On the superregularity of $\bigcup_k \frac{1}{k}L$

August 25, 2025

1 Intro

Given a regular language L, there is a well-known procedure for transforming its automata to one recognizing the language[1] $\frac{1}{2}L$ of all prefixes of words in L that are exactly half the length, so $\frac{1}{2}L$ is again regular. The same applies to $\frac{1}{k}L$ for any $k \in \mathbb{N}_{\geq 1}$. The construction relies on back pointers moving from the final states a fixed number of steps $(k-1)$ at a time.

We show how such an automata can be constructed recognizing any union $\bigcup_k \frac{1}{k}L$ and still be finite. The index set is omitted because k can surprisingly range over any subset $B \subset \mathbb{N}$ and the resulting language is still regular. We prove a certain regularity in the way superpositions of the mentioned backpointers (simultanously for all k) move through the automata as the input word is read, which allows to reduce their otherwise infiniteness to a finite set of transitions.

Contents

[1] $\frac{1}{2}L = \{w \in \Sigma^* : ww' \in L \text{ for some } w' \text{ with } |w| = |w'|\}$

2 Notation of K-linear automata and languages

This article is based on the language presented in *A Bialgebraic Approach to Automata and Formal Language Theory*[2] by James Worthington. This approach suits us for one main reason: it renders correctness proofs trivial – definition (un)packing and application of linearity properties is all we apply.[3]

We summarize the relevant concepts for understanding this article in this first section, along with some interpretation. Throughout, we use:

- a fixed semiring K quantifying "presense" of words and automata states. For example, $\mathbb{B} \coloneqq \langle \{0,1\}, \vee, \wedge \rangle$ captures sort of a "strict" presense: a word is either present or not.
- a fixed finite alphabet Σ;
- the free K-algebra A generated by Σ. That is, A consists of all (formal) finite sums of K-weighted words over Σ.
- A_n to denote the superposition of words of length $n \in \mathbb{N}$: $A_n \coloneqq \sum_{w \in \Sigma^n} w$. As an actual element of A, and not just a set of words, we get e.g. the succinct notation $w \cdot A_{|w|}$ for all words with prefix w that are double its length.
- $\mathrm{Hom}(M, M')$ to denote semimodule homomorphisms between semimodules M and M'. We use freely generated modules only, which allow to define homomorphisms by their action on the generators. Thus a linear map from the algebra A above is defined by an action over pure words, Σ^*.
- the word **cache** to mean additional tensor product terms to an existing automata's states. This allows us to view an automata with states $M \otimes N$ conceptually as an automata with states M and possible cache values in N.
- for an arbitrary set S, K^S to denote the free semimodule generated by S. K^S consists of (formal, finite) sums $\sum_j k_j s_j$ for elements $k_j \in K, s_j \in S$. Thus the algebra A from above is also the free K-semimodule over Σ^*.
- the symbol $\coloneqq$ to define the value of a newly introduced term,
  $\leftrightharpoons$ to define relations and boolean formulas,
  $\equiv$ to denote an equality which is a direct application of a definition.
- $\subset$ to denote *non-strict* subsets.

[2] https://arxiv.org/abs/0807.4553

[3] My gut feel about this is that the backpointers (and the final states in particular) that we use in the cache are philosophically *dual* to "normal" states. While states are covariant w.r.t. reading, backpointers are contravariant: a word w moves a state *forward* but a backpointer *backward*. Contravariance is well modelled in function arguments and linear algebra exploits this thoroughly.

Denote $\mathrm{Hom}(M, M)$ by $\mathrm{End}\, M$ and $\mathrm{Hom}(M, K)$ by M^*.

Definition 1. A *language* is a homomorphism $L \in \mathrm{Hom}(A, K)$.

That is, a language is determined by the K-quantified presense of words $L(w) \in K$ in it.

Definition 2. A K*-linear automata* $\mathcal{A}$ is a $\begin{array}{c} A \otimes \bullet \\ \downarrow \\ * \longrightarrow \bullet \longrightarrow K \end{array}$ -shaped diagram

$$\begin{array}{ccccc} & & M \otimes A & & \\ & & \big\downarrow{\scriptstyle \triangleleft} & & \\ * & \xrightarrow{s} & M & \xrightarrow{\Omega} & K \end{array}$$

defined by the data[4] $\mathcal{A} = (s, M, \triangleleft, \Omega)$ where

- M is a K-semimodule of (linear combinations of) states;
- $\triangleleft$ is an action of A on M, $\triangleleft \in \mathrm{Hom}(M \otimes A, M)$.

