Arithmetics in basis calculus

Research Notes in the ENEXA Project

September 4, 2025

We parametrize numbers by bits in fixed point representations, which are understood as categorical variables in a factored system representation.

1 Modular Calculus

We have two basic functions calculating the mod

$$q : \bigtimes_{k\in[d]} [m_k] \to [2] \quad \text{and} \quad q(x_{[d]}) = \sum_{k\in[d]} x_k \bmod 2$$

and the integer division by two

$$g : \bigtimes_{k\in[d]} [m_k] \to [2] \quad \text{and} \quad g(x_{[d]}) = \left\lfloor \frac{\sum_{k\in[d]} x_k}{2} \right\rfloor$$

2 Sums

Given the bit representations of summands, we want to calculate the bit representation of their sum.

2.1 Binary Addition

Basis calculus of binary additon is a TT architecture, where each core performs the addition of two bits and a carry bit, producing a sum bit and a carry bit.

Addition of two numbers with d bits:

- Bit variables of the first number: $X_{[d]}$
- Bit variables of the second number: $Z_{[d]}$
- Output bit variables: $Y_{[d+1]}$
- Carry bit variables: $C_{[d]}$, with $C_1 = 0$

The sum of any two numbers is represented by the boolean tensor

$$\tau\left[X_{[d]}, Z_{[d]}, Y_{[d+1]}\right] := \left\langle \{\epsilon_0\left[C_0\right], \delta\left[C_{d-1}, Y_d\right]\} \cup \bigcup_{k\in[d]} \{\beta^q\left[Y_k, X_k, Z_k, C_{k-1}\right], \beta^g\left[C_k, X_k, Z_k, C_{k-1}\right]\} \right\rangle \left[X_{[d]}, Z_{[d]}, Y_{[d+1]}\right] ,$$

where Y_k and C_k are the head variables of the basis encodings to q and g. If any only if for given indices $x_{[d]}, z_{[d]}, y_{[d+1]}$ we have $\tau\left[X_{[d]} = x_{[d]}, Z_{[d]} = z_{[d]}, Y_{[d+1]} = y_{[d+1]}\right] = 1$, then the by the indices $y_{[d+1]}$ represented number is the sum of the by $x_{[d]}, z_{[d]}$ represented numbers.

2.2 Generic construction

In general, when adding more than two variables, the carry bits need to be extended to a categorical variable with more than two states. Let $X^i_{[d]}$ be the d bits of the ith number, and let $X^{[n]}_{[d]}$ be all the bit variables (i.e. $n \cdot d$ many) of the n numbers. Then the same construction can be done as above, with cores

$$\beta^q \left[Y_k, X^{[n]}_k, C_{k-1} \right], \beta^g \left[C_k, X^{[n]}_k, Z_k, C_{k-1} \right]$$

Note that C_k now takes values in m_k where

$$m_k = \left\lfloor n \cdot \frac{m_{k-1}}{2} \right\rfloor .$$

Further, the result might have more than $d+1$ bits, so we need further basis encoding cores to q and g.

3 Products

Products of numbers are decomposable into sums involving two bit variables of the factors, that is

$$\sum_{k,\tilde{k} \in [d]} 2^{k+\tilde{k}} \cdot (X_k \wedge Z_{\tilde{k}}) .$$

Reordering the sum, we obtain

$$\sum_{r \in [2d-1]} 2^r \left(\sum_{k,\tilde{k} \in [d] \,:\, k+\tilde{k}=r} X_k \wedge Z_{\tilde{k}} \right) .$$

From this, it is obvious that the calculation can be performed in basis calculus with basis encodings of $\wedge, q, g$. The head variables of the $\wedge$ encoding are used as the summand variabled in q (output: bit of the product) and g (output: carry bit).

4 Application

Any of these tensor network schemes are considered batch schemes to perform arithmetic operations. Contractions of the representing basis encodings calculate the number of true input-output relations, given e.g. a restriction onto specific outputs and inputs (by adding subset encodings of the numbers of interest). One application is the countdown game, when in addition parametrizing the sum/negation operations with an additional selection variable.