Yale University
Department of Computer Science

Inductive Inference of Total Recursive Functions by Probabilistic and Deterministic Strategies

Martinch Krikis Rusins Freivalds

YALEU/DCS/TR-936
November 1992

Preparation and distribution of this work was supported by NSF grant CCR-9014943

Inductive Inference of Total Recursive Functions by Probabilistic and Deterministic Strategies

Mārtiņš Kriķis
Yale University*

Rūsiņš Freivalds
University of Latvia**

November 24, 1992

Abstract

Inductive inference of total recursive functions by probabilistic and deterministic strategies with a bounded number of mind changes is considered.

It is proved that for every nonnegative integer k each class of functions that is identifiable with probability greater than $\frac{2^{k+2}-2}{2^{k+2}-1}$ and at most k mind changes is identifiable deterministically with at most k mind changes.

It also is proved that for every nonnegative integer k and every positive integer n there exist classes of functions that are identifiable with probability $p_k^n = \frac{(2^{k+1}-1)n+2^{k+1}-1}{2^{k+1}n+2^{k+1}-1}$ and at most k mind changes but are not identifiable with probability greater than p_k^n and at most k mind changes.

1 Introduction

This paper concerns inductive inference of total recursive functions by probabilistic and deterministic strategies. The identifying strategies are given the values of

*Department of Computer Science, Yale University, P.O. Box 2158, New Haven, CT 06520, USA. E-mail: krikis@cs.yale.edu

**Institute of Mathematics and Computer Science, University of Latvia, Rainis Blvd. 29, Riga LV-1250, LATVIA. E-mail: rusins@cs.lu.riga.lv

the function at successive points and they may conjecture a hypothesis from time to time—their guess about the function's Gödel number. The Gödel number of a function is a Turing machine program for computing this function. The maximum number of mind changes is bounded. A mind change for the identifying strategy is a situation when it conjectures a new hypothesis which is not equal (as a number or a string) to the previous one. Without loss of generality we consider only strategies that never conjecture the same hypothesis as the previous one. So we can as well say that the number of hypotheses is bounded. Then a bound of k on the maximum number of mind changes is equivalent to a bound of $k+1$ on the total number of hypotheses output.

Since all the finite length ordered tuples of nonnegative integers are efficiently enumerable we can consider the identifying strategies to be functions from nonnegative integers to nonnegative integers. Strategies have to be implementable on Turing machines where probabilistic strategies have to be implementable on probabilistic Turing machines (that is, those equipped with the simplest Bernoulli random number generator with two equiprobable outputs, called a "fair coin"). When infinite computation paths are allowed the Bernoulli random number generator (i.e., coin flips) is sufficient to simulate a random number generator with any number of equiprobable outputs. Therefore we may assume that this more general random number generator is available.

If the last hypothesis of a strategy is a correct Gödel number of the given function then the strategy is said to identify the function. For probabilistic strategies all the possible computation paths generated by different outcomes of the random number generator are considered. A probabilistic strategy is said to identify the function with the probability corresponding to the total probability of all the computation paths in which the last hypothesis is a correct Gödel number of the function. A class of functions is said to be identifiable (deterministically or probabilistically with a certain probability) if there is a strategy (a deterministic or a probabilistic one, respectively) that identifies every function in the class (deterministically or probabilistically with the respective probability).

We will denote by EX_k the class containing those classes of functions that are identifiable deterministically with at most k mind changes and by $EX_k(p)$ the class containing those classes of functions that are identifiable with probability p and at most k mind changes.

We fix the maximum number of mind changes and look at the power of probabilistic and deterministic algorithms with this restriction. We find that when the required probability of identification is sufficiently high probabilistic strategies are not stronger than deterministic ones. We also find for every number k a sequence of "critical probabilities" that have the following property. For each critical probability p_k^n there are classes of functions that are identifiable with probability p_k^n and at most k mind changes but are not identifiable with probability greater than p_k^n and at most k mind changes.

This work generalizes theorems 1 and 2 of Freivalds' previous work [1] and theorem 3.1 in the work of Wiehagen, Freivalds and Kinber [4]. Theorem 3.2 of [4] is in contradiction with Theorem 1 below and therefore seems to be erroneous. Related results have been obtained by Daley and Kalyanasundaram in [3]; these will be discussed in Section 5.

2 Transition from probabilistic to deterministic identification

In [1] it was proved that if a class of functions is identifiable with probability greater than 2/3 and no mind changes then this class is identifiable deterministically with no mind changes. A natural question is how the situation changes if we consider identification with at most one, two, three, etc. mind changes. In each of these cases there exists a certain number such that every class of functions that is identifiable with probability greater than this number is identifiable deterministically. The following theorem proves this assertion.

Theorem 1 *For every nonnegative integer k each class of functions that is identifiable with probability greater than $\frac{2^{k+2}-2}{2^{k+2}-1}$ and at most k mind changes is identifiable deterministically with at most k mind changes. More formally,*

$$\forall k, p, U \ : \ U \in EX_k(p) \ \& \ p > \frac{2^{k+2}-2}{2^{k+2}-1} \Rightarrow U \in EX_k.$$

Proof. Assume that U is a class of functions that is identifiable by a probabilistic strategy F with probability greater than $\frac{2^{k+2}-2}{2^{k+2}-1}$ and at most k mind changes. We will construct a deterministic strategy to identify this class of functions with at most k mind changes.

The deterministic strategy for identifying this class of functions will first simulate the work of the strategy F; however, instead of using the random number generator it will follow all the computation paths of F and remember their respective probabilities. All the computation paths will be considered simultaneously, i.e. a step from each one at a time. In other words, the deterministic strategy will follow the computation tree of F.

Now we will describe the conditions when the deterministic strategy has to conjecture its first, second, etc. hypothesis, up to the $k+1^{st}$ inclusive. Each hypothesis, if conjectured, will be made on the basis of a finite amount of information. And at least one hypothesis will be always conjectured.

In each computation path F will conjecture a different sequence of hypotheses: some first one, then the second, third, etc. Our deterministic strategy will remember these results and the probability of the computation path at the time when each particular hypothesis is conjectured.

The deterministic strategy will conjecture a hypothesis h_i (to be described below) whenever the total probability over all the computation paths of F that have each output at least i hypotheses reaches $\frac{2^{k-i+3}-2}{2^{k+2}-1}$ or more. If it is possible to conjecture more than one hypothesis it will conjecture h_i with the biggest possible index i. The deterministic strategy will not necessarily conjecture h_1, h_2, etc., it will conjecture those h_i for which there are enough $(\frac{2^{k-i+3}-2}{2^{k+2}-1})$ i-th or later hypotheses of F (with respect to their probabilities over all the computation paths).

Now we will define the hypothesis h_i. Since every intuitively computable function is recursive, we do not have to give explicitly the Gödel number of the function to be identified. It suffices to describe how to compute this function.

The hypotheses of F are supposed to be Gödel numbers (i.e., programs) of the function to be identified. Generally, over all the computation paths there should appear sufficiently many correct Gödel numbers with respect to their probabilities. Knowing a Gödel number of a function it is possible to construct a Turing machine that computes this function. Hypothesis h_i will consist of constructing the Turing machines corresponding to the hypotheses of F on basis of which it was conjectured and then running them. That is, from each computation path that has yielded $j \geq i$ hypotheses it will take the i^{th} hypothesis, interpret it as a program and construct the corresponding machine. Also each machine will be assigned a weight equal to the probability of the respective computation path. In order to determine the value of the h_i at point x the Turing machines will be run simultaneously (that is, one step of one machine, one of another, etc.) on argument x until they have returned one value the total weight of which is at least $\frac{2^{k-i+2}-1}{2^{k+2}-1}$. So, it will look for the value that is given by a weighted majority of the initial basis hypotheses.

We cannot claim that every h_i will be correct. But we will now prove that the last h_i will.

So, let us assume that the deterministic strategy has conjectured h_i and is never going to conjecture a different hypothesis. This implies that among the hypotheses of F (that all have order i or higher) less than $\frac{2^{k-i+2}-2}{2^{k+2}-1}$ (with respect to their probability) can be changed in all the computation paths of F. Otherwise the strategy would be forced to conjecture h_{i+1}.

Since strategy F identifies class U with probability exceeding $\frac{2^{k+2}-2}{2^{k+2}-1}$, the total probability of absolutely incorrect hypotheses of F (the ones that are not correct but are never changed) is less than $\frac{1}{2^{k+2}-1}$. Therefore among all the hypotheses of strategy F on basis of which the h_i is conjectured there must be less than

$$\frac{2^{k-i+2}-2}{2^{k+2}-1}+\frac{1}{2^{k+2}-1}=\frac{2^{k-i+2}-1}{2^{k+2}-1}$$

incorrect ones in terms of probability. Hence eventually these Turing machines will return the value $f(x)$ with the total weight exceeding $\frac{2^{k-i+2}-1}{2^{k+2}-1}$ and the last hypothesis always will be correct.

