Magic Tables

|  |  |
| --- | --- |
|  | The input dimension of an algorithm |
|  | At this point we only consider n a natural number that is greater than 0 |
|  | An algorithmic function |
|  | An algorithmic function with input size i and output size o |
|  | An algorithmic function with input size i and output size 1 (the default) |
|  | An algorithmic function of degree 3 |
|  | Modulo range |
|  | A truth table |
|  | Returns the truth table of a single algorithmic function |
|  | Returns the abstract algorithmic function of a truth table |
|  | The abstract notion of an algorithmic function and the concrete aspect of a truth table are equivalent. The truth table is simply one possible representation of an algorithmic function. |
|  |  |
|  | An algorithmic function mapped to a truth table of size 4 (degree 2) |
|  | An algorithmic function mapped to a truth table of size 8 (degree 3) |
|  | A sample showing how an arithmetic operation can be applied to the truth table reprensentation of an algorithmc function |
|  | Hence, algorithmic functions can be combined by applying functions and operations on them.  Here, we use of course modulo arithmetic. |
|  | For a given algorithmic function of dimension d, the size of its truth table is 2^d |
| Binary Numbers |  |
|  | A binary number of size s is an ordered set of binary numbers of size s. |
|  | *Notation*  As a convention, we will write from the right with the least significant bit to the left with the most significant bit.  We provide here an illustrative example with binary numbers of size 2. |
| Retrieve bits from binary numbers |  |
|  | returns the nth bit from a binary number with 0 based index, reading from right (least significant) to left (most significant) |
|  | *Samples* |
| Binary numbers sets |  |
|  | is the set of atomic binary numbers. |
|  | is the exhaustive set of binary numbers of fixed size . |
|  | is the empty set. |
|  |  |
|  |  |
|  |  |
| Algosets |  |
|  | An algoset is an ordered set of algorithmic functions. Why ordered? Just to make our life easier. |
| Exhaustive algosets |  |
|  | *Definition*  An exhaustive algoset is an algoset that contains all possible algorithmic functions for a given pair of input dimension and output dimension. |
|  | *Notation*  This is how an exhaustive algoset is noted, with input dimension and output dimension . |
|  | *Notation*  Shortcut notation for exhaustive algosets with output dimension 1. |
|  | *Notation*  A matrixial representation of an exhaustive algoset of output dimension 1.  Every row represents the truthtable of an algorithm that is part of the algoset. |
|  | The matrixial representation of an exhaustive algoset is enriched here with an explanatory legend.  Index  Ordered  This leads to a nice property where the binary representation of a truthtable = the 0-based index row position of the algorithm in the exhaustive algoset. |
|  | The algoset 0. |
|  | The algoset 1. |
|  | The algoset 2 (incomplete). |
| Truthtable Size |  |
|  | Returns the dimension of truthtables in an algoset  The number of columns in an matrix |
|  |  |

|  |  |
| --- | --- |
| Truthtable vlookup attack and parade |  |
|  | *Notation*  The inverse algorithm of |
|  | This is what we want to avoid. |
|  | By definition, a truthtable gives the mapping of the output value for every possible input values. |
| target\_output  input = 0  loop:  output = a(input)  if output == target\_output return target\_output  input++  goto loop | Hence, if the attacker knows the algorithm, he may populate the truth table by computation. |
|  | The size of the input of an algorithm may be augmented with a reasonably low linear effort increase in terms of computer resource usage. But the size of the resulting table is an exponential function.  This leads to a situation were defending against a truthtable lookup attack is easily accomplished by augmenting the size of the input until the computational cost of executing the attack becomes deterrent. |
| Algoset Size |  |
|  | Returns the size of an algoset, i.e. the number of algorithmic functions within an algoset.  The number of rows in the matrixial representation of an algoset. |
|  | Annotation: just another way to write the same thing. |
|  |  |
|  | This function returns the identifier of an algorithmic function |
|  | The unique and sequential identifier of an algorithm is… its truth table |
| Question: si n | |
|  | Question: |
| Algorithm with key is like algorithm without key |  |
|  | If an arbitrary algorithm takes two parameters: a key and an input. |
|  | For every key it is possible to define an algorithm that receives a single parameter input and embeds the key in its inner functioning. |
| Algorithms with output size > 1 can be composed from algorithms with output size = 1 |  |
|  | For all algorithms with an output size larger than 1, there is an algorithm that returns one particular bit of the original algorithm. |
|  | We define a function & that concatenates the results of several algoritms.  Rule: all algorithms passed as parameters must have the same input size. |
|  | The size of the concatenated algorithm is equal to the sum of the sizes of the algorithms that have been concatenated. |