Automated Witness Derivation for Polynomial Constraint Systems

Christian Reitwießner
chris@ethereum.org

Let $\mathbb{F}$ be a fixed prime field of some order larger than two. Unless stated otherwise, numbers are considered to be elements of that field.

1 Definitions

We are dealing the polynomial constraint system that can be seen as a table where some cells are pre-filled with input numbers and this initial content has to be extended to the whole table such that constraint polynomials evaluate to zero on every pair of adjacent rows. The formal definition follows.

Definition 1.1. *The tuple* $(\mathbb{F}, n, m, P, F, I, O)$ *is called a* polynomial constraint system*, where*

- $\mathbb{F}$ *is a finite field of prime order (which is often omitted),*
- $n \in \mathbb{N}$ *is called the* number of columns*,*
- $m \in \mathbb{N}$ *is an integer (usually a power of two), called the* number of rows*,*
- P *is a set of polynomials in* $2n$ *variables over* $\mathbb{F}$*, called* the constraints*,*
- F *is a partial function* $\{1, \dots, n\} \to \mathbb{F}^m$*, with values denoted as* F_i*, the* fixed columns *and*
- $I, O \subseteq \{1, \dots, n\} \times \{1, \dots, m\}$ *are called the* input *and* output *cells.*

Do we also want to allow individual cell constraints?

lookups

We often omit the field $\mathbb{F}$*.*

Definition 1.2. *A mapping* $s = \mathbb{F}^{n \times m}$ satisfies *the polynomial constraint system* ($\mathbb{F}$, n, m, P, F, I, O) *on the input* $\iota \colon I \to \mathbb{F}$ *and* computes the output $o \colon O \to \mathbb{F}$ *if it*

- *satisfies the inputs:* $\iota(x) = s(x)$ *for all* $x \in I$*,*
- *satisfies the output:* $o(x) = s(x)$ *for all* $x \in O$*,*
- *coincides with the fixed columns:* $s(i, j) = F_i(j)$*, for all* i *where* F_i *is defined and all* $j \in \{1, \dots, m\}$*, and*

- *satisfies the constraints: for every $p \in P$ and every $j \in \{1, \ldots, m\}$:*

$$p(w_1, \ldots, w_n, w'_1, \ldots, w'_n) = 0,$$

with $w_i = s(i, j)$ and $w'_i = s(i, j + 1)$, where $m + 1$ is understood to be 1.

A common danger with polynomial constraint systems is that they are underconstrained, meaning that one input can lead to different outputs depending on how the rest of the table is filled. This is mainly problematic since there usually is a canonical way to fill the table that is provided together with the constraint system. The following definition captures what we mean when we say that a polynomial constraint system is not underconstrained.

Definition 1.3. *A polynomial constraint system $C = (\mathbb{F}, n, m, P, F, I, O)$ has* unique solutions *if for every function $\iota \colon I \to \mathbb{F}$, all mappings s that satisfy C on the input ι, compute the same output.*

While this is the correct definition, we will be dealing with a much stricter definition that is more in line with an algorithm that tries to solve the system step by step instead of performing global reasoning.

Definition 1.4. *A polynomial constraint system admits* unambiguous directed progress *if (informally), any*

define one-row progress, which includes "unknown" variables.

Should we also define progress according to a specific algorithm? This could be useful for the case where there is a unique solution but we just can't find it.

Property 1.5. *A polynomial constraint system (P, I) is*

express the condition that the system is not underconstrained in the sense that there are multiple solutions, but setting some values do not influence the next - what does that mean? What is the output?

Also define if a constraint system admits directed progress