So we have proved that there is a deterministic strategy that identifies class U with at most k mind changes. $\square$

3 Critical Probabilities

In the previous section we found numbers p_k such that identification with probability exceeding p_k and at most k mind changes was not stronger than deterministic identification with at most k mind changes. But we do not yet have any guarantee that the same cannot be said for some other smaller number. Therefore in this section we will show that there are classes of functions that are identifiable with probability p_k but are not identifiable deterministically (with at most k mind changes).

In fact we will prove a much stronger theorem, namely, for each fixed number k of allowed mind changes we will find a sequence $p_k^1 = p_k, p_k^2, p_k^3, \ldots$ of numbers (called "critical probabilities" in [1]) converging to $1-\frac{1}{2^{k+1}}$ with the following property: there exist classes of functions identifiable with probability p_k^n but not identifiable

	1	2	3	4	...	n	$\rightarrow$	∞
0	$\frac{2}{3}$	$\frac{3}{5}$	$\frac{4}{7}$	$\frac{5}{9}$	...	$\frac{n+1}{2n+1}$	$\rightarrow$	$\frac{1}{2}$
1	$\frac{6}{7}$	$\frac{9}{11}$	$\frac{12}{15}$	$\frac{15}{19}$	...	$\frac{3n+3}{4n+3}$	$\rightarrow$	$\frac{3}{4}$
2	$\frac{14}{15}$	$\frac{21}{23}$	$\frac{28}{31}$	$\frac{35}{39}$	...	$\frac{7n+7}{8n+7}$	$\rightarrow$	$\frac{7}{8}$
3	$\frac{30}{31}$	$\frac{45}{47}$	$\frac{60}{63}$	$\frac{75}{79}$	...	$\frac{15n+15}{16n+15}$	$\rightarrow$	$\frac{15}{16}$
4	$\frac{62}{63}$	$\frac{93}{95}$	$\frac{124}{127}$	$\frac{155}{159}$	...	$\frac{31n+31}{32n+31}$	$\rightarrow$	$\frac{31}{32}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$		$\vdots$
k	$\frac{2^{k+2}-2}{2^{k+2}-1}$	$\frac{2^{k+1}.3-3}{2^{k+1}.3-1}$	$\frac{2^{k+1}.4-4}{2^{k+1}.4-1}$	$\frac{2^{k+1}.5-5}{2^{k+1}.5-1}$	...	$\frac{(2^{k+1}-1)n+2^{k+1}-1}{2^{k+1}n+2^{k+1}-1}$	$\rightarrow$	$1-\frac{1}{2^{k+1}}$

Figure 1: Critical probabilities

with probability greater than p_k^n for each positive integer n. The situation is depicted in Figure 1.

Theorem 2 *For every nonnegative integer k and every positive integer n there exist classes of functions that are identifiable with probability $p_k^n = \frac{(2^{k+1}-1)n+2^{k+1}-1}{2^{k+1}n+2^{k+1}-1}$ and at most k mind changes but are not identifiable with probability greater than p_k^n and at most k mind changes. More formally,*

$$\forall k \ \forall n > 0 \ \exists U \ : \ (U \in EX_k(p_k^n) \ \& \ \forall (p > p_k^n) \ : \ U \notin EX_k(p)),$$

where $p_k^n = \frac{(2^{k+1}-1)n+2^{k+1}-1}{2^{k+1}n+2^{k+1}-1}$.

Proof. To prove the theorem we will need a lemma which actually is the Smullyan form of the fixed point theorem (see [2]).

Lemma 1 *Let j be a positive integer and $h_1, h_2, \ldots, h_j$ be total recursive functions of j arguments. Then there exist positive integers $m_1, m_2, \ldots, m_j$ such that*
$\varphi_{m_1} \equiv \varphi_{h_1(m_1,\, m_2,\, ...,\, m_j)}$, $\varphi_{m_2} \equiv \varphi_{h_2(m_1,\, m_2,\, ...,\, m_j)}$, $\cdots$, $\varphi_{m_j} \equiv \varphi_{h_j(m_1,\, m_2,\, ...,\, m_j)}$,
where φ_q denotes the function whose Gödel number is q.

Now we will begin the proof of the theorem. First let us fix some arbitrary k and n. We introduce the following notation to enhance the readability of the formulae in the proof: $N = 2^k(n+1)$, $S = \{1, 2, \ldots, 2N-1\}$. Thus, $p_k^n = \frac{2N-(n+1)}{2N-1}$.

We will consider the following class of total recursive functions U. It will consist of functions f such that:

1. $\forall i \in S \ : \ |\{x | f(x) = i \pmod{2N}\}| \leq k+1$, i.e., for every i from the set S there are at most $k+1$ points x such that $f(x) = i \pmod{2N}$;
2. $|\{i \in S | f \equiv \varphi_{g_i(x_i^*)}\}| \geq 2N - (n+1)$, where $x_i^* = max\{x | f(x) = i \pmod{2N}\}$ and $g_i(x) = \frac{f(x)-i}{2N}$, i.e., there are at least $2N-(n+1)$ different i from the set S such that $\frac{f(x_i^*)-i}{2N}$ is a correct Gödel number of the function f.

In other words, all the values of a function that are not divisible by $2N$ will be considered as possible encodings of a Gödel number for this function. If division by $2N$ gives the remainder $a \neq 0$ we will call this value a "type a code". These functions will not have more than $k+1$ encoded values in each code type and, furthermore, at least $2N-(n+1)$ of the possible $2N-1$ code types will have the property that the last value encoded this way is an encoding of a correct Gödel number for this function.

First we prove that this class of functions is identifiable with probability

$$\frac{2N-(n+1)}{2N-1} = \frac{(2^{k+1}-1)n + 2^{k+1} - 1}{2^{k+1}n + 2^{k+1} - 1} = p_k^n$$

and at most k mind changes. We can assume that the inductive inference machine is equipped with a random number generator that has $2N-1$ equiprobable outputs and therefore to identify the function it can choose a random number i from S and then find the type i codes among the values of the function. Each code found results in a new hypothesis, namely, if $f(x) = i \pmod{2N}$ then the hypothesis will be the number $g_i(x) = \frac{f(x)-i}{2N}$. If the machine guesses a good code type the hypothesis conjectured last will be right. The probability that this will happen is at least $\frac{2N-(n+1)}{2N-1} = p_k^n$.

Now we will prove that the class U cannot be identified with probability exceeding p_k^n. Let us assume the contrary. Suppose there is a strategy F that identifies class U with probability $p > p_k^n$ and at most k mind changes. Let $p > p' > p_k^n$. We will now find a function in class U such that F does not identify it correctly with probability p and at most k mind changes.

We associate every ordered N-tuple of nonnegative integers with N particular functions in the way to be described below. We will be able to use the lemma to get the result that there exists an ordered N-tuple such that the functions corresponding to it all belong to the class U. The definition of the functions for a given N-tuple depends on the possible computation paths of F on initial segments of the functions to be defined. Eventually we will come to the conclusion that not all of these functions can be identified correctly and with probability p.

Now we will describe in detail the construction of the N functions corresponding to an ordered N-tuple of nonnegative integers. In defining the functions we will use a "guiding table" which will have many useful properties. So now we will make a big digression and first describe what the guiding table is and how it can be constructed.

3.1 Construction of the Table

The table consists of N horizontal rows and is split into $k+2$ zones by vertical separating lines. The zones are numbered $1, 2, \ldots, k+2$ from left to right. We will frequently refer to a "lane" in the table, so we will now introduce the concept of a lane. We start with zone 1 and declare that there is one lane (i.e., all the rows in zone 1 belong to only one lane). For zone 2 we separate the table into $n+1$ equally sized and nonintersecting lanes (i.e., each lane consists of 2^k adjacent rows). For zone 3 and further we introduce lanes inductively. We take every lane in the left neighboring zone and split it in equal halves. Continuing such splitting to the right end of the table we finally see that zone $k+2$ consists of N lanes that each occupies one row. An example of an empty table is given in Figure 2.

Now let us describe the contents of the table more precisely. The table holds ordered pairs of positive integers. The right components of the pairs are numbers from the set S and the left components are from the set $\{1, 2, \ldots, N\}$. The number of pairs is not necessarily the same in every row of the table. It does not have to be the same in every row even within the boundaries of a particular zone. However, the table has the following useful properties:

Property 1 For every particular zone z and lane l in it, every row that belongs to that lane has the same pairs in the same order in that zone. That is, within the boundaries of both a particular zone and a particular lane the contents of every row are exactly the same.

