Computer Science Through Urn Games: An Unified Framework for a Hierarchy of Solvable and Unsolvable Problems

Sorin Istrail
Department of Computer Science
Brown University

Dedicated to Alan Turings 100th Birthday and to the memory of Professor Edsger W. Dijkstra

Abstract. In his last published paper, Solvable and Unsolvable Problems, printed in 1954 in the popular journal Science News, Alan Turing presented several elegant puzzles aiming at explaining and popularizing problems for which there are algorithms for solutions – the solvable – as well as some for which no such algorithmic solution exists the unsolvable. This paper could be seen as a continuation of Turings aim to explain and popularize through puzzles, this time using a set of computational problems of various computational difficulties. Similar to Turings paper, where all his puzzles are unified as substitution puzzles, our set of puzzles offers instances of a unified approach based on urn games. Our (m, n1, n2) games have urns of two types: set urns and linear urns; the urns contain balls of m colors; in a move, a number n1 of balls are extracted and a number n2 of balls are returned to the urn; the solitary player performs moves, one after another, based on the rules of the game until no rule can be applied. Our urn games include Turings substitutions puzzles, but defined with different goals. The class of computational problems obtained by varying the urn game parameters (m, n1, n2) turns out to form a hierarchy of complete problems, one for each of the complexity classes NL, P, NP, PSPACE, EXPSPACE, and the unsolvable. Dijkstras game is a (2,2,1) urn game.

The urn games are generalizations of a silly game E.W. Dijkstra presented in his paper Why correctness must be a mathematical concern [Inaugural Lecture for the Chaire Internationale dInformatique at the Universite de Liege, Belgium, 1979 (EWD720)]. The generalizations are inspired by discussions I had with Professor Dijkstra in the wake of my somewhat critical comments of his game. [See my articles Criticizing Professor Dijkstra Considered Harmless and the followup, When Professor Dijkstra Slapped Me in the Quest for Beautiful Code http://www.cs.brown.edu/ sorin/non-tech-writing.htm.]

Dijkstra's silly game is shown to have a certain *incompleteness* property. This incompleteness relates to the apparent inseperability of two problems: (a) demonstrate how to predict the final outcome, and (b) demonstrate that the final outcome is completely predictable. It turns out that predictability is equivalent to associativity and to the existence of logical invariants for correctness proofs.

We analyse the game and some natural variants inspired by the quest for understanding its incompleteness. It will turn out that a complementary problem, the *Unpredictability of a given instance*, offers a pure combinatorial, (i.e., machine-independent) introduction of computational complexity classes. The game and its variants are disguises of decision problems of generic computational difficulty: *Directed graph accessibility, Context-free grammar membership, Satisfiability of propositional Boolean formulas, Context-sensitive grammar membership, Uniform word problem for commutative semigroups, Recursive-enumerable grammar membership.* Indeed, the unpredictability problem for the game and its natural generalizations, turns out to provide us with a hierarchy of complete problem for the complexity classes **NL, P, NP, PSPACE, EXPSPACE**. The last mentioned disguise brings the complexity status of the problem to unsolvable.

1 A beautiful problem of Dijkstra that fails to be a well-defined puzzle

Let us give the author's description of the problem [2]

Consider the following silly game to be played by a single person with an urn and as many white balls and black balls as he needs. To begin with an arbitrary positive number of balls is put into the urn, and as long as the urn contains two or more balls, the player repeats the following move: he shakes the urn and, without looking, he takes two balls from the urn; if those two balls have the same color he throws one black ball back into the urn, otherwise he returns one white ball into the urn. Because each move decreases the

total number of balls in the urn by 1, the game is guaranteed to terminate after a finite number of moves, and it is not difficult to see that the game ends with exactly 1 ball in the urn. The question is: "What can we say about the color of that final ball when we are given the initial contents of the urn?"

I have read for the first time the problem in D. Gries' monograph *The Science of Programming*. By following the advise of the book, I have spent 10 minutes, trying to solve the problem, before reading the solution. But neither did a solution come nor did I really start to solve it. I spent these 10 minutes trying to answer a different question: Was the problem question well-defined? In fact, I tried to convince myself that having started with an initial content of the urn, the color of the final ball would be unique, i.e., would not depend on the order in which I have picked up the balls. It was clear that there were many ways to follow, but it was unclear whether *all the roads lead to Rome*!
The problem question seems to indicate that the color of the final ball is *predictable* when the initial content of the urn is given.

