Bounded Synchronization Codes and aperiodic Languages

We describe yet another characterization of the class of aperiodic languages. This is also due to M.P. Schützenberger.

Our presentation is based on the proofs of this result by V. Diekert & M. Kufleitner and uses the local monoid idea seen earlier in showing LTL is expressively complete.
The references for the original papers are:

V. Diekert, M. Kufleitner: A survey on the local divisor technique. Theoretical Computer Science Volume 610 Part A. Pages 13-23.

V. Diekert, M. Kufleitner: Omega-Rational Expressions with Bounded Synchronization Delay. Theory of Computing Systems Volume 56. Pages 686-696.

Both papers are available via Manfred Kufleitner's home page.

Let Σ be an alphabet. A prefix code P over Σ is a language $\ni$ if $x, y \in P$ then x is not a prefix of y.

A prefix code is of bounded (synchronization) delay d if it satisfies the following property:

$\forall\, uvw \in P^*$
if $v \in P^d$ then $uv \in P^*$.

Remark: • Given a word $w \in P^*$ there is a unique decomposition of w as
$w = w_1 w_2 w_3 \ldots w_n$, $w_i \in P$.

• So in a bound delay code, if we can find a sequence of d code words then the prefix seen so far is a sequence of codewords & by the uniqueness of decompositions, what is left is the valid parse of the rest of the message.

"Can resynchronize after looking at only d code words"

Usually codes are bounded length and so the delay length is bounded.

Exercise: Give an example of a prefix code that is not bounded delay.

Schützenberger also showed that one may drop complementation from star-free regular expressions and replace it with Kleene-star of bounded delay prefix codes. "Synchronization Delay"

Defn: **SD(A):**

$\emptyset, \{a\} \in SD(A)$ $\quad a \in A$

$L_1 \cup L_2 \in SD(A)$ $\quad L_1, L_2 \in SD(A)$

$L_1 \cdot L_2 \in SD(A)$ $\quad L_1, L_2 \in SD(A)$

$L^* \in SD(A)$, $\quad L \in SD(A)$, L - a prefix code of bounded delay

Theorem: Every aperiodic language over Σ is in $SD(\Sigma)$.

Proof: By ind^n on the size of the monoid and size of alphabet.

Basis:

- Size of monoid is 1:

$$L = \Sigma^* \text{ or } \emptyset$$

Clearly both are in $SD(\Sigma)$

- Size of alphabet is 1:

$$L = \{a^{i_1}, a^{i_2} \cdot a^{i_t}\} \cup \{a^i \mid i \geq N\}$$
$$= \{a^{i_1} \dots a^{i_t}\} \cup a^N \cdot a^*$$

clearly L is in $SD(\{a\})$.

<u>Induction:</u> Let $\Sigma = A \cup \{c\}$.

$$L = h^{-1}(X) \qquad h : \Sigma^* \longrightarrow (M, \cdot, 1)$$

aperiodic

$$L = L_0 \cup L_1 \cup L_2$$
$$L_0 = L \cap A^*$$
$$L_1 = L \cap A^* c A^*$$
$$L_2 = L \cap A^* c A^* c \Sigma^*$$

- L_0 is recognized via h, X but is over a smaller alphabet and hence $L_0 \in SD(A)$. Then as it is $L_0 \in SD(\Sigma)$ too

[$SD(A) \subseteq SD(B)$ whenever $A \subseteq B$] is easy to check from the defn.

- $$L_1 = \bigcup_{\substack{\alpha, \beta \\ \alpha h(c) \beta \in M}} (h^{-1}(\alpha) \cap A^*) \cdot c \cdot (h^{-1}(\beta) \cap A^*)$$

$h^{-1}(\alpha) \cap A^*$, $h^{-1}(\beta) \cap A^*$ are in $SD(A)$ and hence in $SD(\Sigma)$.

$\{c\}$ is in $SD(\Sigma)$.