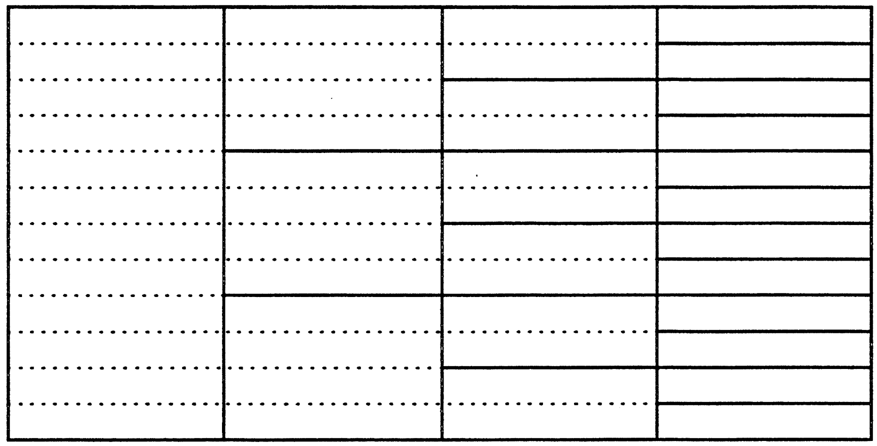

Figure 2: Empty table: $k = 2, n = 2$

Property 2 In every particular row of the table there are at most $k + 1$ pairs with the same right components.

Property 3 Let us define an indicator function $I_{r,z}((u, v))$ for each particular zone z and row r, which will be 1 if and only if the pair (u, v) is the rightmost pair with the right component v in row r and zones 1 through z. Let row r belong to lane l in zone z and let lane l consist of rows r_i through r_j. Then the number of pairs whose indicator function is 1 and whose left component is in $[r_i, r_j]$ is at least $2N - (n + 1)$. That is,

$$|\{(u, v) \mid I_{r,z}((u, v)) = 1 \;\; \& \;\; r_i \leq u \leq r_j\}| \;\geq\; 2N - (n + 1).$$

It is not obvious that such a guiding table exists. And if it does then there is a question how to find it. Therefore we will now describe an algorithm to fill the table with pairs and will prove that the resulting table satisfies all the requirements stated above. We assume that there already are the N rows in the table and it is split in zones and lanes as described. We also assume that writing takes place from left to right, so we are able to immediately write down the next pair to the right of the previous. We also assume that the horizontal size of the zones is sufficiently big, that is, we will never write our pairs past any vertical zone separator.

Algorithm

1. *In zone 1 and row 1 put* $2N - (n+1)$ *pairs with dummy (to be specified later) left components and different right components from the set* S. *Make the other rows exactly the same.*

(Zone 1 is filled out now; see Figure 3.)

2. *For every lane* l *in zone 2 let* R_l *be the set of those numbers in* S *that have not been used as the right components for the pairs in any row of this lane.*

(We call sets R_l the sets on reserve. At this moment they contain numbers that can be valid right components of the pairs but are not used yet. The sets on reserve are the same for every lane of zone 2 now but later different lanes will have different sets; see Figure 3.)

3. *For each lane* l *of zone 2 do*

> **4.** *In the top row of lane* l *cross out all pairs except the ones beginning with the* $(2^{k+1}-1)(l-1)+1^{st}$ *pair and ending with the* $(2^{k+1}-1)l^{th}$ *one. Make the other rows of lane* l *exactly the same.*
>
> (There will be blocks of uncrossed pairs along the diagonal of zone 1, if we think of zone 1 as an array with dimensions $N \times (2^{k+1}-1)$; see Figure 3.)
>
> **5.** *In zone 2 of the top row of lane* l *put pairs with dummy left components and right components taken from the set* R_l. *Make the other rows of lane* l *exactly the same.*
>
> **6.** *Then in the top row of lane* l *put pairs with dummy left components but right components taken from the set* S, *all pairwise different and also different from right components of uncrossed pairs in this row. Put so many of them that the total of uncrossed pairs in this row is* $2N-(n+1)$. *Make the other rows of lane* l *exactly the same.*

7. *Enddo*

(Zone 2 is filled out now; see Figure 4.)

8. *For* $i = 3$ *to* $k+2$ *step 1 do*

(Filling out zone i in the table.)

> **9.** *Cross once more every pair in the table which is crossed just once.*

(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	
(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	R_1={22,23}
(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	
(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	R_2={22,23}
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~	
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)	
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)	R_3={22,23}
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)	
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)	

Figure 3: Zone 1 filled, sets on reserve for zone 2 and crossing pairs

(,22)(,23)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)	R_1={8,9}
(,22)(,23)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)	
(,22)(,23)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)	R_2={8,9}
(,22)(,23)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)	
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,15)(,16)(,17)(,18)(,19)	R_3={20,21}
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,15)(,16)(,17)(,18)(,19)	
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,15)(,16)(,17)(,18)(,19)	R_4={20,21}
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,15)(,16)(,17)(,18)(,19)	
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)	R_5={13,14}
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)	
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)	R_6={13,14}
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)	

Figure 4: Zone 2 filled and sets on reserve for zone 3

(This is done to be able to distinguish in the future between recently crossed pairs and old crossings.)

10. *For every lane l in zone i let R_l be the set of those numbers in S that have been used as the right components for the pairs in any row of this lane less than $i-1$ times and do not appear in the uncrossed pairs.*

(See Figure 4.)

11. *For each lane l in zone i do*

> **12.** *If l is odd, then in the top row of lane l cross out half of the pairs the right component of which is used $i-1$ times. Also cross out any number (possibly none) of the pairs the right component of which is used less than $i-1$ times. Make the other rows of lane l exactly the same.*
>
> **13.** *If l is even, then in every row of lane l cross out all those pairs that remained uncrossed in the rows of lane $l-1$.*
>
> (See Figure 5. The difference between the rows of odd lanes and even lanes of zone i will be introduced only now, after the crossing.)
>
> **14.** *In zone i of the top row of lane l put pairs with dummy left components and right components taken from the set R_l. Make the other rows of lane l exactly the same.*
>
> **15.** *Then in the top row of lane l put pairs with dummy left components and right components corresponding to those of the pairs that are crossed only once and used less than $i-1$ times not counting this usage. Make the other rows of lane l exactly the same.*
>
> **16.** *Then in the top row of lane l put pairs with dummy left components but right components taken from the set S, all pairwise different and also different from right components of uncrossed pairs in that row. Put so many of them that the total of uncrossed pairs in that row is $2N-n-1$. Make the other rows of lane l exactly the same.*

17. *Enddo*

(Zone i is filled out now; see Figure 5.)

18. *Enddo*

(The table is ready except for the left components; see Figure 6.)

19. *For every uncrossed pair in the table set its left component equal to the number of the row it is in.*

~~(,1)(,2)(,3)(,4)~~(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)~~(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
(,1)(,2)(,3)(,4)~~(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
(,1)(,2)(,3)(,4)~~(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)~~(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)~~(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)~~(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)~~(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)~~(,20)(,21)
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)~~(,20)(,21)
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)~~(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)~~(,20)(,21)~~

(,22)(,23)~~(,10)(,11)(,12)(,13)(,14)(,15)~~(,16)(,17)(,18)(,19)(,20)(,21)
(,22)(,23)~~(,10)(,11)(,12)(,13)(,14)(,15)~~(,16)(,17)(,18)(,19)(,20)(,21)
~~(,22)(,23)~~(,10)(,11)(,12)(,13)(,14)(,15)~~(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,22)(,23)~~(,10)(,11)(,12)(,13)(,14)(,15)~~(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,15)(,16)(,17)(,18)(,19)
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,15)(,16)(,17)(,18)(,19)
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,15)(,16)(,17)(,18)(,19)~~
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,15)(,16)(,17)(,18)(,19)~~
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,8)(,9)(,10)(,11)(,12)~~
(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,8)(,9)(,10)(,11)(,12)~~
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,8)(,9)(,10)(,11)(,12)
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,8)(,9)(,10)(,11)(,12)

(,8)(,9)(,1)(,2)(,3)(,4)(,10)(,11)(,12)(,13)	$R_1=\{14,15\}$
(,8)(,9)(,1)(,2)(,3)(,4)(,10)(,11)(,12)(,13)	$R_2=\{14,15\}$
(,8)(,9)(,5)(,6)(,7)(,22)(,23)(,16)(,17)(,18)(,19)	$R_3=\{20,21\}$
(,8)(,9)(,5)(,6)(,7)(,22)(,23)(,16)(,17)(,18)(,19)	$R_4=\{20,21\}$
(,20)(,21)(,13)(,14)(,22)(,23)(,1)(,2)(,3)(,4)	$R_5=\{5,6\}$
(,20)(,21)(,13)(,14)(,22)(,23)(,1)(,2)(,3)(,4)	$R_6=\{5,6\}$
(,20)(,21)(,8)(,9)(,10)(,11)(,12)(,7)(,15)(,16)(,17)	$R_7=\{18,19\}$
(,20)(,21)(,8)(,9)(,10)(,11)(,12)(,7)(,15)(,16)(,17)	$R_8=\{18,19\}$
(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,7)(,8)(,9)(,10)	$R_9=\{11,12\}$
(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,7)(,8)(,9)(,10)	$R_{10}=\{11,12\}$
(,13)(,14)(,20)(,21)(,22)(,23)(,3)(,4)(,5)(,6)	$R_{11}=\{1,2\}$
(,13)(,14)(,20)(,21)(,22)(,23)(,3)(,4)(,5)(,6)	$R_{12}=\{1,2\}$