  By the classical isomorphism[5] $\mathrm{Hom}(M \otimes A, M) \cong \mathrm{Hom}(A, \mathrm{End}\, M)$ we interpret $\triangleleft$ as a state transition "read word" map

  $$\triangleleft : A \to \mathrm{End}\, M$$

  that is K-linear and also preserves algebra multiplication: reading a word $w \cdot w'$ is the same as reading w and then w':

  $$m \triangleleft (w \cdot w') = (m \triangleleft w) \triangleleft w',$$

  or just

  $$\triangleleft(ww') = (\triangleleft w') \circ (\triangleleft w),$$

  an equality in $\mathrm{End}\, M$ (composition in the last equality is reversed because of how $\circ$ works ☹).
- $s \in M$ is a start state;
- $\Omega \in \mathrm{Hom}(M, K)$ is an Ωbservation function.
  For example, with $K = \mathbb{B}$, a state[6] $m \in M$ is K-final if $\Omega(m)$ outputs "presense" $\Omega(m) = \top$, otherwise it is non-final.

Such automata are intrinsically non-deterministic. A classical deterministic automata with a state set Q is realized as a $\mathbb{B}$-linear one by taking the free semimodule $M = \mathbb{B}^Q$.

[4] we omit mentioning A here despite its presense in the diagram because it's defined by the fixed alphabet Σ in the whole article.

[5] it holds at least if we assume M is finitely & freely generated, which is a luxury we can afford here. We only construct free semimodules and tensor products of existing semimodules, so if our starting automata are modeled by free semimodules we're all good.

[6] actually any superpositions of states

Definition 3. For an automata $\mathcal{A}$, the *language* of the automata is the homomorphism $\rho_{\mathcal{A}} \in \mathrm{Hom}(A, K)$ defined by completing

$$\begin{array}{ccccc} A & \xrightarrow{\otimes s} & M \otimes A & \xrightarrow{\triangleleft} & M \\ & \searrow^{\rho} & & & \downarrow \Omega \\ & & & & K, \end{array}$$

i.e.

$$\rho_{\mathcal{A}}(w) = \Omega(s \triangleleft w)$$

for all words $w \in \Sigma^*$, extended linearly.

This should be self-explanatory: a word $w \in \Sigma^*$ is K-recognized by $\mathcal{A}$ if the state it reaches, $s \triangleleft w$, is K-final.

If A acts on a semimodule M, it acts naturally on $M^* = \mathrm{Hom}(M, K)$, too:

Definition 4 (Dual action)**.** For $w \in A, m^* \in M^*$, define $m^* \triangleleft^* w \in M^*$ by

$$(m^* \triangleleft^* w)(m) \coloneqq m^*(m \triangleleft w).$$

Or, in point-free style,

$$\triangleleft^* : A \otimes M^* \to M^*$$
$$m^* \triangleleft^* w \coloneqq m^* \circ (\triangleleft w).$$

Proposition 5. *An action $M \triangleleft A$ is compatible with its dual:*

$$(m^* \triangleleft^* w')(m \triangleleft w) = m^*(m \triangleleft ww')$$

for all $m^* \in M^*, m \in M, w, w' \in \Sigma^*$.

Proof. Multiplicativity of $\triangleleft$. □

3 $\frac{1}{k}$ on languages and automata

Definition. For $k \in \mathbb{N}$ and $w, w' \in \Sigma^*$, w is $\frac{1}{k}$-long prefix of ww' if

$$k\,|w| = |ww'|\,.$$

Definition (preliminary)**.** For $k \in \mathbb{N}$ and a language $L \in \mathrm{Hom}(A, K)$, define the language $\frac{1}{k}L$ of all $\frac{1}{k}$-long prefixes of all words in L,

$$\frac{1}{k}L(w) \coloneqq \sum_{w' \in \Sigma^*}^{k|w|=|ww'|} L(ww').$$

Example. For $\Sigma = \{a, b\}$,

$$\frac{1}{2}L(ab) = L(abaa) + L(abab) + L(abba) + L(abbb)$$

so ab is in $\frac{1}{2}L$ iff at least one of $abaa, abab, abba, abbb$ is in L.

Discussion. The operation $\frac{1}{2}$ is known to preserve regularity of languages, and the proof straightforwardly generalizes to $\frac{1}{k}$. The K-linear translation of the construction has the interesting effect that its correctness proof is a direct application of definitions, while the classical way requires one to prove a state invariant upfront. The K-linear construction seems to somehow already embed this invariant, so it is worth it to present it and prove it here.

