ON THE EXTREME POWER OF NONSTANDARD PROGRAMMING LANGUAGES

TAKUMA IMAMURA

Abstract. Suenaga and Hasuo introduced a nonstandard programming language \texttt{While} which models hybrid systems. We demonstrate why \texttt{While} is not suitable for modeling actual computations.

Suenaga and Hasuo \cite{SuenagaHasuo2014} introduced an imperative programming language \texttt{While}, which is the usual \texttt{While} programming language equipped with a positive infinitesimal \texttt{dt}. This language is intended to hyperdiscretise hybrid systems and enable them to be formally verified by Hoare logic \cite{Hoare1969}. On the other hand, this language is not intended to be a model of actual computation. We demonstrate why \texttt{While} is not suitable for modeling actual computations. The main reason is that \texttt{While} has too strong computational power. We clarify the causes of the power.

We refer to Suenaga and Hasuo \cite{SuenagaHasuo2014} for the definition of \texttt{While}; Robinson \cite{Robinson1960} for nonstandard analysis; Shen and Vereshchagin \cite{ShenVereshchagin2005} for computability theory.

The first cause of the power is that \texttt{While} is furnished with the constant symbols \texttt{c\_r} for all real numbers \texttt{r \in R} and the exact comparison operator \texttt{<}. They bring much strong computational power to this language as we will see below.

**Lemma 1.** \texttt{While} computes the floor function on \( {}^\ast \mathbb{R} \).

**Proof.** The following program computes the floor function.

```pseudocode
Input: x
Output: y
n := 0;
while ¬(n ≤ x < n + 1 ∨ −n ≤ x < −n + 1) do
    n := n + 1;
if x ≥ 0
    then y := n
else y := −n
```

\( □ \)

**Remark 2.** The floor function is a typical example of a noncomputable real function (see Weihrauch \cite{Weihrauch2000} p. 6).

**Proposition 3.** \texttt{While} computes every standard decision problem on \( {}^\ast \mathbb{N} \).

**Proof.** Let \( A \subseteq \mathbb{N} \). Set \( r = \sum_{i=0}^{\infty} 3^{-i} \chi_A (i) \), where \( \chi_A \) is the characteristic function of \( A \). The constant \( r \) has complete information deciding the membership of \( A \).

2000 Mathematics Subject Classification. 34A38; 68Q10; 03H10.

Key words and phrases. hybrid systems; hypercomputation; nonstandard analysis.
Consider the following program.

\[
\begin{align*}
\text{Input : } &\ x \\
\text{Output : } &\ y \\
t &:= r; \\
\text{while } x \neq 0 \text{ do} \\
a &:= 3 \cdot a; \\
x &:= x - 1; \\
u &:= a - 3 \cdot \lfloor (1/3) \cdot a \rfloor; \\
\text{if } u \geq 1 \text{ then} \\
y &:= 1 \\
\text{else} \ y := 0
\end{align*}
\]

This computes the characteristic function $\chi_{\ast A}$ of $\ast A$ for all (standard and nonstandard) inputs. \qed

The second cause is that $\text{While}^{dt}$ can execute infinitely many steps of computation whose computational resource consumption (such as time, space and electricity usage and heat generation) is $\gg 0$.

**Proposition 4.** $\text{While}^{dt}$ computes every $0'$-computable function on $\mathbb{N}$.

**Proof.** Let $f : \mathbb{N} \to \mathbb{N}$ be $0'$-computable. By Schoenfield's limit lemma (see Theorem 48 of [8]), there is a computable function $F : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $f = \lim_{s \to \infty} F(s, -)$. Obviously $\text{While}^{dt}$ computes $F$ for all inputs (with no use of noncomputable real numbers). Consider the following program:

\[
\begin{align*}
\text{Input : } &\ x \\
\text{Output : } &\ y \\
y &:= F(\infty, x)
\end{align*}
\]

This computes the limit function $f$ for all standard inputs. \qed

**Remark 5.** The infinity constant $\infty$ can be eliminated as follows.

\[
\begin{align*}
t &:= 0; \\
u &:= 0; \\
\text{while } t < 1 \text{ do} \\
t &:= t + dt; \\
u &:= u + 1
\end{align*}
\]

The variable $u$ is infinite after executing this program. The while loop is repeated an infinite number of times. The instruction $u := u + 1$ in the loop consumes computational resource $\gg 0$ in each execution.

Hence some restrictions are needed to metamorphose $\text{While}^{dt}$ into a model of actual hybrid computation. The same phenomena we have shown also occur in other nonstandard “models of computation” such as $\text{Sproc}^{dt}$ (Suenaga et.al. [8]), NSF (Nakamura et.al. [4], [3]) and the internal Turing machines (Loo [2]).

**References**

[1] C. A. R. Hoare. An Axiomatic Basis for Computer Programming. *Communications of the ACM*, 12(10):576–580, 583, 1969.
[2] Ken Loo. Internal Turing Machines, 2004. arXiv:0407056.
[3] Hirofumi Nakamura. A functional programming language with infinitely small numbers. Master’s thesis, Kyoto University, 2018.
[4] Hirofumi Nakamura, Kensuke Kojima, Kohei Suenaga, and Atsushi Igarashi. A Nonstandard Functional Programming Language. In APLAS 2017, pages 514–533, 2017.
[5] Abraham Robinson. Non-standard Analysis. North-Holland, 1966.
[6] A. Shen and N. K. Vereshchagin. Computable Functions. American Mathematical Society, 2003.
[7] Kohei Suenaga and Ichiro Hasuo. Programming with Infinitesimals: A While-Language for Hybrid System Modeling. In ICALC 2011, pages 392–403, 2011.
[8] Kohei Suenaga, Hiroyoshi Sekine, and Ichiro Hasuo. Hyperstream Processing Systems: Non-standard Modeling of Continuous-Time Signals. In POPL 2013, pages 417–430, 2013.
[9] Klaus Weihrauch. Computable Analysis. Springer, 2000.

Research Institute for Mathematical Sciences, Kyoto University, Kitashirakawa Oiwake-cho, Sakyo-ku, Kyoto 606-8502, Japan
E-mail address: timamura@kurims.kyoto-u.ac.jp