Figure 5: Odd–even crossing, zone 3 filled and sets on reserve for zone 4

~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)~~(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
(,1)(,2)(,3)(,4)~~(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)~~(,8)(,9)(,10)(,11)(,12)~~(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)~~(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)~~(,21)
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)~~(,20)~~(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)(,13)(,14)~~(,15)(,16)(,17)(,18)(,19)~~(,20)(,21)~~

~~(,22)(,23)(,10)(,11)(,12)(,13)(,14)(,15)(,16)(,17)(,18)~~(,19)(,20)(,21)
(,22)(,23)~~(,10)(,11)(,12)(,13)(,14)(,15)~~(,16)(,17)(,18)~~(,19)(,20)(,21)~~
~~(,22)(,23)(,10)(,11)(,12)(,13)~~(,14)(,15)~~(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,22)(,23)~~(,10)(,11)(,12)(,13)~~(,14)(,15)(,16)(,17)(,18)(,19)(,20)(,21)~~
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,15)(,16)~~(,17)(,18)(,19)
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,15)(,16)~~(,17)(,18)(,19)~~
~~(,22)(,23)~~(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,15)(,16)(,17)(,18)(,19)~~
(,22)(,23)~~(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,15)(,16)(,17)(,18)(,19)~~
~~(,22)(,23)(,1)(,2)~~(,3)(,4)(,5)(,6)~~(,7)(,8)(,9)(,10)(,11)(,12)~~
(,22)(,23)(,1)(,2)~~(,3)(,4)(,5)(,6)(,7)(,8)(,9)(,10)(,11)(,12)~~
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)(,7)(,8)(,9)~~(,10)(,11)(,12)
~~(,22)(,23)(,1)(,2)(,3)(,4)(,5)(,6)~~(,7)(,8)(,9)~~(,10)(,11)(,12)~~

(,8)(,9)(,1)(,2)(,3)(,4)~~(,10)(,11)~~(,12)(,13)	(,14)(,15)(,5)(,6)(,7)(,22)(,23)(,16)(,17)(,18)
~~(,8)(,9)(,1)(,2)(,3)(,4)~~(,10)(,11)~~(,12)(,13)~~	(,14)(,15)(,19)(,20)(,21)(,8)(,9)(,1)(,2)(,3)(,4)
(,8)(,9)(,5)(,6)(,7)(,22)(,23)~~(,16)(,17)~~(,18)(,19)	(,20)(,21)(,1)(,2)(,3)(,4)(,10)(,11)(,12)(,13)
~~(,8)(,9)(,5)(,6)(,7)(,22)(,23)~~(,16)(,17)~~(,18)(,19)~~	(,20)(,21)(,14)(,15)(,8)(,9)(,5)(,6)(,7)(,22)(,23)
(,20)(,21)(,13)(,14)(,22)(,23)~~(,1)(,2)~~(,3)(,4)	(,5)(,6)(,8)(,9)(,10)(,11)(,12)(,7)(,15)(,16)
~~(,20)(,21)(,13)(,14)(,22)(,23)~~(,1)(,2)~~(,3)(,4)~~	(,5)(,6)(,17)(,18)(,19)(,20)(,21)(,13)(,14)(,22)(,23)
~~(,20)(,21)(,8)(,9)~~(,10)(,11)(,12)~~(,7)(,15)~~(,16)(,17)	(,18)(,19)(,13)(,14)(,22)(,23)(,20)(,21)(,8)(,9)
(,20)(,21)(,8)(,9)~~(,10)(,11)(,12)~~(,7)(,15)~~(,16)(,17)~~	(,18)(,19)(,1)(,2)(,3)(,4)(,5)(,6)(,10)(,11)(,12)
~~(,13)(,14)(,15)(,16)~~(,17)(,18)(,19)~~(,7)(,8)~~(,9)(,10)	(,11)(,12)(,20)(,22)(,23)(,1)(,2)(,13)(,14)(,15)(,16)
(,13)(,14)(,15)(,16)~~(,17)(,18)(,19)~~(,7)(,8)~~(,9)(,10)~~	(,11)(,12)(,21)(,3)(,4)(,5)(,6)(,17)(,18)(,19)
(,13)(,14)(,20)(,21)(,22)~~(,23)(,3)(,4)~~(,5)(,6)	(,1)(,2)(,15)(,16)(,17)(,18)(,19)(,7)(,8)(,9)(,23)
~~(,13)(,14)(,20)(,21)(,22)~~(,23)(,3)(,4)~~(,5)(,6)~~	(,1)(,2)(,10)(,11)(,12)(,13)(,14)(,20)(,21)(,22)

Figure 6: Table ready except for the left components

20. *For every crossed pair that is in some zone z, lane l and position j from the left side of zone z set its left component equal to the left component of the unique uncrossed pair in zone z, lane l and position j.*

(The table is ready; see Figure 7.)

3.2 Correctness of the Construction

Now we will prove that this algorithm really produces a table with all the required properties. The crossed pairs are not different from the uncrossed ones, they are valid pairs and have to be considered when proving the properties of the table. Crossing just helps to build the table and will help to analyse it.

Claim 1 *For any zone z, lane l and position j from the left side of zone z there is exactly one uncrossed pair.*

Proof. Let us look at lane l and zone $z > 1$ in the table immediately after it is filled out with pairs containing dummy left components. It has 2^{k+2-z} rows and no pair is crossed in it. However it has $k + 2 - z$ zones to the right of it and with filling out each of these zones (with pairs containing dummy left components) the number of uncrossed pairs in zone z and position j is reduced to half its previous value. That is because each lane is split in two equaly sized lanes and step 13 says: "If l is even, then in every row of lane l cross out all those pairs that remained uncrossed in the rows of lane $l - 1$." Hence, when all zones are filled out just one uncrossed pair will remain in lane l zone z and position j. For the first zone the argument is similar. The difference is only in that we first apply step 4 for crossing out pairs in it which leaves it having 2^k uncrossed pairs in every position. Step 4 is needed for filling out the second zone and since then each lane is being split in two with each new zone, for k remaining zones and only steps 12 and 13 are applied for crossing pairs. Therefore the same argument as for zones $z > 1$ can be used. So the claim is true for any zone z, lane l and position j. □

This claim is a justification for step 20 of the algorithm. And, in fact, there is no need to do the crossing in step 4 so regularly. It would suffice to cross any pairs as long as rows $(i - 1)2^k + 1$ through $i2^k$ have the same pairs crossed, for every i, $1 \leq i \leq n + 1$ and in every row exactly $2^{k+1} - 1$ pairs remain uncrossed.

(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)
(4,1)(4,2)(4,3)(4,4)(2,5)(2,6)(2,7)(6,8)(6,9)(6,10)(6,11)(6,12)(8,13)(8,14)(12,15)(12,16)(12,17)(12,18)(12,19)(10,20)(9,21)

(2,22)(2,23)(4,10)(4,11)(4,12)(4,13)(3,14)(3,15)(2,16)(2,17)(2,18)(1,19)(1,20)(1,21)
(2,22)(2,23)(4,10)(4,11)(4,12)(4,13)(3,14)(3,15)(2,16)(2,17)(2,18)(1,19)(1,20)(1,21)
(2,22)(2,23)(4,10)(4,11)(4,12)(4,13)(3,14)(3,15)(2,16)(2,17)(2,18)(1,19)(1,20)(1,21)
(2,22)(2,23)(4,10)(4,11)(4,12)(4,13)(3,14)(3,15)(2,16)(2,17)(2,18)(1,19)(1,20)(1,21)
(8,22)(8,23)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(6,7)(6,15)(6,16)(5,17)(5,18)(5,19)
(8,22)(8,23)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(6,7)(6,15)(6,16)(5,17)(5,18)(5,19)
(8,22)(8,23)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(6,7)(6,15)(6,16)(5,17)(5,18)(5,19)
(8,22)(8,23)(7,1)(7,2)(7,3)(7,4)(7,5)(7,6)(6,7)(6,15)(6,16)(5,17)(5,18)(5,19)
(10,22)(10,23)(10,1)(10,2)(9,3)(9,4)(9,5)(9,6)(12,7)(12,8)(12,9)(11,10)(11,11)(11,12)
(10,22)(10,23)(10,1)(10,2)(9,3)(9,4)(9,5)(9,6)(12,7)(12,8)(12,9)(11,10)(11,11)(11,12)
(10,22)(10,23)(10,1)(10,2)(9,3)(9,4)(9,5)(9,6)(12,7)(12,8)(12,9)(11,10)(11,11)(11,12)
(10,22)(10,23)(10,1)(10,2)(9,3)(9,4)(9,5)(9,6)(12,7)(12,8)(12,9)(11,10)(11,11)(11,12)

