On Computability of Data Word Functions Defined by Transducers

Léo Exibard [1, 2] Emmanuel Filiot [1] Pierre-Alain Reynier [2]

[1]Méthodes Formelles et Vérification, Université libre de Bruxelles
[2]Laboratoire d'Informatique et Systèmes, Aix-Marseille Université

Context

This work is part of a line of research which aims at extending existing results in the field of synthesis to the realm of data words, i.e. words over a (slightly) infinite alphabet [Bojanczyk, 2018]. In Dave, Filiot, Krishna, and Lhote, 2020, the authors characterised computability for regular functions. We here study the case of functions defined by Nondeterministic Register Transducers (NRT), which are an extension of Register Automata (introduced as Finite Memory Automata in Kaminski and Francez, 1994) modelling relations over data words.

Motivation: Synthesis

The Synthesis Problem

Input: the specification $S \subseteq \mathrm{In} \times \mathrm{Out}$ of the behaviour of a program
→ *What* should be done

Output: a machine M having such behaviour
→ *How* it should be done

In is a set of inputs
Out is a set of outputs

Formally, M has to be such that
for each input $i \in \mathrm{dom}(S)$, $(i, M(i)) \in S$.

Functionality

Here, we study the case where the graph of S is a *function*.

Computability

The Computability Problem

Input: the specification of a function $f : \mathrm{In} \to \mathrm{Out}$

Output: an algorithm which computes f

Computability for Functions Defined over Infinite Words

$f : \Sigma^\omega \to \Gamma^\omega$ is *computable* if there exists a deterministic Turing Machine M which,
on reading longer and longer prefixes of the input w,
produces longer and longer prefixes of the output $f(w)$.

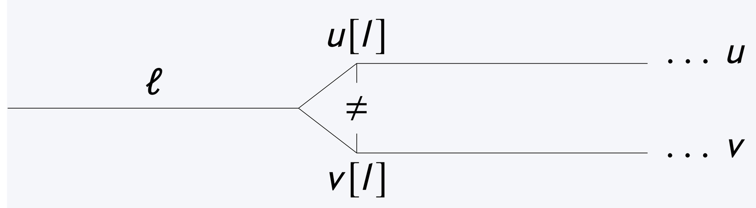
→ If $M(u, k)$ denotes the content of the output tape when the input reading head goes past position k, we have for all $k \in \mathbb{N}$, $M(u, k) \preceq f(u)$ and $\|M(u, k)\| \xrightarrow[k \to +\infty]{} +\infty$

$\preceq$ denotes the prefix relation
$\|w\|$ is the length of the word w.

Continuity

Cantor Distance

For $u, v \in A^\omega$, $d(u, v) = \begin{cases} 0 \text{ if } u = v \\ 2^{-\|u \wedge v\|} \text{ otherwise} \end{cases}$ where $u \wedge v$ is the longest common prefix ℓ of u and v



Continuous Function

$f : \Sigma^\omega \to \Gamma^\omega$ is *continuous at* u if:

$$\forall i \geq 0, \exists j \geq 0, \forall v \in \mathrm{dom}(f), \|u \wedge v\| \geq j \Rightarrow \|f(u) \wedge f(v)\| \geq i$$

f is *continuous* if it is continuous at every point u of its domain.

$f : w \mapsto \begin{cases} a^\omega \text{ if } |w|_a = \infty \\ b^\omega \text{ otherwise} \end{cases}$ is not continuous.

Computability and Continuity

Computability ⇒ Continuity

→ Always holds

Continuity ⇒ Computability

→ Does not hold in general.

Consider for instance the constant (hence continuous) function $f : u \mapsto h$ where for all $i \in \mathbb{N}$, $h[i] = 1$ iff the i-th Turing Machine halts (0 otherwise): f is not computable.

However:

Regular Functions

Theorem [Dave et al., 2020]: For regular functions, i.e. functions recognised by MSO transductions, computability and continuity *coincide* and are *decidable* in PTime.

Remark: regular functions are equivalently recognised by two-way transducers with regular lookahead, and again equivalently by 1way transducers with registers, a.k.a. streaming string transducers.

Main Result: Extension to the Realm of Data Words

For functions defined by Nondeterministic Register Transducers, computability and continuity again **coincide** and are **decidable** in PSpace.

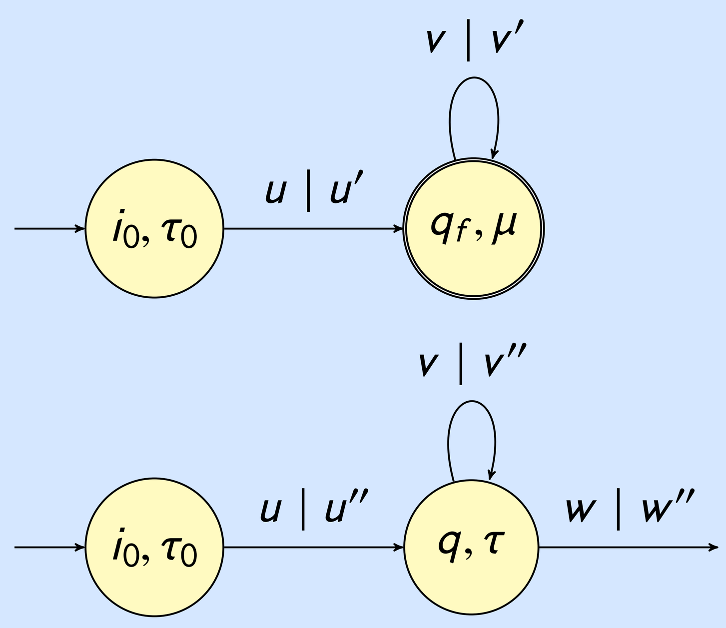
Non-continuity is characterised by the following pattern:

A function f_T realised by a transducer T is *not continuous* iff there exists

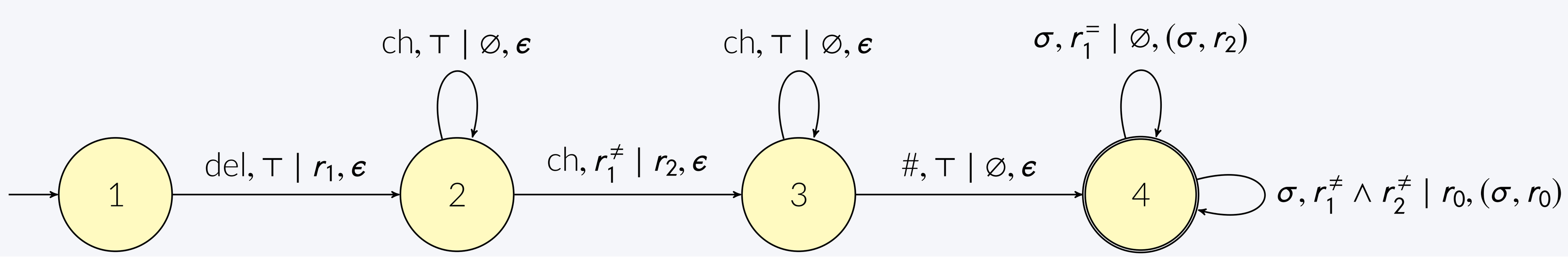
- an initial configuration (i_0, τ_0),
- an accepting configuration (q_f, μ) and a configuration (q, τ),
- finite input data words u, v, finite output data words u', v', u'', v'' admitting runs as depicted on the right, and
- an infinite input data word w admitting an accepting run from configuration (q, τ) producing output w'',

such that $\begin{cases} \mathrm{mismatch}(u', u'') \\ \textit{or} \\ v'' = \varepsilon \textit{ and } \mathrm{mismatch}(u', u''w'') \end{cases}$

$\mathrm{mismatch}(x, y)$ holds when there exists $i \in \{1, \ldots, \min(\|x\|, \|y\|)\}$ such that $x[i] \neq y[i]$



Example of a Nondeterministic Register Transducer



Setting: we deal with logs of communications between a set of clients. A log is an infinite sequence of pairs consisting of a tag in Σ, and the identifier of the client delivering this tag, modelled as an integer.
For a given client that needs to be modified, each of its messages should now be associated with some new identifier. The transformation has to verify that this new identifier is indeed free, *i.e.* never used in the log.
Before treating the log, the transformation receives as input the id of the client that needs to be modified (associated with the tag del), and then a sequence of identifiers (associated with the tag ch), ending with #.
This transducer is non-deterministic as it has to guess which of these identifiers it can choose to replace the one of the client.

Data Words

Sequences of pairs $(a, d) \in \Sigma \times \mathcal{D}$

- Σ finite alphabet of *labels*
- $\mathcal{D}$ infinite set of *data*

1	2	3	3	1	3	1	
req	req	grt	req	grt	grt	req	...

Register Transducers

Transitions are $q \xrightarrow{i, \varphi \mid \downarrow r_{\mathrm{in}}, w_o} q'$:

- $i \in \Sigma_{\mathrm{in}}$ is the input letter
- φ is the test conducted over the input data
- $r_{\mathrm{in}} \in R$ is the register where the input data is stored
- $w_o \in (\Sigma_{\mathrm{out}} \times R)^*$ is a finite sequence of pairs (o, r), meaning each label o is output along with the content of r.

Additional Results: Test-Free NRT

→ A Nondeterministic Register Transducer is *test-free* if for all its transitions, it conducts no test over the input data, i.e. $\varphi = \top$.

The following problems are in polynomial time for this subclass:

- Functionality
- Continuity
- Equivalence

Additional Results: the Class NRT_f

NRT_f denotes the class of functions defined by Nondeterministic Register Transducers.

- It is decidable in PSpace if a given relation $S \in \mathrm{NRT}_f$ (functionality problem)
- NRT_f is closed under composition
- It is decidable in PSpace whether two functions in NRT_f coincide on the intersection of their domain

Future Work

- Equip NRT with nondeterministic reassignment, i.e. the ability to *guess* the content of a register
- Generalise to the case where data are linearly ordered: $(\mathbb{Q}, <)$ and $(\mathbb{N}, <)$
  ($(\mathbb{N}, <)$ is harder because it is not *oligomorphic*)
- Synthesise from a specification which is not functional (already difficult in the finite alphabet case)

References

Bojanczyk, M. (2018). *Slightly Infinite Sets*. Online at https://www.mimuw.edu.pl/bojan/paper/atom-book.

Dave, V., Filiot, E., Krishna, S. N., & Lhote, N. (2020). Synthesis of Computable Regular Functions of Infinite Words. In I. Konnov & L. Kovács (Eds.), *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)* (Vol. 171, 43:1–43:17). LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Kaminski, M. & Francez, N. (1994). Finite-memory automata. *Theor. Comput. Sci. 134*(2), 329–363.

Paper, slides and video