Compact Linear Types

John Skaller

November 25, 2022

1 The Ring of Power

1.1 Preliminary definitions

Definition 1 *A* semi-group *is a set together with an associative binary operation; that is*

$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$

where infix $\cdot$ *is taken as the symbol for the binary operation.*

Associativity is a crucial property for an operator because it allows concurrent evaluation of arbitrary subsequences of a sequence of operands. For example given the operand expression

$$(((1 \cdot 2) \cdot 3) \cdot 4) \cdot 5$$

we can compute $1 \cdot 2$ and $4 \cdot 5$ concurrently:

$$(1 \cdot 2) \cdot 3 \cdot (4 \cdot 5)$$

then then combine 3 to either the LHS or RHS subterm before performing the final combination. Another way of looking at this property is that the values to be combined can be stored in the leaves of a tree and any bottom up visitation algorithm can be used to find the total combination.

Associativity means you can add or remove (balanced pairs of) parentheses freely. In particular it is common practice to leave out the parentheses entirely.

Definition 2 *A* monoid *is a semigroup with a unit* u*, that is*

$$x \cdot u = x \text{ and } \mathrm{u} \cdot \mathrm{x} = \mathrm{x}$$

for all x *in the set.*

The existence of a unit means you can freely add or remove units from anywhere in your computation.

Definition 3 *A* group *is a monoid in which every element has an inverse, that is, for all x there exists an element y such that*

$$x \cdot y = u \text{ and } y \cdot x = u$$

where u is the unit of the underlying monoid. For integers of course, the additive inverse of a value is it's negation.

Definition 4 *An operation is* commutative *if the result is the same with the operands reversed, that is, for all a and b.*

$$a \cdot b = b \cdot a$$

A group is said to be commutative if the group operation is commutative.

Commutativity says you can switch the order of children in the tree representation of an expression.

If an operation is also associative, commutative, and has a unit, then the operation is well defined on a set of operands, taking the operation on the empty set to be the unit.

This means irrespective of what data structure you use to hold the values to be combined, and what algorithm you use to scan them, provided you visit each value exactly once, the result of the operation on them is invariant.

Definition 5 *A* ring *is a set with two operations denoted by $+$ and $*$ such that the set with $+$ is a group, and the set excluding the additive unit is a monoid, and the following rule, called the* distributive law *holds for all a, b and c*

$$a * (b + c) = a * b + a * c$$

If the multiplication operation is commutative then it is called a commutative ring.

1.2 The rings $\mathbb{N}_n$

Definition 6 *Let $\mathbb{N}_n$ be the subrange of the integers $0..n-1$ with addtion, subtraction, multiplication, division and remainder defined as the natural result modulo n. Then $\mathbb{N}_n$ is a commutative ring called a* natural ring.

The usual linear order is also defined. Negation is defined by

$$-x = n - x$$

Natural computations prior to finding the modular residual present an issue we resolve by performing these computations in a much larger ring.

Definition 7 *The* size *of a finite ring R, written $|R|$, is the number of values of the underlying set.*

1.3 Representation

Lemma 1 *The C data types*

```
uint8_t uint16_t uint32_t uint64_t
```

with C builtin operations for addition, subtraction, negation, and multiplication are the rings $\mathbb{N}_{2^8}$ $\mathbb{N}_{2^{16}}$ $\mathbb{N}_{2^{32}}$ $\mathbb{N}_{2^{64}}$ *respectively, with the usual comp[arison operations, unsigned integer division, and unsigned integer modulus.*

Theorem 1 Representation Theorem. *The values of a ring* $\mathbb{N}_n$ *can be represented by values of a ring* $\mathbb{N}_{n^2}$ *and the operations addition, substraction, negation multiplication and modulus computed by the respective operations modulo* n*. Comparisions work without modification.*

In particular we can use `uint64_t` to represent rings of index up to 2^{32}.

2 Ring Products

Definition 8 *Let* R_i *for* i *in* $\mathbb{N}_n$ *be a tuple of* n *rings, then the* tensor product *of the rings, denoted by*

$$R_0 \otimes R_1 \otimes ... \otimes R_{n-1}$$

is a ring with values tuples of corresponding elements, operations defined componentwise, comparisons defined by the usual lexicographic ordering, and iterators sequencing through values in the defined order.

The size of the ring is the product or the ring sizes.

Definition 9 *A ring is compact linear if it is a natural ring, a product of compact linear rings, or a sum of compact linear rings.*

Theorem 2 Compact Linear Product Representation. *A compact linear product can be represented by a single value* $0..N-1$ *where* N *is the product of the sizes of the rings. The encoding of a value* $(v_0, v_1, ..., v_{n-1})$ *is given by*

$$v_0 * r_0 + v_1 * r_1 + ... + r_{n-1} * v_{n-1}$$

where $r_{n-1} = 1$ *and* r_k *for* k *in* $0..n-2$ *is the product of the sizes of the rings* R_j *for* $j > k$*:*

$$r_k = \prod_{j=k+1}^{n-1} |R_j|$$

where the empty product is 1. That is, the product of the sizes of the rings to the right *of ring* R_k *in the ring product formula.*

The projection p_k *of the* k'*th ring is given by*

$$v/r_k \mod |R_k|$$

where $|R|$ *is the size of the ring* R*.*

3 Ring Sums

Definition 10 *Let R_i for i in $\mathbb{N}_n$ be a tuple of n rings, then the* sum *of the rings, denoted by*

$$R_0 \oplus R_1 \oplus ... \oplus R_{n-1}$$

is a ring with values of one of the rings determined by the injection function that constructed the ring.

Theorem 3 Compact Linear Sum Representation. *A compact linear sum can be represented by a single value $0..N-1$ where N is the sum of the sizes of the rings. The injection of a value v_i of ring R_i into the sum is given by*

$$v_i + s_i$$

where $s_{n-1} = 0$ and r_k for k in $0..n-2$ is the sum of the sizes of the rings R_j for $j > k$.

$$s_k = \sum j = k + 1^{n-1}|R_j|$$

where the empty sum is 0. That is, the sum of the sizes of the rings to the right *of ring R_k in the ring sum formula.*

Decoding the sum is achieved as follows. Starting with $i = 0$, if $v >= s_i$, then the injection was for the value $v_i = v - s_i$ of the ring R_i, otherwise increment i and try again. Since for $i = n - 1$, the rightmost ring, $v >= 0$ is true for all possible v, the iteration must terminate.