(1,8)(1,9)(1,1)(1,2)(1,3)(1,4)(2,10)(2,11)(1,12)(1,13)	(1,14)(1,15)(1,5)(1,6)(1,7)(1,22)(1,23)(1,16)(1,17)(1,18)
(1,8)(1,9)(1,1)(1,2)(1,3)(1,4)(2,10)(2,11)(1,12)(1,13)	(2,14)(2,15)(2,19)(2,20)(2,21)(2,8)(2,9)(2,1)(2,2)(2,3)(2,4)
(3,8)(3,9)(3,5)(3,6)(3,7)(3,22)(3,23)(4,16)(4,17)(3,18)(3,19)	(3,20)(3,21)(3,1)(3,2)(3,3)(3,4)(3,10)(3,11)(3,12)(3,13)
(3,8)(3,9)(3,5)(3,6)(3,7)(3,22)(3,23)(4,16)(4,17)(3,18)(3,19)	(4,20)(4,21)(4,14)(4,15)(4,8)(4,9)(4,5)(4,6)(4,7)(4,22)(4,23)
(5,20)(5,21)(5,13)(5,14)(5,22)(5,23)(6,1)(6,2)(5,3)(5,4)	(5,5)(5,6)(5,8)(5,9)(5,10)(5,11)(5,12)(5,7)(5,15)(5,16)
(5,20)(5,21)(5,13)(5,14)(5,22)(5,23)(6,1)(6,2)(5,3)(5,4)	(6,5)(6,6)(6,17)(6,18)(6,19)(6,20)(6,21)(6,13)(6,14)(6,22)(6,23)
(8,20)(8,21)(8,8)(8,9)(7,10)(7,11)(7,12)(8,7)(8,15)(7,16)(7,17)	(7,18)(7,19)(7,13)(7,14)(7,22)(7,23)(7,20)(7,21)(7,8)(7,9)
(8,20)(8,21)(8,8)(8,9)(7,10)(7,11)(7,12)(8,7)(8,15)(7,16)(7,17)	(8,18)(8,19)(8,1)(8,2)(8,3)(8,4)(8,5)(8,6)(8,10)(8,11)(8,12)
(10,13)(10,14)(10,15)(10,16)(9,17)(9,18)(9,19)(10,7)(10,8)(9,9)(9,10)	(9,11)(9,12)(9,20)(9,22)(9,23)(9,1)(9,2)(9,13)(9,14)(9,15)(9,16)
(10,13)(10,14)(10,15)(10,16)(9,17)(9,18)(9,19)(10,7)(10,8)(9,9)(9,10)	(10,11)(10,12)(10,21)(10,3)(10,4)(10,5)(10,6)(10,17)(10,18)(10,19)
(11,13)(11,14)(11,20)(11,21)(11,22)(12,23)(12,3)(12,4)(11,5)(11,6)	(11,1)(11,2)(11,15)(11,16)(11,17)(11,18)(11,19)(11,7)(11,8)(11,9)(11,23)
(11,13)(11,14)(11,20)(11,21)(11,22)(12,23)(12,3)(12,4)(11,5)(11,6)	(12,1)(12,2)(12,10)(12,11)(12,12)(12,13)(12,14)(12,20)(12,21)(12,22)

Figure 7: Table ready, crossing removed

Claim 2 *Within the boundaries of a particular zone and particular lane the contents of every row are exactly the same (property 1).*

Proof. Let us look at any step of the algorithm that writes pairs with dummy left components and certain right components in some zone and lane. These steps are always related to the whole lane in that zone. And it always says: "Put this and this in the top row of the lane. Make the other rows of the lane exactly the same." Therefore we can be sure that the right components of the pairs are the same within the boundaries of a particular zone and lane. Now let us look at the steps that write the left components of the pairs. By claim 1 it follows that step 19 applies only to one pair in a particular zone, lane and position. And for all the other pairs in that zone, lane and position step 20 applies which sets the left components equal to the left component of the pair to which step 19 was applied in that zone, lane and position. Therefore neither writing the right components nor the left ones introduce any difference within the boundaries of one zone and lane. □

Claim 3 *The cardinality of the sets on reserve is always n.*

Proof. The algorithm requires that in any row after filling out a new zone (with pairs having dummy left components) there must be $2N - (n+1)$ uncrossed pairs with pairwise different right components. Since we may use as right components $|S| = 2N - 1$ different numbers but have used only $2N - (n+1)$ of them when filling out the first zone, it follows that after filling it out $|R_l| = n$ for every lane l of zone 2. Now let us look at filling the other zones. Before filling out any new zone we cross out a certain number of the uncrossed pairs. Then we obviously have to add as many pairs as we just crossed out, since there have to be $2N - (n+1)$ uncrossed pairs with different right components. The next useful observation is that at any time there cannot be more pairs with the same right components in any row than the number of zones filled out. That again is because of the requirement that when we add pairs to the table they all must have different right components and different from those of the uncrossed pairs in the table as well. So only with a new zone we may use some right component one more time. From this it follows that the right component of any pair that is crossed out immediately before filling a new zone will either be used in this new zone or else be put in the set on reserve for the appropriate lane in the next zone. From all the above it follows that the new sets on reserve will again contain exactly n numbers. □

Claim 4 *In every particular row of the table there are at most $k+1$ pairs with the same right components (property 2).*

Proof. We will prove the claim by induction. The inductive hypothesis is the following: for every $i > 1$ when the i^{th} zone is filled out with pairs having dummy left components there are exactly $(2^{k-i+3} - 2)n$ pairs the right components of which have been used i times. First we can observe that when the second zone is filled out with pairs having dummy left components there are exactly $(2N-(n+1))-n-(2^{k+1}-1) = 2^{k+1}(n+1) - n - 1 - n - 2^{k+1} + 1 = (2^{k+1} - 2)n$ pairs the right components of which have been used twice. The numbers in this equation mean the following: $2N-(n+1)$ is the required number of uncrossed pairs in the table at any time; n is the cardinality of the set on reserve before filling out zone 2; and $2^{k+1} - 1$ is the number of uncrossed pairs in any row of zone 1 before filling out zone 2. This will be the base case for the induction.

Now let us assume that the inductive hypothesis is true for zone i and let us prove it for zone $i + 1$. Let us look at the actions performed in order to fill out zone $i + 1$. First thing we need to be concerned about is crossing the pairs in zones 1 through i. We will have to add as many new pairs as we cross out right before filling out zone $i + 1$. However we need not be concerned about crossing pairs that have been used less than i times. When filling the table, step 15 requires us to use pairs with these right components before those that have been used i times and so they will appear in zone $i+1$ as being used less than $i+1$ times. But for simplicity of our argument let us denote the number of pairs that have just been crossed but have the right components used less than i times by x. Crossing the pairs with right components that have been used i times is simple—step 12 requires to cross exactly half of them in odd number rows and step 13 requires to cross the other half in even number rows. So by the inductive hypothesis in zone $i+1$ we have to compensate for $x+(2^{k-i+2} - 1)n$ crossed pairs. And we have n pairs from the set on reserve for this purpose; x pairs that have been just crossed out and have the right components used less than i times; and the pairs that have the right components used i times. The algorithm requires that the former two types of pairs are used entirely before the latter may be used. Hence from the latter pairs exactly $(2^{k-1+2} - 2)n$ will appear in zone $i + 1$ and then they will have right components that have been used $i+1$ times. So the inductive hypothesis is proven. The claim follows trivially, setting $i = k+2$ and plugging it into the inductive hypothesis. We get that when the last zone is filled out there are no pairs in it the right components of which have been used $k + 2$ times. Remembering that at any time there cannot be pairs with right components used more times than the number of zones that are filled out, we get that there are no more than $k+1$ pairs with equal right components. □

Here it is worth to note that when filling out zone $k+2$ the cardinality of the set on reserve is the same as the number of pairs that have right components used $k+1$ time and are crossed immediately before filling out zone $k+2$. So we use the set on reserve to compensate for them and there is no need to use these pairs any more (and it is not allowed either). And there are no more sets on reserve.

Claim 5 *For every particular zone z and row r that belongs to lane l in this zone the number of pairs whose indicator function is 1 and whose left component is equal to some row number for rows that belong to lane l in zone z is at least $2N-(n+1)$ (property 3).*

Proof. Let us fix an arbitrary zone z and row r in lane l of zone z. Let us look at the pairs that are uncrossed immediately after zone z is filled out. Their number is $2N-(n+1)$ and their right components are all different by the construction of the table. Furthermore, since the uncrossed pairs always appear to the right of the crossed pairs with the same right components during the construction and the uncrossed pairs have all different right components, it follows that these pairs are the rightmost ones with such right components in zones 1 through z. The only thing which still has to be proved is that their left components are row numbers for rows that belong to lane l of zone z. This is trivially true for $z=1$. We will prove this for $z>1$. As we already noted in the proof of claim 1, there are 2^{k+2-z} rows in every lane of zone z and there are $k+2-z$ zones to be filled out to the right of it. Each of these pairs will remain uncrossed in exactly one row belonging to lane l of zone z, because each crossing separates the uncrossed pairs of the whole lane into two disjoint sets—one that contains pairs that remained uncrossed in one of lanes of the new zone and the other that contains pairs that remained uncrossed in the other of the lanes of the new zone. And since the crossed pairs get the same left components as the unique uncrossed pair in that position in their lane and zone; and since these components are row numbers; and since every lane of zone $z-1$ includes at least two lanes of zone z for $z=2,3,\ldots,k+2$; and since claim 1 holds, these pairs can get only such left components as are row numbers for the rows in lane l of zone z. □

3.3 Back to the Proof

With these 5 claims we have finished the proof that the above algorithm produces a guiding table with all three properties. So now we end our digression and finally concentrate on defining the N functions associated with every ordered N-tuple of nonnegative integers. Recall that we assumed that F identifies the class of functions U with probability $p > p_k^n$ and at most k mind changes, and we chose an arbitrary p' such that $p > p' > p_k^n$. We now show that this leads to a contradiction, proving that U cannot be identified with any probability greater than p_k^n and at most k mind changes.

