Extraction of real Gray-code from proofs using non-deterministic implication

Ulrich Berger
Swansea University

j.w.w.

Hideki Tsuiki
Kyoto University

Workshop on Mathematical Logic and its Aplications

September 17, Kyoto

Overview

- Real numbers as streams: Signed digits and Gray code
- Partiality and non-determinism
- Coinductive representation of Signed digits and Gray code via realizability
- Capturing partiality and non-determinism logically
- Conclusion and future work

Signed digit representation

Let $\mathbb{I} = [-1, 1] \subseteq \mathbb{R}$ and $\mathrm{SD} = \{-1, 0, 1\}$.

A **signed digit representation** of a real number $x \in \mathbb{I}$ is a stream $d_0 : d_1 : \ldots \in \mathrm{SD}^\omega$ such that

$$x = \sum_{i \in \mathbb{N}} d_i * 2^{-(i+1)}$$

All reals in $\mathbb{I}$ have signed digit representations, most of them uncountably many.

The signed digit representation is admissible in the sense of TTE. In particular:

> Exactly the computable reals have a computable signed digit representation.
>
> A function on the reals is computable iff it can be computed on signed digit representations.

Gray code

Gray code (named after Frank Gray in 1946 who called it “reflected binary code”) is an alternative to the binary representation of natural numbers where neighbouring numbers differ in only one digit.

Tsuiki extended this to a representation of real numbers.

H. Tsuiki: Real Number Computation through Gray Code Embedding. TCS 284, 2002.

A similar representation was studied in

P. D. Gianantonio: An abstract data type for real numbers. TCS 221, 1999.

Tsuiki's partial Gray code for real numbers

The **Gray code** or **Gray representation** of $x \in [-1, 1]$ is the itinerary of the tent map $\mathrm{t}(x) = 1 - 2|x|$. This means that the n-th digit is 0 resp. 1 if $t^n(x) < 0$ resp. > 0.

If $t^n(x) = 0$, then the n-th digit is undefined.

Remarkably, **every real in** $[-1, 1]$ **has a unique Gray code**.

Like the signed digit representation Gray code is admissible, though not exactly in the sense of TTE where only total streams are considered as representations.

One easily sees that at most one digit of the Gray code can be undefined. Therefore, computation with the Gray code can be modeled by a *Two-Head-Turing-Machine*.

From Gray code to signed digits (Tsuiki)

Set `neg 0 = 1; neg 1 = 0`

```
gtos (0:g)       = -1 : gtos g
gtos (1:a:g)     =  1 : gtos (neg a : g)
gtos (a:1:c:g) =  0 : gtos (a : neg c : g)
```

The pattern of the third line overlaps with those in the first and second line and the digit a may be undefined.

Therefore, this program is non-deterministic and will not be correctly executed in a functional language such as Haskell.

From signed digits to Gray code (Tsuiki)

Set `nh(a:s) = neg a : s`

```
stog (-1:s) = 0 : stog s
stog ( 1:s) = 1 : nh(stog s)
stog ( 0:s) = a : 1 : nh g
                  where a : g = stog s
```

This function is not productive in the strict sense since in the third line the output of the first digit a is deferred to the recursive call.

Nevertheless the output is a correct Gray code (which may be undefined at one point).

Logic for exact real number computation

We are looking for a logical system such that:

- Data representation such as signed digits or Gray code can be obtained as realizers of coinductive predicates.
- Partial and non-deterministic function such as `stog` and `gtos` can be extracted from proofs.

Related work: Realizing Gray code deterministically

B., Kenji Miyamoto, Helmut Schwichtenberg, Hideki Tsuiki: Logic for Gray-code computation (submitted)

gives a realizability interpretation and Minlog implementation of an intensional version of Gray code, called pre-Gray code, using a conventional constructive system and conventional program extraction.

This skirts the issue of non-determinism at the price of giving up the uniqueness of Gray code.

An earlier version of this talk was presented at CSL 2016.

