UNIVERSITY OF LIVERPOOL

LEVERHULME TRUST

The Leverhulme Research Centre for Functional Materials Design

k-Universality of Regular Languages

Duncan Adamson[1] Pamela Fleischmann[2] Annika Huch[2]
Tore Koß[3] Florin Manea[3] Dirk Nowotka[2]

[1]University of Liverpool, UK

[2]Kiel University, Germany

[3]University of Göttingen, Germany

diamond Hartree Centre Imperial College London ICNS King Abdullah University of Science and Technology University of Southampton

Subsequences

Definition

A word v is a subsequence of the word w, if there exists a set of positions positions $1 \leq i_1 < i_2 < \ldots < i_k \leq |w|$, such that $v = w[i_1]w[i_2] \cdots w[i_k]$, otherwise, v is an **Absent Subsequence** of v. A word w is k-subsequence universal if **every** word of length k is a subsequence of w.

$$w = thethousandkyoto$$

Subsequences

Definition

A word v is a subsequence of the word w, if there exist positions $1 \leq i_1 < i_2 < \ldots < i_k \leq |w|$, such that $v = w[i_1]w[i_2]\cdots w[i_k]$, otherwise, v is an **Absent Subsequence** of v. A word w is k-subsequence universal if **every** word of length k is a subsequence of w.

$w = thethousandkyoto$

$v = tt$

Subsequences

Definition

A word v is a subsequence of the word w, if there exist positions $1 \leq i_1 < i_2 < \ldots < i_k \leq |w|$, such that $v = w[i_1]w[i_2]\cdots w[i_k]$, otherwise, v is an **Absent Subsequence** of v. A word w is k-subsequence universal if **every** word of length k is a subsequence of w.

$$w = thethousandkyoto$$

$$v = tenkyoto$$

Subsequences

Definition

A word v is a subsequence of the word w, if there exist positions $1 \leq i_1 < i_2 < \ldots < i_k \leq |w|$, such that $v = w[i_1]w[i_2] \cdots w[i_k]$, otherwise, v is an **Absent Subsequence** of v. A word w is k-subsequence universal if **every** word of length k is a subsequence of w.

$$w = thethousandkyoto$$

$$v = tokyo$$

Subsequences

Definition

A word v is a subsequence of the word w, if there exist positions $1 \leq i_1 < i_2 < \ldots < i_k \leq |w|$, such that $v = w[i_1]w[i_2]\cdots w[i_k]$, otherwise, v is an **Absent Subsequence** of v. A word w is k-subsequence universal if **every** word of length k is a subsequence of w.

$$w = \mathit{thethousandkyoto}$$

$$v = \mathit{osaka}$$

Subsequences and Universality

Notation. Subseq(w) denotes the set of subsequences of w, $\text{Subseq}_k(w)$ denotes the set of subsequences of length exactly k.

A word w is k-universal if $\text{Subseq}_k(w) = \Sigma^k$.

$$\text{Subseq}(11100) = \{1, 0, 00, 10, 11, 100, 110, 111, 1100, 1110, 11100\}$$
$$\text{Subseq}_3(11100) = \{100, 110, 111\}$$

Universality Index

Definition

The universality index $\iota(w)$ is the unique integer such that w is $\iota(w)$-universal but not $(\iota(w)+1)$-universal.

Definition (Arch-Factorisation, Hébrard 1991)

Let $w \in \Sigma^*$. Then $w = \mathrm{ar}_w(1) \cdots \mathrm{ar}_w(\iota(w)) r(w)$ such that $\iota(\mathrm{ar}_w(i)) = 1$, the last letter of $\mathrm{ar}_w(i)$ occurs exactly once in $\mathrm{ar}_w(i)$ and $\iota(r(w)) = 0$. $\mathrm{ar}_w(i)$ are called the *arches of w* and $r(w)$ is called the *rest of w*.