Discussion. Linearity of L implies we can rewrite

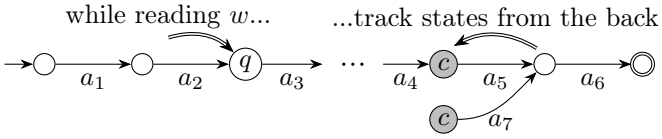
$$\begin{aligned}\frac{1}{k}L(w) \equiv \sum_{w' \in \Sigma^*}^{k|w|=|ww'|} L(ww') &\overset{\text{lin}}{=} L\Big(w \cdot \sum_{w' \in \Sigma^*}^{k|w|=|ww'|} w'\Big) \\ &= L\left(wA_{(k-1)|w|}\right)\end{aligned}$$

so we can redefine more succinctly:

Definition 6 ($\frac{1}{k}L$, again)**.** For a language L and $k \in \mathbb{N}$,

$$\frac{1}{k}L(w) \coloneqq L\left(wA_{(k-1)|w|}\right)$$

Discussion. A well-known automata recognizing $\frac{1}{2}L$ is formed from $\mathcal{A}$ by adding cache that tracks from the back the states from which a final state is reachable in the same number of steps as the length of the word currently being read:



so that when the current state q hits one of the cached c from the back, we know that the word read so far is a prefix of a twice as long word in L. The possible values for the cache are in the powerset 2^M, thus are finitely many and $\frac{1}{2}\mathcal{A}$ is still finite.

For a general $k \geq 1$, an automata recognizing $\frac{1}{k}L$ is formed analogously - in that case the back pointers are moved $(k-1)$ steps on each symbol read.

The formal construction in the K-linear language now follows.

Definition 7. For $k \in \mathbb{N}$ and an automata $\mathcal{A} = (s, M, \triangleleft, \Omega)$ define $\frac{1}{k}\mathcal{A}$ by the diagram

$$\begin{array}{ccccc} & & A \otimes M \otimes M^* & & \\ & & \Big\downarrow{\scriptstyle \triangleleft_{1/k}} & & \\ * & \xrightarrow{s \otimes \Omega} & M \otimes M^* & \xrightarrow{\text{tr}} & K \end{array}$$

where the action of A on $M \otimes M^*$ is defined by

$$(m \otimes f) \triangleleft_{1/k} w \coloneqq (m \triangleleft w) \otimes \left(f \triangleleft^* A_{(k-1)|w|}\right)$$

for $w \in A$ and pure states $m \otimes f$, extended linearly to all $M \otimes M^*$. The backpointer f here is acted upon via the dual action $\triangleleft^*$ as in definition 4.

The observation function here tr is the usual trace operator $\mathrm{tr}(m \otimes f) = f(m)$.

Discussion. Notice how the formal definition of $\triangleleft_{1/k}$ follows directly the idea in the discussion above: the old $\mathcal{A}$ state m is moved forward with w, while the backpointers f are pulled $(k-1)\,|w|$ steps (symbols) back.

Discussion 8. It is also interesting to observe how the pullback is done by a *forward* action $\triangleleft A_{(k-1)|w|}$ applied at a *contravariant* position (i.e. dually). This in contrast with the classical set-theoretic solution where we have to move m forward and the backpointers set backward. Here we only ever use the same go-forward operator $\triangleleft$ and just apply it at different positions.

Proposition 9 (Correctness of $\frac{1}{k}\mathcal{A}$)**.** *For $k \in \mathbb{N}$, an automata $\mathcal{A}$ and its language L, the automata $\frac{1}{k}\mathcal{A}$ defined above recognizes exactly the language $\frac{1}{k}L$:*

$$\rho_{\frac{1}{k}\mathcal{A}} = \frac{1}{k}\rho_{\mathcal{A}}.$$

Proof. (Un)Folding definitions and multiplicativity of $\triangleleft$ applied[7] at $\overset{!}{=}$:

$$\begin{aligned}
\rho_{\frac{1}{k}\mathcal{A}}(w) &\overset{\rho}{\equiv} \mathrm{tr}((s \otimes \Omega) \triangleleft_{1/k} w) \\
&\overset{\triangleleft}{\equiv} \mathrm{tr}\left((s \triangleleft w) \otimes \left(\Omega \triangleleft^* A_{(k-1)|w|}\right)\right) \\
&\overset{\mathrm{tr}}{\equiv} \left(\Omega \triangleleft^* A_{(k-1)|w|}\right)(s \triangleleft w) \\
&\overset{\triangleleft^*}{\equiv} \Omega\left(s \triangleleft w \triangleleft A_{(k-1)|w|}\right) \\
&\overset{!}{=} \Omega\left(s \triangleleft w A_{(k-1)|w|}\right) \\
&\overset{\rho}{\equiv} \rho_{\mathcal{A}}\left(w A_{(k-1)|w|}\right) \\
&\overset{1/k}{\equiv} \frac{1}{k}\rho_{\mathcal{A}}.
\end{aligned}$$

□

4 Regularity of the backpointers

In this section we fix a K-semimodule M of states and an action $M \triangleleft A$. Again, Σ is the alphabet and for $n \in \mathbb{N}$, Σ^n is the set of words of length n. We also fix a set $B \subset \mathbb{N}$.

Here we look at how much chaos theere is (not) in the way our backpointers move.

[7] We could apply proposition 5 to do $\overset{\triangleleft^*}{\equiv}$ and $\overset{!}{=}$ in on go, but we prefer to stress on the trivial character of the proof.