B. Extracting Non-Deterministic Concurrent Programs. LIPIcs 26, 2016.

Intuitionistic Fixed Point Logic (IFP)

- Intuitionistic many-sorted logic in finite types.
- Sorts represent abstract mathematical structures given by $\vee$-free axioms.
- Inductive and coinductive definitions of predicates as least and greatest fixed points of monotone predicate transformers.
- Realizers are untyped recursive programs.
- The definition of realizability is usual except that quantifiers are interpreted uniformly:
  - $a\,\mathbf{r}\,\exists x\, A(x)$ means $\exists x\,(a\,\mathbf{r}\,A(x))$.
  - $a\,\mathbf{r}\,\forall x\, A(x)$ means $\forall x\,(a\,\mathbf{r}\,A(x))$.

Real and natural numbers

The structure of the *real numbers* $\mathbb{R} = (0, 1, +, *, -, /, <, |\cdot|)$ is treated as a sort specified by $\vee$-free axioms.

The *natural numbers* are defined as the least subset of $\mathbb{R}$ that contains 0 and is closed under successor:

$$\mathbb{N} \stackrel{\mu}{=} \{\, 0 \,\} \cup \{\, x + 1 \mid x \in \mathbb{N} \,\}$$

Realizability automatically associates with this definition the unary representation of natural numbers and with proofs of closure properties of $\mathbb{N}$ programs operating on that representation.

Signed digits and Gray code in logical form

Signed Digit representation

$\mathrm{C}(x) \overset{\nu}{=} \exists d \in \mathrm{SD} \,.\, x \in \mathbb{I}_d \wedge \mathrm{C}(2x - d)$

where $\mathbb{I}_d := [d/2 - 1/2, d/2 + 1/2]$ and $\overset{\nu}{=}$ means 'largest'.
Note that $x \in \mathbb{I}_{-1}$ iff $x \leq 0$, $x \in \mathbb{I}_1$ iff $x \geq 0$, $x \in \mathbb{I}_0$ iff $|x| \leq 1/2$.

- $s \,\mathbf{r}\, \mathrm{C}(x)$ iff s is a signed digit representation of x.

Gray code

$\mathrm{G}(x) \overset{\nu}{=} (x \neq 0 \to x \leq 0 \vee x \geq 0) \wedge \mathrm{G}(\mathrm{t}(x))$

- $s \,\mathbf{r}\, \mathrm{G}(x)$ iff s is a Gray code of x.

We want to show constructively $\mathrm{C} = \mathrm{G}$ which will give us programs translating between the two representations.

Is it possible? Will we get `gtos` and `stog`?

Non-Determinism in logical form

In order to prove $\mathrm{C} = \mathrm{G}$ constructively we extend IFP by

- Non-Deterministic implication: $A \mid B$ (read "A if B")
- Archimedean Induction (AI), a computationally meaningful version of the Archimedean property.

Non-Deterministic implication

Classically, $A \mid B$ is the same as implication, $B \to A$.

Constructively, $A \mid B$ is distinguished by a non-deterministic realizabilty interpretation:

$$a\,\mathbf{r}\,(A \mid B) := (\mathbf{r}(B) \to \mathrm{Def}(a)) \land a\,\mathbf{fr}\,A$$

where

$$\begin{array}{rcl}
\mathbf{r}(B) & := & \exists b\, b\,\mathbf{r}\,B \qquad (\text{“}B \text{ is realizable”}) \\
\mathrm{Def}(a) & \overset{\mu}{=} & (\exists b\ a = \mathrm{Res}(b)) \lor \\
 & & \exists b, c\ a = \mathrm{Amb}(b, c) \land (\mathrm{Def}(b) \lor \mathrm{Def}(c)) \\
a\,\mathbf{fr}\,A & \overset{\nu}{=} & \forall b\,(a = \mathrm{Res}(b) \to b\,\mathbf{r}\,A) \land \\
 & & \forall b, c\,(a = \mathrm{Amb}(b, c) \to b\,\mathbf{fr}\,A \land c\,\mathbf{fr}\,A)
\end{array}$$