Then puzzled by the puzzle I read the solution.

"*Looking at the three single moves possible*,

1. $\circ\circ \to \bullet$
2. $\bullet\circ \to \circ$
3. $\bullet\bullet \to \bullet$

we observe that the last two leave the number of white balls in the urn unchanged, while the first move reduces the number of white balls in the urn by 2. In other words, each move leaves the so-called parity of the number of white balls in the urn unchanged: an even number of white balls in the urn remains even, and an odd number of white balls in the urn remains odd. In short: if the initial number of white balls is even, the final ball is black, and if the initial number of the white balls is odd, the final ball is white. And that answers the question!

Note that this single argument settles the question for all initial contents of the urn, and per initial contents for all of the perhaps many possible games."

I realized that the invariant pointed out by the solution assured the predictability property of the color of the final ball. I wondered whether in order to show that the problem is well-defined we have to solve it! This paper is about this *incompleteness* property of the problem of Dijkstra. The *incompleteness* relates to the apparent inseperability of two problems: (a) demonstrate how to predict the final outcome, and (b) demonstrate that the final outcome is completely predictable.

Even after seeing the solution, I couldn't explain in a crystal clear way why the final color was unique; available seems to be only an *aposteriori* proof. After a little bit of search I discovered the reason of my previous feelings. Consider the following *bad* game given by the rules:

1. $\circ\circ \to \bullet$
2. $\circ\bullet \to \bullet$
3. $\bullet\bullet \to \circ$

for which the final outcome is *unpredictable* for some initial contents of the urn. Indeed, consider the urn having the following composition $\{\bullet \circ \bullet\}$. Then the following two plays of the bad game end up in balls of different colors:

$\{\bullet \circ \bullet\} \rightarrow^{rule3} \{\circ\circ\} \rightarrow^{rule1} \{\bullet\}$
$\{\bullet \circ \bullet\} \rightarrow^{rule2} \{\bullet\bullet\} \rightarrow^{rule3} \{\circ\}$

What is the property that distinguishes *good* games from *bad* games?

2 Predictable Games

Let us denote $\bullet$ by 0, $\circ$ by 1 and let f be a function $f : \{0,1\} \times \{0,1\} \rightarrow \{0,1\}$ function. Let $Ball$=$\{0,1\}$.

Definition 1 f-terms *and their* values, *are defined inductively.*

- *If b is in Ball then b is an f-term; its value is val(b)=b.*
- *If t_1, t_2 are f-terms, then so is $f(t_1, t_2)$; the value of t=f(t_1, t_2) is*

$$val(t) = f(val(t_1), val(t_2)).$$

Let S be a sequence of balls of positive size; denote its size by $|S|$. Our move will be different this time: we will pick two consecutive f-terms t_1 and t_2, remove them and put in their place the f-term $f(t_1, t_2)$.

Definition 2 *Let us define a sequence* f-game *as follows: Given an initial sequence S of f-terms of positive size, the player repeats the following move as long as the sequence has size two or more: he takes two consecutive f-terms from the current sequence, say t_1, t_2 and returns in their place in the sequence the f-term $f(t_1, t_2)$. Denote the move (t_1, t_2) on S by $S \rightarrow_f S'$, where S_1, S_2 are sequences such that $S = S_1 t_1 t_2 S_2$ and $S' = S_1 f(t_1, t_2) S_2$. Let $\rightarrow_f^*$ be the reflexive transitive closure of $\rightarrow_f$. A play of the sequence f-game on S is given by*

$$S \rightarrow_f^* t$$

The play ends up with a sequence having only one term t, called the final f-term of the play.