So $L_1 \in SD(\Sigma)$

- $$L_2 = \bigcup_{\substack{\alpha, \beta, \gamma \\ \alpha\beta\gamma \in X}} (h^{-1}(\alpha) \cap A^*) \cdot (h^{-1}(\beta) \cap c\Delta) \cdot (h^{-1}(\gamma) \cap A^*)$$

<u>where</u> $\Delta = (A^* c)^*$

clearly $h^{-1}(\alpha) \cap A^*$, $h^{-1}(\gamma) \cap A^*$ are in $SD(\Sigma)$ using arguments as above.

<u>What about $h^{-1}(\beta) \cap c\Delta$:</u>

The Wilke trick adapted to the language of monoids by Diekert/Gastin.

Let $g : \Delta \longrightarrow M^*$ ← treat as alphabet

$g(x_1 c x_2 c \dots x_k c) = h(x_1) h(x_2) \dots h(x_r)$ as a word

Let $K_\beta \subseteq M^*$ be defined as follows

$\{ m_1 m_2 \ldots m_k \mid h(c) m_1 h(c) m_2 \ldots m_k h(c) = \beta \}$

clearly,

$$h^{-1}(\beta) \cap c\Delta = c . g^{-1}(K_\beta)$$

<u>claim:</u> $g^{-1}(K)$ is in $SD(\Sigma)$ whenever K is in $SD(M)$. (For any K)

<u>Proof:</u> By indn. on the expression for K.

- $g^{-1}(\phi) = \phi$
- $g^{-1}(\{m\}) = (h^{-1}(m) \cap A^*) . c \in SD(\Sigma^*)$
- $g^{-1}(L_1 \cup L_2) = g^{-1}(L_1) \cup g^{-1}(L_2)$
- $g^{-1}(L_1 . L_2) = g^{-1}(L_1) . g^{-1}(L_2)$
- $g^{-1}(L^*) = (g^{-1}(L))^*$

a prefix code of bounded delay

Suffices to show This language is a prefix code of bounded delay

Let L be a prefix code with delay d.
Consider $g^{-1}(L)$.

(1) Suppose $u \in g^{-1}(L)$, $uv \in g^{-1}(L)$.

$u = x_1 c x_2 c \ldots x_k c \in g^{-1}(L)$

$uv = x_1 c x_2 c \ldots x_k c \, y_1 c y_2 c \ldots y_m c \in g^{-1}(L)$

$\therefore \quad h(x_1) h(x_2) \ldots h(x_k) \in L$

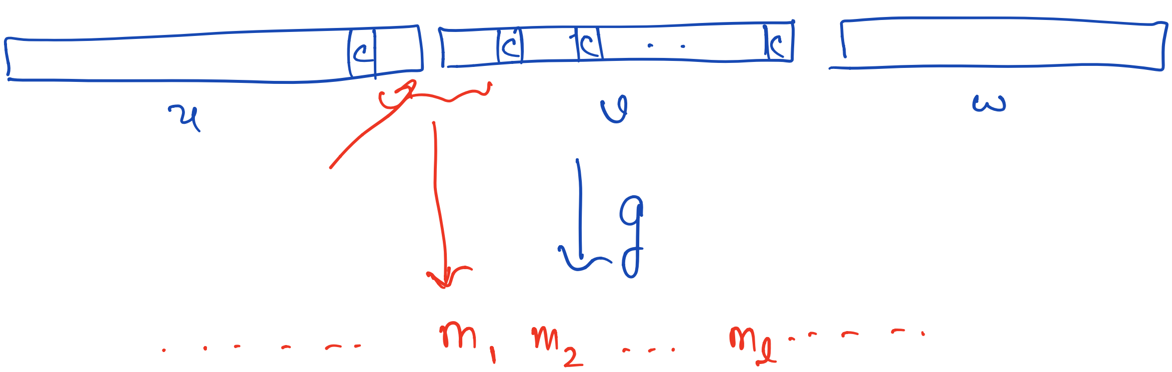
$ \quad h(x_1) h(x_2) \ldots h(x_k) h(y_1) \ldots h(y_m) \in L$

$\Rightarrow \quad h(y_1) \ldots h(y_m) = \varepsilon$ ($\because$ L is a prefix code)