Definition 10. For $n \in \mathbb{N}$, define the operator $\sigma_n \in \operatorname{End} M$, advancing states simultaneously by any number of steps divisible by n,

$$\sigma_n(m) \coloneqq \sum_{k \in B} m \triangleleft A_{kn} = m \triangleleft \left(\sum_{k \in B} A_{kn} \right).$$

Discussion 11. By construction

$$\Omega \circ \sigma_n$$

matches states that can reach a final state in a number of steps divisible by n.

Just like in $\frac{1}{k}\mathcal{A}$, we would like to use $\Omega \circ \sigma_n$ as a cache in an automata for $\sum_{k \in B} \frac{1}{k} L$, so that after reading a word of length $|w| = n$ we can test with $\Omega(\sigma_n(w))$ if w extends to a word ww' in the original language L of length a multiple of n. We could try to use the set $\{\sigma_n : n \in \mathbb{N}\} \subset \operatorname{End} M$, which is finite. The problem is that a mapping in $\operatorname{End} M$ of the form $\sigma_n \mapsto \sigma_{n+1}$ (independent from n) does not generally exist, and so we cannot define an unambiguous 1-step transition function over these states.

Example (an ill-defined $\sigma_n \mapsto \sigma_{n+1}$)**.** Consider for example an automata with 3 states a, b, c, over a 1-letter alphabet, with state transitions $a \mapsto b \mapsto c \mapsto a$. It is easy to see that $\sigma_{3k} = \mathrm{id}$, while $\sigma_{3k+1} = \sigma_{3k+2} = \mathrm{const}_{a+b+c}$. The periodicity of σ is of the shape

$$1\ 0\ 0\ 1\ 0\ 0\ \ldots,$$

and so a right-shift action is only definable if we have access to the current index.

So we need at least some access to the indices. Let us linearize and consider the entire object $\sigma_\bullet$, *including* the indices:

Definition 12. The (unindexed) $\sigma \in \operatorname{Hom}(K^{\mathbb{N}} \otimes M, M)$ then is just

$$\begin{aligned} \sigma &: K^{\mathbb{N}} \otimes M \to M \\ \sigma(n \otimes m) &\coloneqq \sigma_n(m) \end{aligned}$$

for $n \in \mathbb{N}, m \in M$. It is K-linear with respect to the M tensor term because σ_n is linear.

We could use σ directly (or just $K^{\mathbb{N}}$, rather) as a cache for our automata for $\sum_k \frac{1}{k} L$ and it would be correct. It would just not be finite.

Recall that $A_1 = \sum_{a \in \Sigma} a$ is just the superposition of all letters in the alphabet, so $\triangleleft A_1$ is the action of advancing states by one step.

Proposition 13. *The sequence in* $\operatorname{End} M$

$$\triangleleft A_1, \triangleleft A_2, \triangleleft A_3, \triangleleft A_4, \ldots$$

is eventually periodic of some period $p \in \mathbb{N}$, starting from some $n_0 \in \mathbb{N}$.

In fact, it is the sequence of n-fold iterations of $\triangleleft A_1$.

Proof. By multiplicativity of $\triangleleft$, $\triangleleft A_n$ means either side of

$$\triangleleft\left(A_1^n\right) = \left(\triangleleft A_1\right)^n,$$

where $\begin{array}{c} A_1^n \equiv A_1 \cdot \cdots \cdot A_1, \\ \left(\triangleleft A_1\right)^n \equiv \left(\triangleleft A_1\right) \circ \cdots \circ \left(\triangleleft A_1\right) \end{array}$ are repeated concatenation and composition, respectively.

Then $\triangleleft A_{n+1} = \left(\triangleleft A_n\right) \circ \left(\triangleleft A_1\right)$, so every next term in the sequence is entirely determined by the previous one. Each $\triangleleft A_n$ lies in $\operatorname{End} M$, which is finite. □

Proposition 14. *The sequence $\sigma_1, \sigma_2, \ldots$ is eventually periodic.*

That is, the history of our wannabe cache is regular enough, and that will allow us to actually encode it in a *finite* automata.

Proof. Take n_0, p as in the above proposition. For $n \geq n_0$,

$$\sigma_{n+p} \equiv \sum_{k \in B} \triangleleft A_1^{k(n+p)} = \sum_{k \in B} \left(\triangleleft A_1^p\right)^k \circ \left(\triangleleft A_1^{kn}\right) \overset{!}{=} \sum_{k \in B} \triangleleft A_1^{kn} \equiv \sigma_n.$$

The equality $\overset{!}{=}$ uses that $\triangleleft \Sigma^p$ is the identity operator on its input (by the eventual periodicity of $\triangleleft \Sigma$). □

Discussion. The eventual periodicity of $\sigma_1, \sigma_2, \sigma_3, \ldots$ means that a finite prefix exists indexed by $N = \{0, \ldots, n_0, \ldots, n_0 + p - 1\}$ which admits a well-defined (index-dependent) right-shift operation compatible with the global right-shift on the entire σ:

$$\begin{cases} \sigma_n \mapsto \sigma_{n+1} & \text{if } n < n_0 + p - 1 \\ \sigma_n \mapsto \sigma_{n_0} & \text{if } n = n_0 + p - 1 \end{cases}.$$

The next propositions achieves this, but without the case analysis on the indices, and allows for arbitrary $\sigma_n \mapsto \sigma_{n+n'}$ moves directly from periodicity of σ, using a suitable equivalence on $\mathbb{N}$.