We now describe the values of the N functions $f_1, f_2, \ldots, f_N$ associated with the ordered N-tuple $(m_1, m_2, \ldots, m_N)$ in the first $2N-(n+1)$ points, i.e., define $f_r(x)$ for $x = 0, 1, \ldots, 2N-(n+1)-1$ and $r = 1, 2, \ldots, N$. We will refer to this stage of definition as phase 1. Let $(a_{r,j}, b_{r,j})$ denote the pair in the r^{th} row and j^{th} position of zone 1 in the guiding table. Then $f_r(j-1)$ is defined to be $2Nm_{a_{r,j}} + b_{r,j}$. In other words, left components of the pairs in the table tell which element of the given N-tuple to encode in the value of the function and the right components tell which code type to use for encoding. Row r in the guiding table corresponds to function f_r to be defined, not only at this phase of definition, but at any time. When the first $2N-(n+1)$ values of the functions are defined we have used all the pairs in zone 1 of the table. By property 1 of the table all the N functions are still the same, i.e., have the same initial segments. We end phase 1 of the definition by simulating strategy F on these functions setting one more of the undefined values to zero with every step of strategy F. We follow all the possible computation paths of F and wait for the first or higher hypotheses of F to reach the total probability of p'. That is, $f_r(2N-(n+1)+t)$ gets defined to zero unless F has given first or higher hypothesis with total probability of p' over all its computation paths in its first t steps, for $t = 0, 1, 2, \ldots$ and $1 \leq r \leq N$. We do it in parallel for all N functions, that is, we take N copies of strategy F and run on different functions. Since all the functions still have the same initial segments all N simulations should be identical. As soon as F conjectures the first or higher hypotheses with total probability at least p' we suspend the simulation and define the functions further as follows. For functions that correspond to rows belonging to lane l of zone 2 in the guiding table we set the next l values to be zero. That is, the first 2^k functions are defined with one more zero, the next 2^k functions get two more zeros, etc. Therefore unless we later define some more points of the first functions to be zero (but we will not) we have introduced a difference between the functions corresponding to different lanes of zone 2. Then we go on to phase 2 of the definition.

Instead of describing phase 2 of the definition we will generalize the method and describe phase z for $2 \leq z \leq k+2$. Phase z can be applied only for functions that have phase $z-1$ completed. To define the next values of function f_r we take the pairs that are in row r and zone z of the table. Suppose the function was defined up to value $f_r(w)$ inclusively during the first $z-1$ phases of definition. Then, as in phase 1, $f_r(w+j)$ is defined to be $2Nm_{a_{r,j}} + b_{r,j}$ for $1 \leq j \leq |S_{r,z}|$, where $|S_{r,z}|$ denotes the number of pairs in row r and zone z of the table and $(a_{r,j}, b_{r,j})$ is the pair located in row r and position j of zone z. Since all the functions that corresponded to the rows of the same lane in zone $z-1$ as f_r had the same initial segments when the simulation was resumed in phase $z-1$, the results of the simulation had to be identical and therefore phase z could be applied to them as well. By property 1 of the table all functions belonging to the same lane of zone z will still have the same initial segments. We end phase z of the definition by defining all the rest of the undefined values to be zero if $z = k+2$ or by resuming the simulation of strategy F in parallel for all the functions that still have the same initial segments from the place where we suspended it at the end of phase $z-1$ if $z < k+2$. And, of course, the strategy is simulated on the functions that we are defining; we are deciding on new values for the functions maybe just a little bit before strategy F needs to input them. We do it now separately for every lane in zone z where the phase z of definition is applied, since phase $z-1$ need not necessarily end simultaneously for all lanes of zone $z-1$. That is, we might be in different phases of definition for different functions. Synchronous simulating is only for the functions that still have the same initial segments. We follow all the possible computation paths of F and wait for the z^{th} or higher hypotheses of F to reach total probability at least

$$q_z = \frac{2^{z-1}(n+1)-1}{2^{z-2}(n+1)}p' - 2 + 2^{2-z}.$$

While this does not happen we define one more value to be zero with every step of F. When F conjectures the z^{th} or higher hypotheses with total probability q_z we define the respective functions for which it happens further as follows. Each lane is split into two beginning with the next zone. If row r belongs to the upper one of these lanes in zone $z+1$ we set one more value of the function f_r to be zero; if row r belongs to the lower lane in zone $z+1$ we set two more values of f_r to be zero. We do this for each f_r that F gave the z^{th} or higher results with probability q_z on. Again, this is the point where we introduce a difference between functions corresponding to different lanes (in fact an irretrievable difference will be introduced in the very beginning of the phase $z+1$ when we add nonzero values to these functions). Now we go on to phase $z+1$ of the definition.

See Figure 8 and Figure 9 for a possible situation in function definition. (Functions are represented by their values in successive points. The N-tuple elements used in values are determined by the left components of the pairs in the table. The upper indices correspond to the right components and show which code type is used to encode an element.)

Phase 1:

$$
\begin{aligned}
f_1 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 0\\
f_2 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 0\\
f_3 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 0\\
f_4 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 0\\
f_5 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 00\\
f_6 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 00\\
f_7 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 00\\
f_8 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 00\\
f_9 &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 000\\
f_{10} &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 000\\
f_{11} &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}00\ldots0\ 000\\
f_{12} &= m_4^1m_4^2m_4^3m_4^4m_2^5m_2^6m_2^7m_6^8m_6^9m_6^{10}m_6^{11}m_6^{12}m_8^{13}m_8^{14}m_{12}^{15}m_{12}^{16}m_{12}^{17}m_{12}^{18}m_{12}^{19}m_{10}^{20}m_9^{21}\underbrace{00\ldots0}\ 000
\end{aligned}
$$

Waiting for the 1^{st} hypotheses to reach p'

Phase 2:

$$
\begin{aligned}
&(f_1 \text{ continued})\ m_2^{22}m_2^{23}m_4^{10}m_4^{11}m_4^{12}m_4^{13}m_3^{14}m_3^{15}m_2^{16}m_2^{17}m_2^{18}m_1^{19}m_1^{20}m_1^{21}00\ldots\\
&(f_2 \text{ continued})\ m_2^{22}m_2^{23}m_4^{10}m_4^{11}m_4^{12}m_4^{13}m_3^{14}m_3^{15}m_2^{16}m_2^{17}m_2^{18}m_1^{19}m_1^{20}m_1^{21}00\ldots\\
&(f_3 \text{ continued})\ m_2^{22}m_2^{23}m_4^{10}m_4^{11}m_4^{12}m_4^{13}m_3^{14}m_3^{15}m_2^{16}m_2^{17}m_2^{18}m_1^{19}m_1^{20}m_1^{21}00\ldots\\
&(f_4 \text{ continued})\ m_2^{22}m_2^{23}m_4^{10}m_4^{11}m_4^{12}m_4^{13}m_3^{14}m_3^{15}m_2^{16}m_2^{17}m_2^{18}m_1^{19}m_1^{20}m_1^{21}\underbrace{00\ldots}\\
&(f_5 \text{ continued})\ m_8^{22}m_8^{23}m_7^1\ m_7^2\ m_7^3\ m_7^4\ m_7^5\ m_7^6\ m_6^7\ m_6^{15}m_6^{16}m_5^{17}m_5^{18}m_5^{19}00\ldots0\ 0\\
&(f_6 \text{ continued})\ m_8^{22}m_8^{23}m_7^1\ m_7^2\ m_7^3\ m_7^4\ m_7^5\ m_7^6\ m_6^7\ m_6^{15}m_6^{16}m_5^{17}m_5^{18}m_5^{19}00\ldots0\ 0\\
&(f_7 \text{ continued})\ m_8^{22}m_8^{23}m_7^1\ m_7^2\ m_7^3\ m_7^4\ m_7^5\ m_7^6\ m_6^7\ m_6^{15}m_6^{16}m_5^{17}m_5^{18}m_5^{19}00\ldots0\ 00\\
&(f_8 \text{ continued})\ m_8^{22}m_8^{23}m_7^1\ m_7^2\ m_7^3\ m_7^4\ m_7^5\ m_7^6\ m_6^7\ m_6^{15}m_6^{16}m_5^{17}m_5^{18}m_5^{19}\underbrace{00\ldots0}\ 00\\
&(f_9 \text{ continued})\ m_{10}^{22}m_{10}^{23}m_{10}^1m_{10}^2m_9^3\ m_9^4\ m_9^5\ m_9^6\ m_{12}^7m_{12}^8m_{12}^9m_{11}^{10}m_{11}^{11}m_{11}^{12}00\ldots\\
&(f_{10} \text{continued})\ m_{10}^{22}m_{10}^{23}m_{10}^1m_{10}^2m_9^3\ m_9^4\ m_9^5\ m_9^6\ m_{12}^7m_{12}^8m_{12}^9m_{11}^{10}m_{11}^{11}m_{11}^{12}00\ldots\\
&(f_{11} \text{continued})\ m_{10}^{22}m_{10}^{23}m_{10}^1m_{10}^2m_9^3\ m_9^4\ m_9^5\ m_9^6\ m_{12}^7m_{12}^8m_{12}^9m_{11}^{10}m_{11}^{11}m_{11}^{12}00\ldots\\
&(f_{12} \text{continued})\ m_{10}^{22}m_{10}^{23}m_{10}^1m_{10}^2m_9^3\ m_9^4\ m_9^5\ m_9^6\ m_{12}^7m_{12}^8m_{12}^9m_{11}^{10}m_{11}^{11}m_{11}^{12}\underbrace{00\ldots}
\end{aligned}
$$