$\Rightarrow \quad m = 0$ (remember, if $y_i = \varepsilon$ then $h(\varepsilon) = 1$)

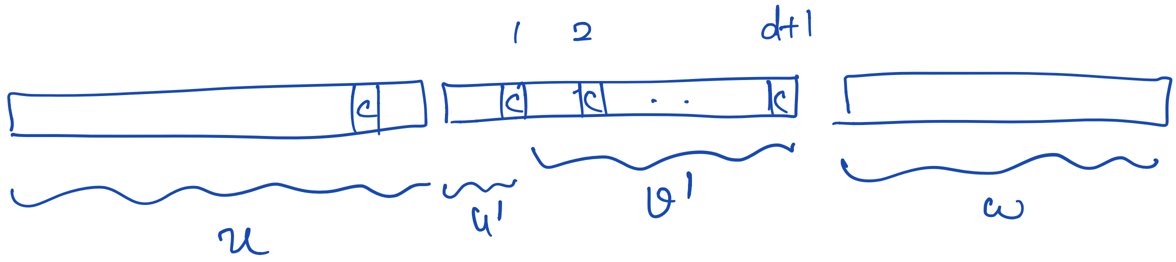
Thus, $v = \varepsilon$ and so $g^{-1}(L)$ is a prefix code.

(2) Suppose $uvw \in (g^{-1}(L))^*$, $v \in (g^{-1}(L))^{d+1}$
(where d is the delay of the prefix code L. The reason for the $+1$ will be explained below)



$g(uvw)$ may not factor as $g(u)g(v)g(w)$.

By adding 1, we ~~omit~~ may omit the segment upto the first c and apply g "homomorphically" & still get into L^d as needed.



$$g(uu'v'w) = g(uu').g(v').g(w) \in L^*$$

But $g(v') \in L^d$

$$\therefore \quad g(uu')g(v') \in L^*$$

$$\Rightarrow \quad g(uu'v') \in L^*$$

$$= \quad g(uv) \in L^*$$

$$\therefore \quad \underline{uv \in (g^{-1}(L))^*} \; (= g^{-1}(L^*))$$

Thus $g^{-1}(K)$ is in $SD(\Sigma)$ whenever $K \in SD(M)$.

$\boxtimes$

Thus to complete the proof, it suffices to show that K_β is in $SD(M)$. Here we use the "localization" construction.

Localization: Let $(M, \cdot, 1)$ be a monoid. Then $(cM \cap Mc, \circ, c)$ is a monoid, known as M localized at c, where

$\mathrm{Loc}_c(M)$

$$xc \circ cy \triangleq xcy$$

claim: $\circ$ is well-defined on $cM \cap Mc$, & $\mathrm{Loc}_c(M)$ is a monoid.

Proof: (1) $\because$ $xc = cx'$
and $cy = y'c$
xcy is also in $cM \cap Mc$

(2) $$xc \circ yc = xc \circ cy'$$
$$= xcy'$$
$$= (xy)c$$

Thus $\circ$ is an associative operation with c as identity.

Claim: $\mathrm{Loc}_c(M)$ divides M.

Proof: Let $M_c = \{m \mid mc \in cM\}$

- M_c is a submonoid of M.

Clearly $1 \in M_c$

Also, if $xc = cy$
$x'c = cy'$

<u>Then</u> $xx'c = xcy' = cyy'$

So $xx' \in M_c$.

- $h : M_c \longrightarrow \mathrm{Loc}_c(M)$

$x \longmapsto xc$ is a morphism.

<u>Proof:</u>

- $1 \xmapsto{h} c$
- $xy \xmapsto{h} xyc$

$$= xcy', \text{ where } yc = cy'$$
$$= (xc)\circ(cy')$$
$$= (xc)\circ(yc)$$
$$= h(x)\circ h(y).$$

- h is surjective.

Let $m \in cM \cap Mc$

$m = cx = yc$

$\therefore$ $y \in M_c$ <u>and</u> $h(y) = m$.

