Machine Spaces: Axioms and Metrics

Jörg Zimmermann and Armin B. Cremers
Institute of Computer Science
University of Bonn, Germany
{jz, abc}@iai.uni-bonn.de

An axiom system for machine spaces is introduced. Time is modeled as a totally ordered monoid, thus allowing discrete, continuous, and transfinite time structures. A generalized metric on these machine spaces is defined, having values in a directedly ordered monoid. This enables the notion of ε-balls in machine space, which, for example, can be used to explore model risk and robustness in statistics.

1 Introduction

The notion of computability is inseparably linked to the notion of a machine, an object realizing the computational process. The specific machine chosen determines the set of computable functions and influences complexity classes, i.e., sets of functions computable within a specific resource bound. Here we do not consider specific machine classes, but try to use the axiomatic method in order to investigate the notion of an abstract machine and its implied set of computable functions from a general point of view.

2 Time Axioms

Time is the driver of the computational processes, so we want to isolate the abstract properties of time which are relevant from a computational point of view. Here we model time as a totally ordered monoid, i.e., all time elements can be compared and be combined by an associative operation, denoted by '+', which has a neutral element 0_t. Furthermore, the order structure and the algebraic structure are compatible:

(Compatibility) $\forall t_1, t_2, t_3 : t_1 \leq t_2 \Rightarrow t_1 + t_3 \leq t_2 + t_3$ and $t_3 + t_1 \leq t_3 + t_2$.

Compatibility is required in both possible ways because the monoid operation has not to be commutative (which is, for example, the case for the sum of ordinal numbers).

3 Machine Axioms

Machines operate on a state space Σ, about which we want to say nothing specific here. Further we introduce the input space I, output space O, and program space P. An *initializer* is a mapping *init* from $P \times I$ to Σ and an *output operator* is a mapping *out* from Σ to O. A machine wrt. a time structure T and a state space Σ is a mapping M from $\Sigma \times T$ to Σ, denoted by $M_t(s)$. Finally, there is a subset $HALT$ of Σ. States in $HALT$ will be used to signal termination of a computation. Let $TERM(M,s)$ denote the set of time points t with $M_t(s) \in HALT$. Now we require the following axioms:

(Action) $\forall t_1, t_2 \in T : M_{t_1+t_2} = M_{t_2} \circ M_{t_1}$.

Submitted to:
MCU 2013

(Start) $\forall s \in \Sigma : M_{0_t}(s) = s$ (i.e., $M_{0_t} = id_\Sigma$).

These two axioms states that the time monoid is operating on the state space via machine M.

(Stop) $\forall s \in \Sigma, t_1, t_2 \in T : t_1 \in TERM(M,s)$ and $t_1 \leq t_2 \Rightarrow M_{t_1}(s) = M_{t_2}(s)$.

That is, after reaching a termination state, nothing changes anymore, i.e., termination states are fixpoints of the machine dynamics.

(Well-Termination) $\forall s \in \Sigma : TERM(M,s) \neq \emptyset \Rightarrow \exists t_1 \in TERM(M,s)\ \forall t_2 \in TERM(M,s) : t_1 \leq t_2$.

Well-termination requires that if a machine terminates on s, i.e., reaches $HALT$ for some point in time, then there is a first point in time when this happens, or, equivalently, the set of time points taking s to $HALT$ has a least element, if it is not empty. If $TERM(M,s)$ is non-empty and t_1 its least element, then define $M(s) := out(M_{t_1}(s))$, otherwise $M(s)$ is left undefined.

Definition: A function $f : I \to O$ is *implemented* by $p \in P$ on M iff $f(x) = M(init(p,x))$ for all $x \in I$.

Functions f which are implementable on a machine M are called "M-computable". $[p]_M$ denotes the function implemented by p on M.

4 A Metric on Machine Spaces

Metrics are a fundamental tool to organize sets. They enable an intuitive understandig of the relationship between objects and a form of geometric reasoning about the objects, e.g. by using the triangle inequality. We now want to introduce a metric on machine spaces which represents some kind of distance between different machines by relating the resources needed by the machines to solve the same problems. Here we focus on time complexity as our resource measure:

Definition: $time_p^M(x) = min(TERM(M, init(p,x)))$.