Definition 3 *The sequence f-game is* predictable on S *if for every sequences S_1, S_2, whenever*

- $S \rightarrow_f^* S_1, S \rightarrow_f^* S_2, i = 1, 2$*and*
- $|S_1| = |S_2| = 1$

it follows that $val(S_1) = val(S_2)$.
The sequence f-game is predictable if for every sequence S it is predictable on S.
Theorem 1 *For every function f, the sequence f-game is predictable if f is associative.*

Proof. Let us consider a play $S \rightarrow_f^* t$ of the f-game on an initial sequence of balls S. If we ignore the commas, parentheses, and the 'f' symbols in the final term t, what is left is the initial sequence of balls. Call this sequence the ball sequence of the f-term.
As a corollary, different initial sequences will give rise to plays having (*syntactically*) different final f-terms (which may, or may not, be *semantically* equivalent, i.e., terms with the identical value). Consider now, for a fixed S, all the plays on S having the same final term t.
The final term of the play t is the same for all the plays that have the same set of moves made at identical positions in the corresponding sequences, but performed in possible different order. Therefore an f-term is an equivalence class for plays that differ in the arbitrary choice imposed on *independent* moves. This abstraction clears the way of understanding predictability.
Indeed, a play on sequence $S = b_1 b_2 ... b_n$ provides with a final term t, which is an arrangement of parentheses

on S, giving rise to an expression defining and element of *Ball*. Conversely, any arrangement of parentheses in the sequence S can be converted into an f-term which is a final term for a class of plays.
Therefore we can conclude that predictability on S is exactly the requirement that all the f-term having ball sequence S must have identical value.
This means exactly that f is associative on S. Now predictability means associativity on $\square$
Let us remark that predictability is also equivalent to another interesting property.

Theorem 2 *The following are equivalent.*

1. *The sequences f-game is predictable.*
2. *There exists an Invariant for the f-game.*

Proof. Let us say that predicate ϕ over the set of sequences is an Invariant for an f-game if the following holds: whenever $S \models \phi$ and $S \rightarrow_f S'$ follows that $S' \models \phi$.

Let us consider a directed graph $G = (V, E)$ that has the set of ball sequences as vertex set V and $\rightarrow f$ as edges. It is easy to see that the f-game is predictable if the graph has exactly two connected components: one containing the vertex 0 and the other one containing the vertex 1. Indeed, predictability means that from every sequence S we can reach exactly one sequence of size one. Let V_0 be the set of vertices connected by a directed path to 0. Similarly for V_1. The $\{V_0, V_1\}$ is a partition of the set of ball sequences exactly when the game is predictable. Describing one of these two sets, say V_0 by a predicate ϕ_0 gives us an Invariant for the game. Indeed, in this case we have that $S \models \phi_0$ and $S \rightarrow fS^1$ implies $S^1 \models \phi_0$.

Among the 16 binary boolean functions there are only 6 that give rise to predictable games, i.e., they are associative. They are

1. $f_1(x, y) = 0$, Invariant ϕ_0: *All sequences*
2. $f_2(x, y) = 1$, Invariant ϕ_0: *No sequence*
3. $f_3(x, y) = \bar{x}y \vee x\bar{y}$, Invariant ϕ_0: *The number of white balls is even*
4. $f_4(x, y) = xy \vee \bar{x}\bar{y}$, Invariant ϕ_0: *The number of black balls is even*
5. $f_5(x, y) = xy$, Invariant ϕ_0: *One or more black balls*
6. $f_6(x, y) = x \vee y$, Invariant ϕ_0: *One or more white balls*

Let us consider now the *set f-games*, the generalization of the game of Dijkstra. We return to urns, that is, the balls initially are given as a set. In order to have f-games we need *commutative* functions f.
The analog of the $f - term$ is now called *commutative f-term* which is a class of f-terms.

Definition 4 Commutative f-terms *and their values are defined inductively.*

- $If b is in Ball then b is a commutative f - term; its value is val(b) = b.$
- $If t_1, t_2 are commutative f-terms, then so is f\{t_1, t_2\}; the value of t = f\{t_1, t_2\} is val(t) = f(val(t_1), val(t_2)).$

Let U be an urn having a positive number of balls; denote its size by $|U|$. The move of the game is the following: we will pick two commutative f-terms t_1 and t_2, remove them and put in their place the f-term $f\{t_1, t_2\}$.