Thus $\mathrm{Loc}_c(M)$ divides M.

Observation:

(1) $1 \in LOC_c(M)$ iff $c = 1$

when M is aperiodic

(2) If e is an idempotent then

$LOC_e(M)$ is a submonoid of M.

Proof:

(1) $1 \in LOC_c(M)$

$\Rightarrow 1 \in cM \cap Mc$

$\Rightarrow 1 \in cM$

$\Rightarrow 1 = cx$

If M is aperiodic $c = x = 1$.

(2) $xe \circ ey = xey$

$xe \cdot ey = xey$

$\boxtimes$

Thus $LOC_c(M)$ aperiodic and a strictly smaller divisor of M whenever M is aperiodic and $c \neq 1$. This allows us to establish our desired result via induction on the size of the monoid.

Lemma: K_β is recognized by $LOC_{h(c)}(M)$.

Proof: Let

$$\sigma : M^* \longrightarrow LOC_{h(c)}(M)$$

$$\sigma(m) = h(c)\, m\, h(c)$$

Then

$$\sigma^{-1}(\beta) = \{ m_1 m_2 \ldots m_n \mid h(c) m_1 h(c) m_2 \ldots h(c) = \beta \}$$

$$= K_\beta .$$

What if $\beta = 1$?

Then, $h^{-1}(\beta) = B^*$ for some B.

$\therefore$ $B^* \cap \Delta = \emptyset$ if $c \notin B$

which is in $SD(Z)$

Θ, $B^* \cap \Delta = B^* c$ if $c \in B$

which is in $SD(Z)$ as well.

Otherwise $LOC_{h(c)}(M)$ is a smaller monoid and so

$$h^{-1}(\beta) \cap \Delta = g^{-1}(K_\beta)$$

But K_β is recognized by a smaller aperiodic monoid & so $K_\beta \in SD(M)$. We already showed that for any $K \in SD(M)$, $g^{-1}(K) \in SD(\Sigma)$ and this completes the proof.

$\boxtimes$

The converse is also true.

Lemma: Let $L \in SD(\Sigma)$. Then L is an aperiodic language.

Proof: $\emptyset$, $\{a\}$ are aperiodic.
Aperiodic languages are closed under $\cup$, concat.
Let $L \in SD(\Sigma)$ be a prefix code of delay d, and let L be aperiodic.
Then, $\exists n$. $xy^n z \in L$ iff $xy^{n+1} z \in L$ $\forall x, y, z$.
We show that there is $m \ni xy^m z \in L^*$ iff $xy^{m+1} z \in L^*$, $\forall x, y, z$.

First we show $\Rightarrow$:

Let
$$xy^m z = v_1 v_2 \dots v_k.$$

We shall arrive at the appropriate value for m by considering two cases.

Case 1: Suppose one of the v_i's includes n full copies of y.

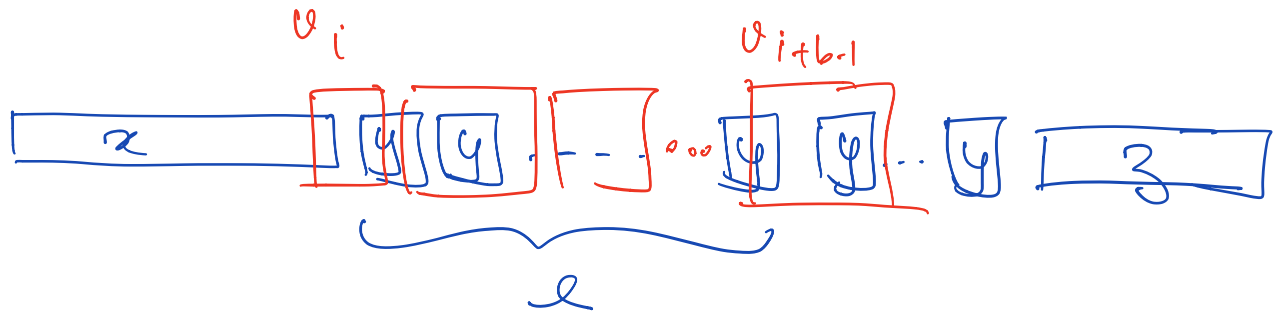
$$\therefore \quad v_i = x' y^n z' \in L$$
$$\Rightarrow \quad v_i' = x' y^{n+1} z' \in L$$
$$\Rightarrow \quad v_1 v_2 \dots v_{i-1} v_i' v_{i+1} \dots v_k = x y^{m+1} z \in L^*$$

Case 2: Suppose all v_i's contain at most $n-1$ full copies of y.