Definition: $\tau : T \to T$ is an *admissible time transfer function (attf) from M_1 to M_2* iff τ is monotone and

$\forall p_1 \in P_1 \exists p_2 \in P_2 : [p_1]_{M_1} = [p_2]_{M_2}$ and $\forall x \in I : time_{p_2}^{M_2}(x) \leq \tau(time_{p_1}^{M_1}(x))$.

Two machines M_1 and M_2 are time-compatible if they operate on the same time structure, input space and output space. Note that the state space and program space have not to be the same. Next we define a generalized metric Δ_t which takes two time-compatible machines and maps them on an element of $\mathscr{P}(T^T)$, i.e., the distance is represented by a set of functions on the time structure T:

$\Delta_t(M_1, M_2) = \{\tau | \tau$ is an attf from M_1 to $M_2\}$.

We use the whole set of attfs, because there seems to be no obvious way to single one out. Additionally, one can combine and compare sets of functions much like single functions. However, future research may identify a canonical element in the set of attfs, maybe by introducing additional axioms.

Definition: Let $\alpha, \beta \subseteq T^T$. Then define their composition as $\alpha \circ \beta := \{\tau_1 \circ \tau_2 | \tau_1 \in \alpha, \tau_2 \in \beta\}$.

Definition: $\alpha \leq \beta$ iff $\forall \tau_2 \in \beta \exists \tau_1 \in \alpha : \tau_1 \leq \tau_2$.

With these definitions the sets of attfs become a directedly orderd monoid (dom). A directed order is a partial order where two elements always have an upper bound, and directed monoids can be used as ranges for generalized metrics, allowing many standard constructions of topology [3]. Let $< \tau >$ denote the upward closure of an attf, then $< id_T >$ is the neutral element of this monoid, also denoted by 0_T. Our metric can then be classified as a dom-valued directed pseudometric, i.e., it non-positive and asymmetric (directed), and not strict, i.e., $\Delta_t(M_1, M_2) = 0_T$ does not imply $M_1 = M_2$. But machines have self-distance zero and satisfy the following triangle inequality:

$$\Delta_t(M_1, M_3) \leq \Delta_t(M_2, M_3) \circ \Delta_t(M_1, M_2)$$

Machine models play an important part in many fields of science, not always explicitly acknowledged. For example, the choice of a reference machine for defining the complexity of statistical models is central for approaches to general induction. It would be interesting to evaluate the implications of choosing a specific reference machine by analysing how sensitive the drawn conclusions are to variations of the reference machine. The concept of ε-balls in machine space could lead to a systematic way to facilitate such robustness analyses, thus addressing the vital problem of "model risk", i.e., the problem of unreliable conclusions which are very sensitive to the chosen model class. Also, we hope that advances in machine theory will lead to a "standard reference machine", a default reference machine for which there is a consensus that it takes a canonical position in machine space.

The proposed ideas and concepts are based on work started and continued in [1], [5], [6], [2], and [4].

References

[1] R. Gandy (1980): *Church's Thesis and principles for mechanisms*. In J. Barwise, H. J. Keisler & K. Kunen, editors: *The Kleene Symposium*, North-Holland, pp. 123–148, doi:10.1016/S0049-237X(08)71257-6.

[2] R. Glaschick (2012): *A size index for multitape Turing machines*. *Technical Report NI12061-SAS, Isaac Newton Institute for Mathematical Science, Cambridge, UK*.

[3] R. Kopperman (1988): *All topologies come from generalized metrics*. *The American Mathematical Monthly* 95(2), pp. 89–97, doi:10.2307/2323060.

[4] T. Neary, D. Woods, N. Murphy & R. Glaschick (2013): *Wang's B machines are efficiently universal, as is Hasenjaeger's small universal electromechanical toy*. Available at `http://arxiv.org/abs/1304.0053`.

[5] M. B. Pour-El & I. Richards (1982): *Noncomputability in models of physical phenomena*. *International Journal of Theoretical Physics* 21(6-7), pp. 553–555, doi:10.1007/BF02650184.

[6] W. Sieg (2008): *Church Without Dogma: Axioms for Computability*. In S. Barry Cooper, Benedikt Löwe & Andrea Sorbi, editors: *New Computational Paradigms*, Springer, pp. 139–152, doi:10.1007/978-0-387-68546-5_7.