$\mathrm{Amb}(b, c)$ means non-deterministic choice between a and b

Realizable laws

$A \to A \mid B$

$A \mid \bot$

$(B \to A_0 \vee A_1) \to (\neg B \to A_0 \wedge A_1) \to (A_0 \vee A_1) \mid B$,

if A_1, A_1, B are non computational

$A \mid B \to (A \to A' \mid B) \to A' \mid B$

$A \mid B \to (B' \to B) \to A \mid B'$

$A \mid B \to A \mid B' \to A \mid \neg\neg(B \vee B')$

$A \mid B \to B \to A$ if A is non computational

Nondeterminism monad

$$\mathrm{S}(A) \quad := \quad A \mid \mathrm{True}$$

Derived laws

$A \to \mathrm{S}(A)$

$\mathrm{S}(A) \to (A \to \mathrm{S}(A')) \to \mathrm{S}(A')$

$\mathrm{S}(A) \to A$ if A is non computational

$A \mid B \to A \mid \neg B \to \mathrm{S}(A)$

Nondeterministic signed digits and Gray code

$$\mathrm{C}(x) \overset{\nu}{=} \mathrm{S}(\exists d \in \mathrm{SD}\,.\, x \in \mathbb{I}_d \wedge \mathrm{C}(2x - d))$$

$$\mathrm{G}(x) \overset{\nu}{=} ((x \leq 0 \vee x \geq 0) \mid x \neq 0) \wedge \mathrm{G}(\mathrm{t}(x))$$

Theorem
$\mathrm{G} \subseteq \mathrm{C}$

The extracted program is `gtos`.

Archimedean Induction

Archimedean Property

Version 1	$\forall x \exists n \in \mathbb{N}\, x \le 2^n$	not realizable
Version 2	$\forall x\, (\forall n \in \mathbb{N}\, x \le 2^{-n} \to x = 0)$	non computational

Archimedean Induction (AI)

$$\frac{\forall x \neq 0\ ((|x| \le 1/2 \to A(2x)) \to A(x))}{\forall x \neq 0\ A(x)}$$

(AI) is realized by the fixed point combinator, i.e. if f realizes the premise, then the least fixed point of f realizes the conclusion.

From signed digits to Gray code

Theorem (AI)

$\mathrm{C} \subseteq \mathrm{G}$

Proof.

By coinduction. To show:

(1) $\mathrm{C}(x) \to x \neq 0 \to x \leq 0 \vee x \geq 0$

(2) $\mathrm{C}(x) \to \mathrm{C}(\mathrm{t}(x))$

Part (1) can be shown by Archimedean Induction.

Part (2) can be shown by coinduction.

□

Extracted program (from proof of $\mathrm{C} \subseteq \mathrm{G}$)

```
t s = l1 s : t (l2 s)
```

where `l1` is extracted from (1) $\mathrm{C}(x) \to x \neq 0 \to x \leq 0 \vee x \geq 0$
and `l2` is extracted from (2) $\mathrm{C}(x) \to \mathrm{C}(\mathrm{t}(x))$

```
l1 (-1:s) = 0          l2 (-1:s) = s
l1 ( 1:s) = 1          l2 ( 1:s) = - s
l1 ( 0:s) = l1 s       l2 ( 0:s) = 1 : l2 s
```

Analysis:

```
t (-1:s) = 0 : t s
t ( 1:s) = 1 : t (- s) = 1 : nh (t s)    (easy to see)
t ( 0:s) = l1 s : t (1 : l2 s)
         = l1 s : 1 : nh (t (l2 s))
         = head(t s) : 1 : nh (tail(t s)  (by def of t)
         = a : 1 : nh g   where a : g = t s
```

Hence `t = stog`.

Conclusion and further work

- We have shown that it is possible to capture nondeterminism logically through realizability and program extraction

- Computing with other models of real numbers such the interval domain requires nondeterminism as well (Potts, Edalat, Escardo, 1997). In fact, nondeterminism is unavoidable (Escardo, Hofmann, Streicher, 2004).

- To do:
  - extract `stog` directly from a suitable proof (without transformation),
  - implement the system and evaluate the performance of memoized computation with Gray code.

- Related project: Logic for extracting imperative programs (j.w.w. Gregory Woods).