Let $v_i v_{i+1} \dots v_{i+b-1}$ be any segment of length b.

Then, $b(n-1)+b+1$ is an upper bound on the number of y's with any overlap with this segment. (at most $n-1$ contained entirely in each and Each boundary may divide a y.)

Consider the smallest segment $v_i, v_{i+1} \cdots v_{i+b-1}$ that fully contains the first l y's in $xy^m z$.



Also $v_{i+1} \cdots v_{i+b-2}$ is entirely contained within the y^l. (i.e) $b-2$ of them).

$$b(n-1) + b \geq l$$

[The left boundary does not split any y's. So we drop the $+1$]

$$bn \geq l$$

$$b \geq l/n$$

Picking $l \geq (d+2).n$

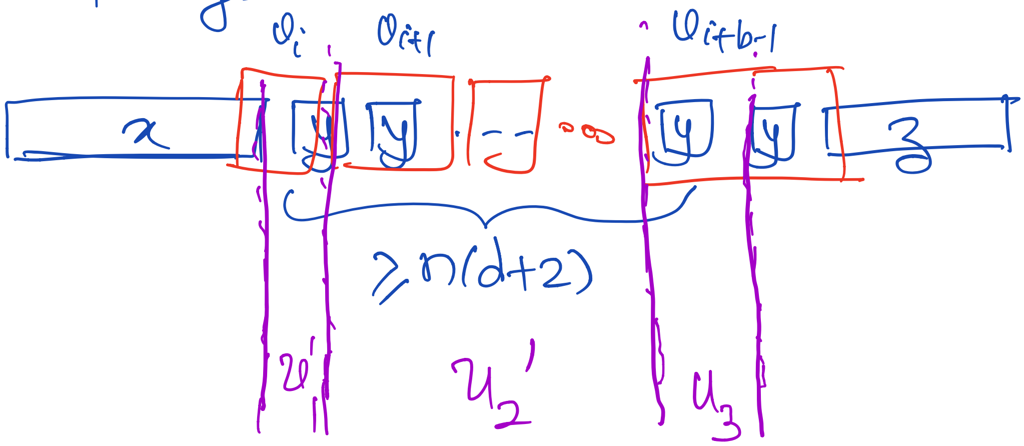
$$b \geq \frac{(d+2)n}{n} = d+2$$

So atleast d v_i's are contained entirely in the first y^l when $l \geq n(d+2)$.

Thus taking $m \geq n(d+2) + 1$

$$xy^m z = x\, y^{n(d+2)} \cdot y\, z$$

and applying the above argument to the first $m-1$ y's



$$x y^m y z = x u_1' u_2' u_3 y z.$$

By the above, $u_2 \in L^r$ for $r > d$. So we may rewrite this as

$$x y^m y z = x u_1 u_2 u_3 y z$$

with $u_2 \in L^d$.

Applying the bounded delay requirement

$$x u_1 u_2 \in L^*, \quad u_3 y z \in L^* \text{ --- } (1)$$

Also, since $u_1 u_2 u_3 y = y u_1 u_2 u_3$ we have

$$x y u_1 u_2 \in L^*, \quad u_3 z \in L^* \text{ --- } (2)$$

Combining the left part of (2) & right part of (1) we get

$$x y u_1 u_2 u_3 y z \in L^*$$

$$\therefore \quad x y^{m+1} z \in L^*$$

Combining the two cases we see that $x y^m z \in L^* \Rightarrow x y^{m+1} z \in L^*$ whenever $m \geq n(d+2)+1$.