Technicalities

So far we established how σ is essentially finite in amount of data. The rest of this section builds the finite analogue of σ formally.

To keep track of time (reading symbols), we prepare

Definition 15. As a monoid Σ^* (multiplicatively) acts on $\mathbb{N}$ (additively) by $n \triangleleft w = n + |w|$.

We now show how the infinite sequence $\sigma : \mathbb{N} \to \operatorname{End} M$ factors

$$\sigma = \hat{\sigma} \circ \pi$$

through some *finite* $\hat{\sigma} : N \to \operatorname{End} M$ that still supports unambiguous right-shifts (unlike the unindexed $\operatorname{End} M$). More precisely:

Proposition 16. *There exists a* finite *factorization*

$$\mathbb{N} \xrightarrow{\pi} N = \mathbb{N}/\!\sim$$

respected both by σ *and the action* $\mathbb{N} \triangleleft \Sigma^*$*. That is, if* $n \sim n'$*, then* $\sigma_n = \sigma_{n'}$ *and for all* $w \in \Sigma^*$*,* $(n \triangleleft w) \sim (n' \triangleleft w)$*.*

This immediately implies

Corollary 17. *The projection* $\pi : \mathbb{N} \to N$ *pushes forward both* σ *and* $\mathbb{N} \triangleleft \Sigma^*$*, inducing*

1. *the so wanted* $\hat{\sigma} : N \to \operatorname{End} M$ *such that* $\sigma_n = \hat{\sigma}_{\pi(n)}$*.*
2. *a well-defined action* $N \triangleleft \Sigma^*$*:* $\pi(n) \triangleleft w = \pi(n + w)$*;*

Proof (of proposition 16). Define N to be the factor of $\mathbb{N}$ over the equivalence relation[8]

$$n \sim n' \leftrightharpoons \forall k \geq 0 : \sigma_{n+k} = \sigma_{n'+k}.$$

N is finite because σ_n is periodic.[9]

1. σ respects $\sim$: put $k = 0$ in the definition of $\sim$.
2. the action $\mathbb{N} \triangleleft A$ respects $\sim$ too: put $k = |w|$ in the definition of $\sim$.

□

Discussion 18. The monoid action $N \triangleleft \Sigma^*$ induces an action $K^N \triangleleft A$ between the respective free K-semimodules (recall A is a free semimodule over Σ^*).

Discussion 19. Here again we can linearize the sequence $\hat{\sigma} : N \to \operatorname{End} M$ to a morphism $\hat{\sigma} \in \operatorname{Hom}(K^N \otimes M, M)$, so that we can use it in the linear automata.

5 The $\sum_{k \in B} \frac{1}{k}$ operation

Definition 20. For an automata $\mathcal{A} = (s, M, \triangleleft, \Omega)$ and a set $B \subset \mathbb{N}$ define an automata

$$\textstyle\sum_{k \in B} \frac{1}{k} \mathcal{A} \coloneqq \left(\hat{s} = s \otimes 0, \hat{M} = M \otimes K^{\mathbb{N}/\sim}, \hat{M} \mathbin{\hat{\triangleleft}} A, \hat{\Omega} \right)$$

with diagram

$$\begin{array}{ccccccc} & & & & A \otimes M \otimes K^N & & \\ & & & & \mathbin{\hat{\triangleleft}} \big\downarrow & & \\ * & \xrightarrow{s \otimes 0} & M \otimes K^N & \xrightarrow{\hat{\sigma}} & M & \xrightarrow{\Omega} & K \end{array}$$

where

[8] That is, we identify positions in the sequence σ_n that have identical tails in order to 1) reduce it to a finite object, but 2) still preserve enough of σ to have a well-defined "going right" operation.

[9] Should we elaborate more, or is this clear enough?

I. The states are
$$\hat{M} \coloneqq M \otimes K^N,$$
where $N = \mathbb{N}/\!\sim$ is the tail-factorization of σ from Proposition 16 (or just the number $N = n_0 + p - 1$ with n_0, p from the propositions earlier). That is, in $\hat{M}$ we use the states of M and cache a congruence on how many symbols we've read.

II. The start state is
$$\hat{s} \coloneqq s \otimes 0,$$
i.e. we start from the start and and record 0 steps from the end (note that $\hat{\sigma}_0 = \mathrm{id}$).