Arch Factorisation

$$
\begin{aligned}
w &= 111222312332 1112\\
  &= (1112223)(123)(321)(112)\\
\\
\mathrm{ar}_1(w) &= 11122\mathbf{3}\\
\mathrm{ar}_2(w) &= 12\mathbf{3}\\
\mathrm{ar}_3(w) &= 32\mathbf{1}\\
r(w) &= 112\\
\\
\iota(w) &= 3
\end{aligned}
$$

Finite Automata

Definition

A finite automaton is a 5-tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an alphabet, $\delta : Q \times \Sigma \to 2^Q$ is the transition function, $q_0 \in Q$ is the initial state and $F \subseteq Q$ is a set of final states. If $|\delta(q, a)| \leq 1$ for all $q \in Q$ and $a \in \Sigma$ we call $\mathcal{A}$ *deterministic* (DFA), otherwise we call it *non-deterministic* (NFA).

Definition

Given an automaton $\mathcal{A}$, the word w is **recognised** by $\mathcal{A}$ if the (or at least one) path starting at the initial state q_0 and following the edges with a labelling corresponding to w ends at a final state. The **language** of $\mathcal{A}$ is the set of **all** words recognised by $\mathcal{A}$.

Subsequence Universality for Languages

Definition

- L is k-∃-universal iff there is a word in L which is k-universal.
- L is k-∀-universal iff every word in L is k-universal.

Problem

How efficient can we decide, given a language L defined by a finite automaton $\mathcal{A}$ and an integer k, whether L is k-∃-universal (k-ESU) or k-∀-universal (k-ASU)?

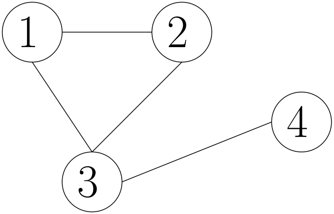
$k - \exists$-universality

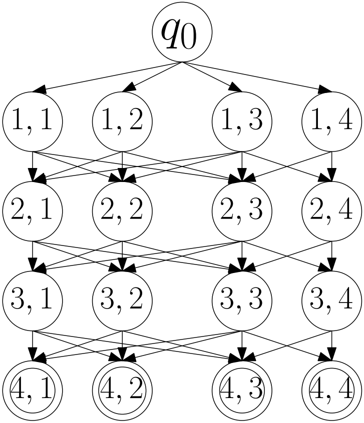
Result. Determining if a language L defined by a finite automaton $\mathcal{A}$ is $k - \forall$-universal is NP-Complete even when $k = 1$.

Sketch. From the Hamiltonian Path problem.

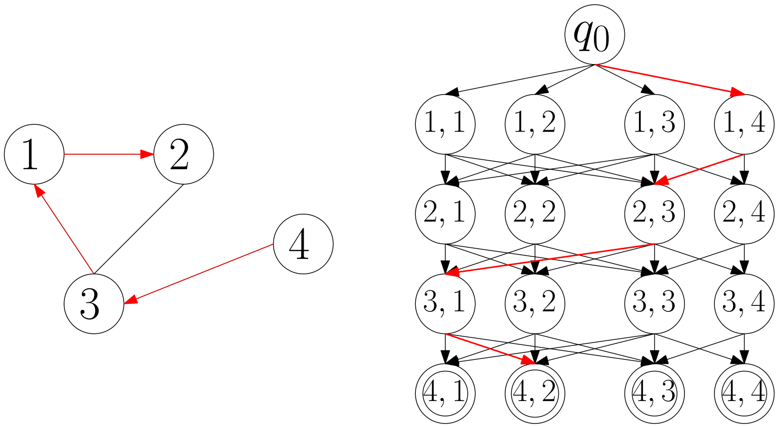
- Take a graph $G = (V, E)$ where $n = |V|$.
- Construct an automaton $\mathcal{A}$ recognising words of length exactly n corresponding to paths of length n in G.
- Therefore, if any word corresponds to a path containing every vertex in G, then that word corresponds to a Hamiltonian path in G.
- As this path is the only one that can be 1-univerisal, L is $1 - \exists$-universal iff G has a Hamiltonian path.

$k-\exists$-universality





$k - \exists$-universality



k-∃-universality – NP-membership

Lemma

If $\mathcal{A}$ accepts a k-universal word it also accepts a k-universal word of length at most $kn\sigma$

Sketch.

