C $b = \{z \cdot b^{@}k \mid z\ k.\ z \in C \wedge z \neq b\}$
⟨*proof*⟩

lemma *no-head-genE*: **assumes** $u \in$ *no-head-gen* C b
obtains z k **where** $z \in C$ **and** $z \neq b$ **and** $u = z \cdot b^{@}k$
⟨*proof*⟩

context *code*
begin

We show that this indeed is a set of generators

lemma *emp-not-in-Cb*: $\varepsilon \notin$ *no-head-gen* $\mathcal{C}$ b
⟨*proof*⟩

lemma *no-head-sub'*: $b \in \mathcal{C} \Longrightarrow$ *no-head-gen* $\mathcal{C}$ $b \subseteq$ *no-head* $\mathcal{C}$ b
⟨*proof*⟩

lemma *no-head-generates0*: **assumes** $v \in \langle\mathcal{C}\rangle$ **shows**
$u \neq \varepsilon \longrightarrow u \in \langle$*no-head-gen* $\mathcal{C}$ $b\rangle \longrightarrow u \cdot v \in \langle$*no-head-gen* $\mathcal{C}$ $b\rangle$
⟨*proof*⟩

theorem *no-head-generates*: **assumes** $b \in \mathcal{C}$ **shows** $\langle$*no-head-gen* $\mathcal{C}$ $b\rangle =$ *no-head* $\mathcal{C}$ b

⟨*proof*⟩

Moreover, the generating set *no-head-gen* is a code

lemma *lists-no-head-sub*: $b \in \mathcal{C} \Longrightarrow us \in lists\ (no\text{-}head\text{-}gen\ \mathcal{C}\ b) \Longrightarrow us \in lists\ \langle\mathcal{C}\rangle$
⟨*proof*⟩

lemma *ref-hd*: **assumes** $z \in \mathcal{C}$ **and** $b \in \mathcal{C}$ **and** $z \neq b$ **and** $vs \in lists\ (no\text{-}head\text{-}gen\ \mathcal{C}\ b)$
shows $refine\ \mathcal{C}\ ((z \cdot b^{@}k)\ \#\ vs) = [z] \cdot [b]^{@}k \cdot refine\ \mathcal{C}\ vs$
⟨*proof*⟩

lemma *no-head-gen-code-ind-step*:
assumes $vs \in lists\ (no\text{-}head\text{-}gen\ \mathcal{C}\ b)$ $us \in lists\ (no\text{-}head\text{-}gen\ \mathcal{C}\ b)$ $b \in \mathcal{C}$
and *eq*: $[b]^{@}ku \cdot (refine\ \mathcal{C}\ us) = [b]^{@}kv \cdot (refine\ \mathcal{C}\ vs)$
shows $ku = kv$
⟨*proof*⟩

lemma *no-head-gen-code′*:
$b \in \mathcal{C} \Longrightarrow xs \in lists\ (no\text{-}head\text{-}gen\ \mathcal{C}\ b)$
$\Longrightarrow ys \in lists\ (no\text{-}head\text{-}gen\ \mathcal{C}\ b) \Longrightarrow concat\ xs = concat\ ys \Longrightarrow xs = ys$
⟨*proof*⟩

end

theorem *no-head-gen-code*:
assumes *code* C **and** $b \in C$
shows *code* $\{z \cdot b^{@}k \mid z\ k.\ z \in C \wedge z \neq b\}$
⟨*proof*⟩

We are now ready to prove the Graph Lemma

theorem *graph-lemma*: $\mathfrak{B}_F\ X = \{hd\ (Dec\ (\mathfrak{B}_F\ X)\ x) \mid x.\ x \in X \wedge x \neq \varepsilon\}$
⟨*proof*⟩

1.1 Binary code

We illustrate the use of the Graph Lemma in an alternative proof of the fact that two non-commuting words form a code. See also $[\![u_0 \cdot u_1 \neq u_1 \cdot u_0;\ us \in lists\ \{u_0,\ u_1\};\ vs \in lists\ \{u_0,\ u_1\};\ concat\ us = concat\ vs]\!] \Longrightarrow us = vs$ in *Combinatorics-Words.CoWBasic*.

First, we prove a lemma which is the core of the alternative proof.

lemma *non-comm-hds-neq*: **assumes** $u \cdot v \neq v \cdot u$ **shows** $hd\ (Dec\ \mathfrak{B}_F\ \{u,v\}\ u) \neq hd\ (Dec\ \mathfrak{B}_F\ \{u,v\}\ v)$
⟨*proof*⟩

theorem assumes $u \cdot v \neq v \cdot u$

shows $xs \in lists\ \{u,\ v\} \Longrightarrow ys \in lists\ \{u,\ v\} \Longrightarrow concat\ xs = concat\ ys \Longrightarrow xs = ys$
$\langle proof \rangle$

end

References

[1] J. Berstel, D. Perrin, J. Perrot, and A. Restivo. Sur le théorème du défaut. *Journal of Algebra*, 60(1):169–180, 1979.