III. The action of A on $\hat{M}$ is defined either point-free by a pair of left-looking eyes
$$\hat{\triangleleft} \coloneqq (\cdot \triangleleft \cdot) \otimes (\cdot \triangleleft \cdot),$$
(the two $\triangleleft$ denoting the actions of A on M and K^N, respectively), pointwise on A as
$$\begin{aligned} A &\to \mathrm{End}\left(M \otimes K^N\right) \\ w &\mapsto (\triangleleft_M w) \otimes (\triangleleft_N w), \end{aligned}$$
or explicitly,
$$(m \otimes n) \mathbin{\hat{\triangleleft}} w \coloneqq (m \triangleleft w) \otimes (n \triangleleft w),$$
for all words $w \in \Sigma^*$ and pure states $m \otimes n$, extended linearly.

That is, we "push m forward with w in the original automata, and pull $\Omega \circ \hat{\sigma}_n$ backward with the length of w".

IV. The final states $\hat{\Omega}$ are defined by completing
$$\begin{array}{ccc} M \otimes K^N & \xrightarrow{\hat{\sigma}} & M \\ & \overset{\hat{\Omega}}{\searrow} & \downarrow \Omega \\ & & K, \end{array}$$
that is,
$$\hat{\Omega}(m \otimes n) \coloneqq \Omega\left(\hat{\sigma}(n \otimes m)\right) \equiv \Omega\left(\hat{\sigma}_n(m)\right),$$
or "we are finished whenever there is a final state (detected by Ω in the original automata) among the states that the current $m \in M$ can reach in a number of steps divisible by n (denoted by $\hat{\sigma}_n(m)$)".

Discussion. The link
$$M \otimes K^N \xrightarrow{\hat{\sigma}} M$$
is arguably the most important arrow in the automata diagram because it tells us at any time which states we'd be in if we were to read (starting from the current state) simultaneously all words of any multiple of the length of the word read so far.

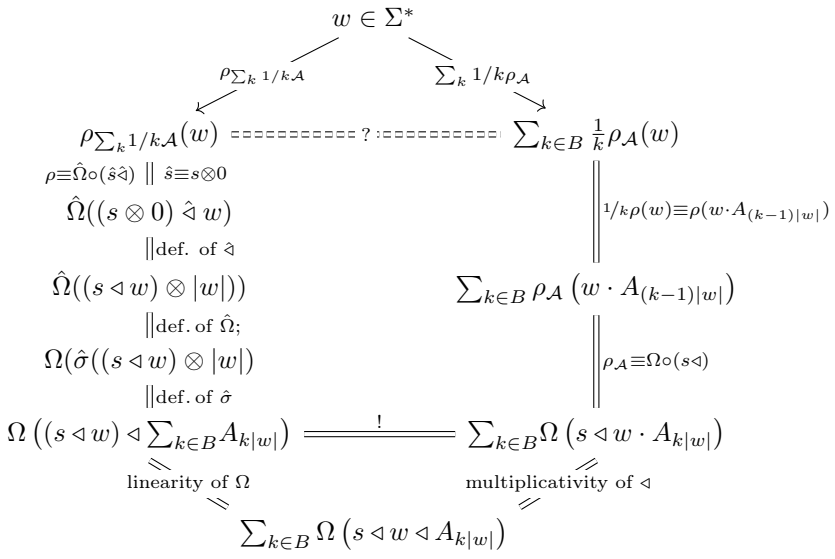
Theorem 21 (Correctness of $\sum_{k\in B}\frac{1}{k}\mathcal{A}$)**.** *The above construction is correct:*

For a set $B \subset \mathbb{N}$, an automata A and its language L, the automata $\sum_{k\in B}\frac{1}{k}\mathcal{A}$ defined above recognizes exactly the language $\sum_{k\in B}\frac{1}{k}L$:

$$\rho_{\sum_{k\in B} 1/k\mathcal{A}} = \sum_{k\in B}\frac{1}{k}\rho_{\mathcal{A}}.$$

Proof. Like for $\frac{1}{k}\mathcal{A}$, the proof is just associativity of $\triangleleft$ and linearity of Ω, squeezed between direct application of definitions.

Definition expansions of the two sides of the equality are laid out on the respective sides here:



An equality chain with all steps explicit follows:

$$\begin{aligned}
\rho_{\sum_{k\in B} 1/k\mathcal{A}}(w) &\overset{\rho}{\equiv} \hat{\Omega}(\hat{s} \mathbin{\hat{\triangleleft}} w) \\
&\overset{\hat{s}}{\equiv} \hat{\Omega}((s \otimes 0) \mathbin{\hat{\triangleleft}} w) \\
&\overset{\hat{\triangleleft}}{\equiv} \hat{\Omega}\Big((s \triangleleft w) \otimes |w|\Big) \\
&\overset{\hat{\Omega}}{\equiv} \Omega\left(\hat{\sigma}\left((s \triangleleft w) \otimes |w|\right)\right) \\
&\overset{\hat{\sigma}}{\equiv} \Omega\left((s \triangleleft w) \triangleleft \textstyle\sum_{k\in B} A_{k|w|}\right) \\
&\underset{\text{lin}}{\overset{\Omega}{\equiv}} \sum_{k\in B} \Omega\left(s \triangleleft w \triangleleft A_{k|w|}\right) \\
&\underset{\text{mul}}{\overset{\triangleleft}{=}} \sum_{k\in B} \Omega\left(s \triangleleft w \cdot A_{k|w|}\right) \\
&\overset{\rho}{\equiv} \sum_{k\in B} \rho_{\mathcal{A}}\left(w \cdot A_{k|w|}\right) \\
&\overset{1/k}{\equiv} \sum_{k\in B} \frac{1}{k}\rho_{\mathcal{A}}(w).
\end{aligned}$$

□

Corollary 22. *If L is a regular language, so is $\sum_{k\in B} \frac{1}{k}L$.*

Proof. If L is regular, it admits an automata $\mathcal{A}$ with a finite number of states M. By the correctness proof, the automata $\sum_{k\in B} \frac{1}{k}\mathcal{A}$ recognizes $\sum_{k\in B} \frac{1}{k}L$. That automata has $M \otimes K^N$ as a state set, which is still finite since M and N are. □

The chain of reasoning is:

$$\begin{array}{c}
M \text{ is finite} \\
\Downarrow \\
\operatorname{End} M \text{ is finite} \\
\Downarrow \\
\sigma_n \text{ is eventually periodic} \\
\Downarrow \\
N = \mathbb{N}/\sim \text{ is finite} \\
\Downarrow \\
\hat{M} = K^N \otimes M \text{ is finite}
\end{array}$$

More duality

Alternatively, we could have defined the $\sum_{k\in B} \frac{1}{k}\mathcal{A}$ automata by

$$\begin{array}{ccccc}
 & & A \otimes M \otimes (K^N \otimes M)^* & & \\
 & & {\scriptstyle \triangleleft\otimes\triangleleft^*}\big\downarrow & & \\
* & \xrightarrow{s\otimes(\Omega\circ\hat{\sigma})} & M \otimes (K^N \otimes M)^* & \xrightarrow{\text{tr}\circ(0\otimes)} & K
\end{array}$$

so that the states $\hat{M} = M \otimes (K^N \otimes M)^*$ track the actual backpointers in the cache $(K^N \otimes M)^*$ and not just a number (K^N). The start state is

$$\hat{s} \coloneqq s \otimes (\Omega \circ \hat{\sigma}),$$

i.e. the backpointers are set to the final states, the transition action is simply advancing the M-state and the backpointers simultaneously

$$(m \otimes f) \mathbin{\hat{\triangleleft}} w \coloneqq (m \triangleleft w) \otimes (f \triangleleft^* w)$$

where the dual action is

$$(f \triangleleft^* w)(n \otimes m) \equiv f\left((n \triangleleft w) \otimes (m \triangleleft w)\right)$$

as before, and the final states are determined by evaluating the backpointer

$$\hat{\Omega}(m \otimes f) \coloneqq f(0 \otimes m).$$

We pass 0 because the backpointer f will internaly advance it as much as necessary, as a consequence of the $\triangleleft^*$ action on it.

This way of presenting it highlights the forward-state/backpointer duality better, much like our $\frac{1}{k}\mathcal{A}$ construction: notice here how Ω is used in the start state and 0 is used in the finals observer, which is the opposite of the construction in definition 20. It is more abstract though and might be confusing to understand on a first read, so we went with the direct K^N-tracking approach above. It has the same feature of only using the forward $\triangleleft$ operator, applied on covariant and contravariant positions as necessary, like in discussion 8.

6 Classical construction

A deficiency of the linear framework in our case is that it does not lend itself easily to arbitrary predicates on the set of the backpointers in the superposition $\sigma_n(m)$.

The classical set-theoretic definition of $\frac{1}{k}L$ is

$$\frac{1}{k}L = \{w \in \Sigma^* : ww' \in L \text{ for some } w' \text{ with } k|w| = |ww'|\}.$$

Assume a classical automata $A = (s, Q, \delta, F)$ is given that recognizes the language L. Fix a set $B \subset \mathbb{N}$ and ask: is $\bigcup_{k\in B} \frac{1}{k}L$ regular?

6.1 Preparation

For ease of notation, instead of δ, we use relation

$$q \xrightarrow{n} q' \leftrightharpoons \exists w \in \Sigma^* : (|w| = n \text{ and } q' \in \delta(q, w))$$

for $n \in \mathbb{N}_0$ and $q, q' \in Q$.