Definition 5 Let us define an urn f-game as follows: Given an initial urn U of commutative f-terms of positive size, the player repeats the following move as long as the urn has size two or more: he takes two commutative f-terms from the current urn, say t_1, t_2 and returns in the urn the commutative f-term $f\{t_1, t_2\}$. Denote the move $\{t_1, t_2\}$ on U by $U \to fU'$, where $U' = (U - \{t_1, t_2\} \cup \{f\{t_1, t_2\}\}$. Let $\to *_f$ be the reflexive transitive closure of $\to_f$. A play of the urn f-game on U is given by

$$U \to^* f\{t\}$$

The play ends up with an urn having only one commutative f-term t, called the final commutative f-term of the play.

Definition 6 The urn f-game is predictable on U if for every urns U_1, U_2, whenever

- $U \to_f^* U_1, U \to_f^* U_2, i = 1, 2$ *and*
- $|U_1 = \{t_1\}, U_2 = \{t_2\}$

it follows that val(t_1)=val(t_2).
The urn f-game is predictable if for every urn U it is predictable on U.

Theorem 3 *For every function f, the following are equivalent:*

1. *The urn f-game is predictable.*
2. *The function f is associative.*
3. *There exists an Invariant for the urn f-game.*

Let us return to the original game. Dijkstra's ur f-game uses the commutative function fs. fs is the sum modulo 2 of the two arguments. Its associativity is responsible for the fact that the game is predictable.

3 The Complexity of Unpredictability

We are going to analyze the difficulty of prediction on a given urn or sequence. Moreover, the reductions of the game to different problems of generic computational difficulty suggest slight generalizations of the game that complete the complexity picture. If we play the game by taking out m balls and returning n balls back, i.e., using n functions o farity m, then we call this game the (m, n)-game. Let (2,2) stand for $\{(2, 1), (2, 2)\}$ and (2,2) stand for $\{(1, 2), (2, 2), (2, 1)\}$. In a similar way we can define the (2,2)-game and the (2,2)-game. Let us remark that there may be infinite plays in some generalized versions of the games.

THE UNPREDICTABILITY PROBLEM

Given: A set of functions described by a, and a sequence S (urn U).
Question: Does there exist two plays of the a-game ending up in different balls?

3.1 The Complexity of Unpredictability for Games on Sequences

The sequence f-games, (which are sequence (2,1)-games in this new terminology) are reminiscent to Chomsky's context-free grammars and the parsing problem for them. Let $G = (V_N, V_T, x_0, F)$ be a context-free grammar in Chomsky normal-form. We are going to present three reductions of the Membership problem for grammars to the Unpredictability problem for the sequence a-games. We give the reduction of the Membership problem for context-free grammars to the Unpredictability problem for the sequence (2,1)-game. The

other two reductions are similar.

We can associate a function f to the grammar. $f : K \times K \to K$ be defined as follows.
Let $K = V_N \cup V_T \cup \{\sharp, g, G\}$ and

- $f(y, z) = x$ if $x \to yz \in F$
- $f(x, \sharp) = \sharp = f(\sharp, x)$, for $x \in K - \{G\}$
- $f(x_0, g) = G$
- $f(\sharp, G) = G$
- $f(u, v) = \sharp$, otherwise

Lemma 1 *Let $w \in (V_N \cup V_T)^+$. The f-game played on the sequence $\sharp wg$ is unpredictable if $w \in L(G)$.*

Theorem 4

1. *The unpredictability problem for the sequence (2,1)-game is complete for* P.
2. *The unpredictability problem for the sequence (2,2)-game is complete for* PSPACE.
3. *The unpredictability problem for the sequence (2,2)-game is unsolvable.*

Proof

1. The membership problem for context-free grammars is complete for P [3].
2. The membership problem for context-sensitive grammars is complete for PSPACE [3].
3. The membership problem for recursive-enumerable grammars is unsolvable [3].

3.2 The Complexity of Unpredictability for Games on Sets

First let us consider the (2,1)-games. The following problem will be used in the reduction.

DIRECTED GRAPH ACCESSIBILITY PROBLEM (GAP)

Given: A directed graph $G = (\{1, 2, ..., n\}, A)$.
Question: Does there exist a path from vertex 1 to vertex n?