Converse: We need to show that for any sufficiently large m, $x y^{m+1} z \in L^* \Rightarrow x y^m z \in L^*$.

The argument above is essentially reversible.

Case 1: If any v_i contains y^{n+1} entirely then

$$v_i = x' y^{n+1} z'$$

let $v_i' = x' y^n z'$

So $v_1 \cdots v_{i-1} v_i' v_{i+1} \cdots v_k = x y^{m-1} z \in L^*$

<u>Case 2:</u> Suppose each v_i contains at most n y's in full. Then repeating the above Case 2, with

$$m \geq (n+1)(d+1)+1$$

we get

$$xy^m yz = xu_1 u_2 u_3 yz$$

with $u_2 \in L^d$.

Applying the bounded $\leq$ delay requirement

$$xu_1 u_2 \in L^*, \quad u_3 yz \in L^* \text{ --- } (1)$$

Also, since $u_1 u_2 u_3 y = y u_1 u_2 u_3$ we have

$$xyu_1 u_2 \in L^*, \quad u_3 z \in L^* \text{ --- } (2)$$

Now combining the left of (1) & right of (2) we get

$$xu_1 u_2 u_3 z \in L^*$$

(ie) $$xy^{m-1} z \in L^* \text{ as required.}$$

Picking an $m \geq (d+2)(n+1)+1$ means both directions hold. Thus,

$$xy^m z \in L^* \quad \underline{\text{iff}} \quad xy^{m+1} z \in L^*$$

Thus every language in $SD(\Sigma)$ is aperiodic.

$\boxtimes$

Having inflicted this somewhat painful calculation on the reader, we now show how to avoid it altogether by a reduction from $SD(\Sigma)$ to star-free expressions over Σ.

<u>Lemma:</u> Every language in $SD(\Sigma)$ has a star-free expression over Σ.

<u>Proof:</u> By ind^n on the SD expression. $\{\varepsilon\}, \{a\}, \emptyset$ are star-free and star-free expressions include $+, \cdot$ it suffices to prove the following claim.

<u>Claim:</u> Let P be a prefix code of bounded delay d that has a star-free expression. Then P^* has a star-free expression.

<u>Proof:</u> Consider any word $w \in \overline{P^*}$.

$w = uv$ where $u \in P^*$ and v contains no prefix in P, $|v| \geq 1$.

$$\{v \mid v \text{ has no prefix in } P\} = \overline{P\Sigma^*}.$$

Thus

$$\overline{P^*} = P^* \cdot \overline{P\Sigma^*}$$

$$= \overline{P\Sigma^*} \cup P.\overline{P\Sigma^*} \cup P^2.\overline{P\Sigma^*} \cdot \cup P^i\, \overline{P\Sigma^*} \cup \ldots \quad — (1)$$

<u>Now:</u>

$$P^{d+i}\, \overline{P\Sigma^*} \subseteq \Sigma^* P^d\, \overline{P\Sigma^*} \quad — (2)$$

<u>Interestingly</u> $\Sigma^* P^d\, \overline{P\Sigma^*} \subseteq \overline{P^*}$

Because $w \in P^*$, $w \in \Sigma^* P^d\, \overline{P\Sigma^*}$ — (3)

means

$$w = xyz \in P^*$$

with $x \in \Sigma^*$, $y \in P^d$, $z \in \overline{P\Sigma^*}$

But by bounded synch delay requirement

$$xy \in P^*$$

An by unique parsability of prefix codes,

$$xy \in P^*, \quad z \in \overline{P\Sigma^*}$$

$$\Rightarrow \quad xyz \notin P^*. \text{ A contradiction}$$

Thus ③ holds.

Using ②, ③, we can rewrite ① has

$$\overline{P^*} = \overline{P\Sigma^*} \cup P\,\overline{P\Sigma^*} \ldots \cup P^d\,\overline{P\Sigma^*} \cup \Sigma^* P^d\,\overline{P\Sigma^*}$$

This completes the proof.

$\boxtimes$