Waiting for the 2^{nd} hypotheses to reach q_2

Figure 8: Defining the functions for $(m_1, m_2, \ldots, m_N)$ w.r.t. the table

Phase 3:

$(f_5 \text{ continued})\ m_5^{20}m_5^{21}m_5^{13}m_5^{14}m_5^{22}m_5^{23}m_6^{1}\ m_6^{2}m_5^{3}\ m_5^{4}\qquad 00\ldots0\ 0$
$(f_6 \text{ continued})\ m_5^{20}m_5^{21}m_5^{13}m_5^{14}m_5^{22}m_5^{23}m_6^{1}\ m_6^{2}m_5^{3}\ m_5^{4}\qquad \underbrace{00\ldots0}\ 00$
$(f_7 \text{ continued})\ m_8^{20}m_8^{21}m_8^{8}\ m_8^{9}\ m_7^{10}m_7^{11}m_7^{12}m_8^{7}m_8^{15}\ m_7^{16}m_7^{17}00\ldots$
$(f_8 \text{ continued})\ m_8^{20}m_8^{21}m_8^{8}\ m_8^{9}\ m_7^{10}m_7^{11}m_7^{12}m_8^{7}m_8^{15}\ m_7^{16}m_7^{17}\underbrace{00\ldots}$

Waiting for the 3^{rd} hypotheses to reach q_3

Phase 4:

$(f_5 \text{ continued})\ m_5^{5}m_5^{6}m_5^{8}\ m_5^{9}\ m_5^{10}m_5^{11}m_5^{12}m_5^{7}\ m_5^{15}m_5^{16}\qquad 00\ldots$
$(f_6 \text{ continued})\ m_6^{5}m_6^{6}m_6^{17}m_6^{18}m_6^{19}m_6^{20}m_6^{21}m_6^{13}m_6^{14}m_6^{22}m_6^{23}00\ldots$

Figure 9: Continuation of Figure 8

So we have described in detail the definition of the functions $f_1, f_2, \ldots, f_N$ associated with every ordered N-tuple $(m_1, m_2, \ldots, m_N)$. Let us now consider functions $h_1, h_2, \ldots, h_N$, where h_r is the Gödel number of f_r, for $r = 1, 2, \ldots, N$. That is, $\varphi_{h_r} \equiv f_r$, or in other words h_r is a program for a Turing machine that computes f_r. In order to simplify things we may assume that h_r is our description of how to define f_r. We can now assume that the functions $h_1, h_2, \ldots, h_N$ are functions of arguments $m_1, m_2, \ldots, m_N$, since our definition of $f_1, f_2, \ldots, f_N$ was dependent on $m_1, m_2, \ldots, m_N$. Now let us use Smullyan's lemma. It says that there exist positive integers $m'_1, \ldots, m'_N$ such that $\varphi_{m'_1} \equiv \varphi_{h_1(m'_1, \ldots, m'_N)}, \ldots, \varphi_{m'_N} \equiv \varphi_{h_N(m'_1, \ldots, m'_N)}$. That is, $\varphi_{m'_1} \equiv f'_1$, $\varphi_{m'_2} \equiv f'_2$, $\ldots$, $\varphi_{m'_N} \equiv f'_N$, where we denote by $f'_1, f'_2, \ldots, f'_N$ the functions corresponding to the ordered N-tuple $m'_1, m'_2, \ldots, m'_N$. So it appears that in the values of these particular functions we have not encoded just some meaningless numbers from the N-tuple but quite frequently the Gödel numbers of these functions. And that is why all these N functions belong to class U, which we now prove.

Claim 6 *Functions $f'_1, f'_2, \ldots, f'_N$ belong to class U.*

Proof. Let us look at any of these functions, for instance, f'_r. Let us assume that it has gone through z phases of definition, where $1 \leq z \leq k+2$, and not more. That is, we have used the pairs from row r in the table up to zone z inclusively to define its values and further it contains only zeros. Let row r belong to lane l in zone z of the table. The table is built in such a way that all rows in some lane of some zone are contained in some bigger lane of the previous zone, or, in other words, each lane of zone i is split into two or more lanes in zone $i+1$, for every $1 \leq i \leq k+1$. From this fact and property 1 it follows that every function that corresponds to some row belonging to lane l of zone z is identical to f'_r. For $z = k+2$ there is only one function in each lane of zone z, that is, $r = l$. For $z \leq k+1$ that is because the functions have the same initial segments at any moment of definition and the simulation of strategy F on them is done synchronously. If F did not give the expected results (in terms of probability over all the computation paths) on f'_r it would not give them on other functions corresponding to rows of lane l in zone z. Synchronicity, in fact, is not important; it is mentioned just to make the definition more understandable. The strategy has to give the same results over all the computation paths on functions with the same initial segments. Since the functions that correspond to rows of lane l in zone z are the same they must have the same Gödel numbers. That is, any Gödel number of one of these functions is a Gödel number for the others as well. From property 3 it now follows that for the function f'_r there are at least $2N-(n+1)$ different code types such that the last encoded value in each code type is a Gödel number for f'_r. Property 2 insures that no more than $k+1$ values are encoded in each code type. Therefore f'_r belongs to class U. And since we picked an arbitrary function f'_r, all N functions $f'_1, f'_2, \ldots, f'_N$ belong to class U. □

The only thing that remains to be proven is that at least one of the functions $f'_1, \ldots, f'_N$ cannot be identified correctly with probability p' and at most k mind changes. We assumed that F succeeds for any function of class U at the very beginning of this proof. We will prove that for some two functions we always get to phase $k+2$ of the definition (except for the case $k=0$ which is special in that there are $n+1$ functions) and that at least one of them cannot be identified with probability p and at most k mind changes although it belongs to the class U.

Let us look at the definition of these functions at phase 1, particularly when simulating F on the initial segments of them. Suppose F never gives its first or higher hypotheses with total probability p' over all the computation paths on the initial segments of these functions, that is, we never get to phase 2 of the definition. Then all the functions are continued with zeros to infinity and turn out to be exactly the same. We already know that they belong to class U and so strategy F has to give some results with total probability p. So we can conclude that at some finite time F

gives the first or higher results with probability $p_1 \geq p'$ on these initial segments.

First we have to look at the case when $k = 0$, that is, when phase 2 amounts to just using all the pairs from zone 2 of the table and then setting all the remaining values to be zeros. In this case we have obtained $n+1$ different functions that all belong to class U. There is at least one of these functions such that at most $\frac{p_1}{n+1}$ of the results of F can be correct for this function, since F gives these results before any difference among the functions is introduced, i.e., these results are the same for all $n+1$ functions. This function will be the one that is not identified with probability p. Suppose later strategy F gives more correct results for this function and the total of them over all computation paths reaches p.

But since in this case only one hypothesis is allowed for F (no mind changes), it can give no more than $1 - p_1$ additional correct results. Hence, remembering that $p' \leq p_1$:

$$\frac{p_1}{n+1} + 1 - p_1 \geq p > p',$$
$$p' + \frac{n}{n+1} p_1 < 1,$$
$$\frac{2n+1}{n+1} p' < 1,$$
$$p' < \frac{n+1}{2n+1} = p_0^n.$$

But $p' > p_k^n$, so in case $k = 0$ we have got a contradiction to the assumption that F identifies U with probability $p > p_k^n$ and at most k mind changes.