- If there is a translation labelled by x, for any $x \in \Sigma$, along any path from the state q, then the shortest path from q containing this transition has length at most n (i.e. the number of states in the automaton).
- As each arch needs σ symbols, to get a one universal word, we need a path of length at most $n \cdot \sigma$ (and thus a word of length $n \cdot \sigma$) to have a 1-universal word/path.
  - N.B., this might note be an accepted word/path, just a prefix of one.
- As we need k such arches, the maximum length of the shortest k-universal word is $nk\sigma$.

k-∃-universality – FPT

Theorem

Given an automaton $\mathcal{A}$ with n states over an alphabet of size σ, we can decide k-ESU in $O^*(n^3 2^\sigma)$ (where the star only hides poly(σ)-factors resulting from arithmetic with large integers).

k-∃-universality – FPT Outline

(i) Remove non-accessible and non-co-accessible states in $O(n^3)$

(ii) Check whether there is a loop labelled with a 1-universal word, if so accept independently from k.

(iii) Otherwise, for every $q \in Q$, find maximal set V_q of letters occurring in a word β_q which is label of a path from q to q (V_q is unique since the path may contain q more than twice) in $O^*(n^3 2^\sigma)$.

(iv) We can maximise the universality of any word $w \in L(\mathcal{A})$ by pumping β_q^2 for every state q in an accepting path labelled with w.

(v) Determine maximal universality of words in $L(\mathcal{A})$ in $O^*(n^3 2^\sigma)$ with dynamic programming: let $M[\cdot][\cdot]$ be an $n \times 2^\sigma$ matrix such that $M[q_r][V]$ is the maximal universality of a word w labelling a path from q_0 to q_r such that $r(w) = V$.

$k - \forall$-Universality

Result. Given a automaton $\mathcal{A}$ with n states, over an alphabet of size σ, we can decide if $\mathcal{A}$ is $k-\forall$-universal in $O(n^3\sigma)$ time.

Note. For any language L the set $L^{\forall}$ of words occurring as subsequences in all words $w \in L$ is finite ($L^{\forall} = \bigcap_{w \in L} \text{Subseq}(w)$ and $\text{Subseq}(w)$ is finite) but can still be exponential in the length of the shortest word in L.

$k - \forall$-Universality-algorithm outline

(i) For $q, q' \in Q$ we define a relation R_a for every $a \in \Sigma$ such that qR_aq' if and only if there is a state q'' such that there is a path from q to q'' not containing any a and also a transition from q'' to q' labelled by a.

(ii) Let qRq' if and only if there is $a \in \Sigma$ such that qR_aq'.

(iii) Let $Q' = \{q \in Q \mid \text{there is a non-universal path from } q \text{ to } F\}$.

(iv) Let $G = (V, E)$ be a directed graph with $V = Q$ and $(q, q') \in E$ if and only if qRq'.

(v) There is an ℓ-universal word, for an $\ell < k$, accepted by $\mathcal{A}$ if and only if there is a path of length at most $k - 1$ from q_0 to any node corresponding to a state in Q' in G.

Counting and Ranking k-universal Words

Let $L \subset \Sigma^*$ be a formal language.

- The problem of counting words of L is to determine the size of L.
- The problem of ranking a word $w \in L$ is to determine the size of the set $\{v \in L \mid v \prec w\}$ where $\prec$ is an arbitrary ordering of Σ^*, e.g. the length-lexicographic ordering.

Note 1. Both problems are NP-hard as answering either with a non-zero value shows that L is $k - \exists$-universal.
Note 2. In NFAs, as a word may correspond to multiple paths, we instead count (resp. rank) the number of paths corresponding to a k-universal word.

Counting

Observation

The number of words accepted by an deterministic automaton $\mathcal{A}$ is equal to the number of paths in $\mathcal{A}$ starting at q_0 and ending at some final state.