Now we define the classical analogues of the linear operators $(\triangleleft A_n), \sigma_n \in \operatorname{End} M$:

$$\beta_n : 2^Q \to 2^Q$$
$$\beta_n(S) \coloneqq \bigcup_{q\in S} \left\{q' \in Q : q \xrightarrow{n} q'\right\},$$

and

$$\sigma_n : 2^Q \to 2^Q$$
$$\sigma_n(S) \coloneqq \bigcup_{k\in B} \beta_{kn}.$$

Now $\beta_n = (\beta_1)^n$ is the n-fold iteration $\beta_1 \circ \cdots \circ \beta_1$, so each next term $\beta_{n+1} = \beta_1 \circ \beta_n$ depends only on the previous one. Combined with the finiteness of 2^Q, $\{\beta_n\}_n$ is thus eventually periodic of some period p, starting from some position n_0.

Again, for $n \geq n_0$,

$$\sigma_{n+p} \equiv \bigcup_{k\in B} \beta_{k(n+p)} = \bigcup_{k\in B} (\beta_p)^k \circ \beta_{kn} \overset{!}{=} \bigcup_{k\in B} \beta_{kn} \equiv \sigma_n$$

where in $\overset{!}{=}$ we use that β_p is the identity on its input coming from β_{kn}, because $n \geq n_0$ (except for $k = 0$ when the β_p term does not participate at all).

So $\{\sigma_n\}_n$ is periodic too. Now redefine p to be the period of σ_n (just in case σ_n has a shorter period than β_n).

Put $N = \{0, \ldots, n_0, \ldots, n_0 + p - 1\}$ and a successor action on it by

$$\begin{aligned} \mathrm{suc} &: N \to N \\ \mathrm{suc}(n) &\coloneqq \begin{cases} n+1 & \text{if } n < n_0 + p - 1 \\ n_0 & \text{if } n = n_0 + p - 1 \end{cases}, \end{aligned}$$

so that we start from 0 but then cycle between n_0 and $n_0 + p - 1$.

We're ready to define an automata recognizing $\bigcup_k \frac{1}{k} L$.

6.2 Automata construction

Define a common state set and transition function:

$$\begin{aligned} \hat{Q} &\coloneqq Q \times N \\ \hat{s} &\coloneqq (s, 0) \\ \hat{\delta}((q, n), a) &\coloneqq (\delta(q, a), \mathrm{suc}(n)) \,. \end{aligned}$$

To define an automata $\overline{\mathcal{A}} \coloneqq (\hat{s}, \hat{Q}, \hat{\delta}, \hat{F})$, all that is left is to choose the final states. When in state (q, n) we can do a few different comparisons between the set of the future states $\sigma_n\{q\}$ and the set of original finals F:

1. To recognize $\bigcup_k \frac{1}{k} L$, we check if *any* of the relevant future states is final:

$$(q, n) \in F_\cup \leftrightharpoons \sigma_n\{q\} \cap F \neq \emptyset$$

2. ... but just as well we can also consider what would happen if *all* of the relevant future states are final:

$$(q, n) \in F_\supset \leftrightharpoons \sigma_n\{q\} \subset F.$$

3. In $F_\cup$ we checked disjointness, in $F_\supset$ – one way inclusivity; we can also check inclusivity in the other direction:

$$(q, n) \in F_\subset \leftrightharpoons F \subset \sigma_n\{q\}.$$

To prove correctness, assume a word $w \in \Sigma^*$ has been read and the automatas are in the state (q, n). It is an invariant of the automata[10] that $|w|$ and n differ by a multiple of p (a period of σ_n), so $\sigma_{|w|} = \sigma_n$. But $\sigma_{|w|}\{q\}$ is the set of states reachable from q by any relevant[11] multiple of $|w|$ steps. So with

$\hat{F} = F_\cup$**:** $\sigma_{|w|}\{q\}$ contains a final state in F iff w is a prefix of a word of length some multiple of $|w|$ that is in L, i.e. $w \in \bigcup_{k \in B} \frac{1}{k} L$.

$\hat{F} = F_\supset$**:** $\sigma_{|w|}\{q\}$ contains *only* final states iff all Σ^* words ww' of length some multiple of $|w|$ are in L, i.e. $w \left(\Sigma^{|w|}\right)^* \in L$.

$\hat{F} = F_\subset$**:** we don't know what happens.

[10] that is elementary to prove by induction. Here lies the elegance of the K-linear approach for the unions - it does not require upfront invariant proofs. Correctness is embedded in the definition of $\hat{\sigma}_n$.

[11] by "relevant multiple" we mean that the multiplicator is in B

7 Further questions

It is interesting to research what bounds on the size of $N = \mathbb{N} /\sim$ can be established so that we know how much the $\sum_{k \in B} \frac{1}{k}\mathcal{A}$ automata grows with respect to $\mathcal{A}$. The gut feel of the author is that N is bounded by the least common multiple of the lengths of the simple cycles in the classical picture of $\mathcal{A}$ viewed as a graph.