Let us now assume $k \geq 1$ and look at the remaining phases of the definition. We will inductively prove that for at least 2 functions we get to phase $k+2$ of the definition. For the base case we use the fact that we always get to phase 2 of the definition and that strategy F at that time has given its first or higher hypotheses with total probability $p_1 \geq p'$ over all its computation paths. Now let us assume that for the functions corresponding to the rows of some lane l of zone z, where $1 \leq z \leq k$, the strategy F has given its z^{th} or higher results with total probability $p_z \geq q_z$ over all the computation paths (for completeness we set $q_1 = p'$). We will look only at the functions corresponding to the rows of that lane and prove that for the functions that correspond to the rows of some particular lane of zone $z+1$ we get to phase $z+2$ of the definition, i.e., this is our inductive hypothesis. At phase $z+1$ we use the pairs of zone $z+1$ to define the next values of the functions and then resume the suspended simulation of F on these functions. Functions that correspond to different lanes of zone $z+1$ in the table are already made different. But the functions corresponding to the same lane of zone $z+1$ are still the same (i.e., their initial segments are the same). We are simulating F on them and waiting for it to give its $z+1^{st}$ or higher hypotheses with total probability at least q_{z+1}. Suppose this never happens for any of these functions. Then all the functions are continued with zeros to infinity and the ones belonging to the same lane in zone $z+1$ are exactly the same (by property 1).

First we will prove that this is not possible in the case when $z=1$. In this case we have $n+1$ different functions that belong to class U. Therefore total probability of correct results of F over all the computation paths on any of them has to reach p' in some finite time. Since the first or higher results that F conjectures with probability p_1 are the same for the functions corresponding to different lanes of zone 2 (and therefore different) there must be at least one lane of zone 2 such that the functions corresponding to it have at most $\frac{p_1}{n+1}$ (in terms of the probability over all the computation paths) correct results among these. For any function there could be still at most $1-p_1$ correct first hypotheses (in terms of probability over all the computation paths) and some number of second hypotheses. From the assumption that F identifies every function with probability p there should appear at least

$$p'-(1-p_1)-\frac{p_1}{n+1}=p'+\frac{n}{n+1}p_1-1\geq\frac{2n+1}{n+1}p'-1=q_2$$

correct second or higher hypotheses at some finite time. So in the case when $z=1$ our assumption that it is possible for F not to give its second or higher results with probability q_{z+1} over all the computation paths on any of these functions has failed. Therefore there will be at least one lane of zone 2 such that for functions corresponding to the rows of this lane we always get to phase 3 of the definition.

Now we will prove the impossibility of this for $2\leq z\leq k$. In this case we look at the functions that correspond to the rows of lane l of zone z. We know that they belong to class U. Therefore the total probability of correct results of F over all the computation paths on any of them has to reach p eventually. Since the functions that correspond to rows of different lanes in zone $z+1$ are different, one of these lanes is such that the functions corresponding to it have at most $p_z/2$ (in terms of the probability over all the computation paths) correct z^{th} or higher results among those that F conjectured at the end of phase z (which were the same for all the functions belonging to the rows of the same lane in zone z). Even if all the other z^{th} or lower hypotheses were correct for these functions or eventually changed to such it would add at most $1-p_z$ correct results (in terms of probability over all the computation paths). Hence, over all the computation paths of F on these functions there have to appear at some finite time $z+1^{st}$ or higher hypotheses with total probability at least

$$p' - (1 - p_z) - p_z/2 = p' + p_z/2 - 1 \geq p' + q_z/2 - 1 =$$

$$= p' + \frac{2^{z-1}(n+1) - 1}{2^{z-1}(n+1)} p' - 1 + 2^{1-z} - 1 =$$

$$= \frac{2^z(n+1) - 1}{2^{z-1}(n+1)} p' - 2 + 2^{1-z} = q_{z+1}.$$

Therefore our assumption that F never gives its $z + 1^{st}$ or higher hypotheses with total probability at least q_{z+1} has failed for $2 \leq z \leq k$ as well and we will get to phase $z + 2$ of the definition.

So we have proved by induction that for the functions that correspond to the rows of some lane l in zone $k+1$ we will reach phase $k+2$ of the definition. Let us assume that F conjectured its $k + 1^{st}$ hypotheses with probability

$$p_{k+1} \geq q_{k+1} = \frac{N-1}{N/2} p' - 2 + 2^{1-k}$$

at the end of phase $k+1$. There are exactly two rows in every lane of zone $k+1$ (since we now consider only $k \geq 1$) and we will look at both functions that correspond to the rows of lane l of zone $k + 1$. At phase $k + 2$ they both get values according to the pairs of zone $k+2$ in their respective rows and the rest of their values are defined to be zero. We know that both functions belong to class U and so F should identify them with total probability p over all the computation paths. At least one of these functions has at most $p_{k+1}/2$ (in terms of the probability over all the computation paths) correct $k + 1^{st}$ results among those that F conjectured at the end of phase $k+1$ (which were the same for both functions). Let us assume that for this function all the hypotheses of lower order are correct or will eventually change to correct ones. In any case, it cannot have more than $1 - p_{k+1}$ (in terms of probability over all the computation paths) correct results in addition to those conjectured at the end of phase $k + 1$. Suppose that these results suffice for F to identify this function with probability p. Then:

$$1 - p_{k+1} + p_{k+1}/2 \geq p > p',$$

$$1 - p_{k+1}/2 > p',$$

$$1 - q_{k+1}/2 > p',$$

$$p' < 1 - \frac{N-1}{N} p' + 1 - 2^{-k},$$

$$p' + \frac{N-1}{N} p' < 2 - 2^{-k},$$

$$\frac{2N-1}{2N}p' < 1 - 2^{-k-1},$$

$$p' < \frac{2N(1-2^{-k-1})}{2N-1},$$

$$p' < \frac{2^{k+1}(n+1)(1-2^{-k-1})}{2N-1},$$

$$p' < \frac{2^{k+1}(n+1)-(n+1)}{2N-1},$$

$$p' < \frac{2N-(n+1)}{2N-1} = p_k^n.$$

But $p' > p_k^n$, so this is a contradiction to our assumption that strategy F identifies class U with probability $p > p_k^n$ and at most k mind changes and the theorem is proved. □

4 Acknowledgements

This work is a translation and revision of the first author's senior thesis at the University of Latvia in 1991. The second author was his advisor there. We are very grateful to Dana Angluin[1] for her advice and help with writing it up in English. We acknowledge NSF grant CCR-9014943 for support for preparation and distribution of this paper.

5 Conclusions

The results obtained in this work are summarized in Figure 1. At first we hoped to prove the generalization of theorem 3 in [1] for the critical probabilities in Figure 1. However, the recent work of Daley and Kalyanasundaram [3] shows that this is not possible. In particular, they prove the following theorem that does generalize theorem 3 of [1].

[1] Department of Computer Science, Yale University

Theorem (Daley and Kalyanasundaram[3])

a) For all integers $n \geq 0$, $k \geq 0$, $EX_k(\frac{(2^{k+2}-3)n+1}{(2^{k+2}-2)n+1}) \subset EX_k(\frac{(2^{k+2}-3)(n+1)+1}{(2^{k+2}-2)(n+1)+1})$*;*

b) For all integers $n \geq 0$, $k \geq 0$, *if* $\frac{(2^{k+2}-3)(n+1)+1}{(2^{k+2}-2)(n+1)+1} < p \leq \frac{(2^{k+2}-3)n+1}{(2^{k+2}-2)n+1}$, *then* $EX_k(p) \subseteq EX_k(\frac{(2^{k+2}-3)n+1}{(2^{k+2}-2)n+1})$.

Note that the critical probabilities for 0 mind changes in this theorem are the same as in Figure 1 and that all of the critical probabilities for $k \geq 1$ mind changes in this theorem (i.e., $\frac{(2^{k+2}-3)+1}{2^{k+2}-2)+1}, \frac{(2^{k+2}-3)2+1}{2^{k+2}-2)2+1}, \frac{(2^{k+2}-3)3+1}{2^{k+2}-2)3+1}, \ldots, \rightarrow \frac{(2^{k+2}-3)}{2^{k+2}-2)}$) are contained between the two critical probabilities $\frac{2^{k+2}-2}{2^{k+2}-1}$ and $\frac{2^{k+1}\cdot 3-3}{2^{k+1}\cdot 3-1}$ for $k \geq 1$ mind changes in Figure 1. This suggests that quite possibly the situation is more complicated than one might first think.

References

[1] R. Freivalds. Finite identification of general recursive functions by probabilistic strategies. In L. Budach, editor, *Proceedings of the Conference on Algebraic, Arithmetic, and Categorial Methods in Computation Theory*, pages 138–145, 1979.

[2] Smullyan R. M. Theory of formal systems. *Annals of Mathematics Studies*, 47(7), 1961.

[3] Daley R. and Kalyanasundaram B. Capabilities of probabilistic learners with bounded mind changes. Extended abstract, 1992.

[4] R. Wiehagen, R. Freivalds, and E. Kinber. On the power of probabilistic strategies in inductive inference. *Theoretical Computer Science*, 28:111–133, 1984.