Counting

Approach. To count the number of k-subsequence universal paths accepted by the automaton $\mathcal{A}$ of length m.
Let T be a table of size $m+1 \times k \times n \times 2^\sigma$ where $T[\ell, c, q, \mathcal{R}]$, for $\ell \in [0, m], c \in [0, k-1], q \in Q, \mathcal{R} \subset \Sigma$ is the number of c-universal paths of length ℓ ending at state q with a rest of $\mathcal{R}$.Then:

$\mathcal{R} \neq \emptyset$

$$T[\ell, c, q, \mathcal{R}] = \sum_{\substack{q' \in Q \\ x \in \mathcal{R}}} \begin{cases} 0 & \delta(q', x) \neq q \\ \begin{array}{r} T[\ell-1, c, q', \mathcal{R}]+ \\ T[\ell-1, c, q', \mathcal{R} \setminus \{x\}] \end{array} & \delta(q', x) = q \end{cases}$$

$\mathcal{R} = \emptyset$

$$T[\ell, c, q, \mathcal{R}] = \sum_{q' \in Q} \begin{cases} \delta(q', x) \neq q \\ T[\ell-1, c-1, q', \Sigma \setminus \{x\}]] & \delta(q', x) = q \end{cases}.$$

Counting

- We use a second table U of size $m + 1 \times n$ to collect the number of k-universal paths of length 0 to m ending at state q.
- Formally, $U[\ell, q]$ contains the number of k-universal paths of length ℓ ending at state q.
- U is computed analagously to T.
- The total number of k-universal paths of length m is then given by $\sum_{q \in F} U[m, q]$.

Result. The number of k-universal paths of length (resp. at most) m accepted by an automaton $\mathcal{A}$ containing n states, over an alphabet of size σ can be computed in $O^*(m^2 n^2 k 2^\sigma)$

Counting every k-universal word in the language

Observation. The automaton $\mathcal{A}$ accepts a k-universal word if and only if $\mathcal{A}$ accepts a k-universal word of length at most $kn\sigma$.

Result. The number of k-universal paths accepted by an automaton $\mathcal{A}$ containing n states, over an alphabet of size σ can be computed in $O(n^4 k^3 2^\sigma)$ time.

Ranking

Result. The rank of a k-universal path p within the set of all paths (resp. all paths of length at exactly/at most m) accepted by an automaton $\mathcal{A}$ can be computed in $O^*(n^4 k^3 2' \sigma)$ time (reps. $O^*(m^2 n^2 k 2' \sigma)$).

Sketch.

- We use the same approach as for counting, however, now we only allow paths with a prefix of the form $p_1 p_2 \dots p_\ell x$ where $p_1 p_2 \dots p_\ell$ is the prefix of p with length ℓ, and $x < p_\ell$.
- This constraint can be integrated with the tables T and U in the same way as counting.
- This gives the time complexity.

Conclusion

Complexity.

Problem	Complexity Class	Best Algorithm
$k - \exists$-universality	NP-Complete	$O^*(n^3 2^\sigma)$
$k - \forall$-universality	P	$O(n^3 \sigma)$

Algorithms.

Type	Length	Complexity
Counting	unrestricted	$O^*(n^4 k^3 2^\sigma)$
Counting	exactly m	$O^*(n^2 m^2 k 2^\sigma)$
Counting	at most m	$O^*(n^2 m^2 k 2^\sigma)$
Ranking	unrestricted	$O^*(n^4 k^3 2^\sigma)$
Ranking	exactly m	$O^*(n^2 m^2 k 2^\sigma)$
Ranking	at most m	$O^*(n^2 m^2 k 2^\sigma)$

Conclusion

Complexity.

Problem	Complexity Class	Best Algorithm
$k-\exists$-universality	NP-Complete	$O^*(n^3 2^\sigma)$
$k-\forall$-universality	P	$O(n^3\sigma)$

Algorithms.

Type	Length	Complexity
Counting	unrestricted	$O^*(n^4 k^3 2^\sigma)$
Counting	exactly m	$O^*(n^2 m^2 k 2^\sigma)$
Counting	at most m	$O^*(n^2 m^2 k 2^\sigma)$
Ranking	unrestricted	$O^*(n^4 k^3 2^\sigma)$
Ranking	exactly m	$O^*(n^2 m^2 k 2^\sigma)$
Ranking	at most m	$O^*(n^2 m^2 k 2^\sigma)$

Thank you for listening!