Theorem 5 *The unpredictability problem for the urn (2,1)-games is* NL*-hard.*

Proof We are going to reduce the GAP to the unpredictability problem for the urn (2,1)-games. Let $g-(\{1, 2, ..., n\}, A)$ be an instance of GAP. Take $K = \{1, ..., n, \bar{1}, ..., \bar{n}, b, in, out, g\}$ and define $f : K \times K \to K$ by

- $f(in, 1) = \bar{1}$
- $f(\bar{i}, j) = \bar{j}$, if $(i, j) \in A$
- $f(\bar{n}, out) = g$
- $f(g, x) = g, x \in \{1, ..., n, b\}$ item $f(x, y) = b$, otherwise.

Consider now the initial urn U to be

$$U = \{1, ..., n, b, in, out\}.$$

The possible final balls are the $g(good)$ and $b(bad)$. While b is always a final ball, we have that

there is a play having g as a final ball if G has a path from vertex 1 to vertex n.

It is easy to see that the reduction can be done in log space.

We are now considering the complexity of unpredictability for the urn $\{(2,1),(2,2)\}$-game. The decision problem we use in order to classify its complexity is the following.

SATISFIABILITY OF PROPOSITIONAL BOOLEAN FORMULAS (SAT)

Given: A set of variables U and ϕ a Boolean formula over U.
Question: Does there exist a truth statement to the variables from U which satisfies ϕ?
Theorem 6 *The unpredictability problem for the urn* $\{(2,1),(2,2)\}$*-game is* NP*-complete.*

Proof Without loss of generality we can consider ϕ in conjunctive normal form. We can view now ϕ as being a set of clauses C that should be simultaneously satisfied. Again without loss of generality, each clause may be restricted to have exactly 3 literals.
Let (U, C) be an instance of SAT. We are going to associate an instance of our game as follows. Let $U^z = \{x^z | x \in U\}, z = 0, 1$. For each clause $c \in C, c = y_1 \vee y_2 \vee y_3$, defined $= [y_1 \vee y_2 \vee y_3], c_1^1 = [1], c_1^0 = [y_2 \vee y_3], c_{12}^1 = [1], c_{12}^0 = [y_3]$. Let $\tilde{c} = \{c^{'}, c_1^1, c_1^0, C_{12}^1, c_{12}^0\}$ and

$$K = U \cup U^0 \cup U^1 \cup \{0, 1, b, g, G, [0], [1], \} \cup (\cup_{c \in C} \tilde{c}).$$

We now define the function

$$f : K \times K \to (K \times K) \cup K$$

as follows.

1. $f(x, z) = x^z, z \in \{0, 1\}$
2. $f(x^z, z) = \{x^z, x^z\}, z \in \P 0, 1\}$
3. $f([y_1 \vee y_2 \vee y_3], x^z) = a$. If $y_1 = x$ and $z = 1$ then $a = [y_2 \vee y_3]$; the other cases are similar
4. $[y_2 \vee y_3], x^z) = a$. If $y_2 \bar{x}$ and $z = 0$ then $a = [1]$; the other cases are similar
5. $f([y_3], x^z) = a$. If $y_3 = x$ and $z = 1$ then $a = [1]$; the other cases are similar
6. $f([z_1], [z_2]) = [z_1 * z_2], z_1, z_2 \in \{0, 1\}$.
7. $f([1], g) = G$
8. $f(G, z) = G, z \in \{0, 1\}$
9. $f(p, q) = b$, otherwise

Now we take as an initial set of balls $K = U \cup \{0, ..., 0, 1, ..., 1, \} \cup (\cup_{c \in C} c^{'}) \cup \{g\}$. Let us analyze the game. By our construction, the final ball can be b, meaning *badway* and G meaning *there exists a truth assignment making C true.* While b is always a final ball of a play, G will be the final ball of a play if C is satisfiable. Let us explain how the reduction works.

- By 1 the variables from U may receive truth values x^z.

- Using 2 copies of the balls created in 1 are provided in sufficient quantity, that is, for every occurrence of x in C, a corresponding x^z is available.
- Rules $3, 4$ and 5 simulate the assignment in the first, second and third position in the clause.
- The conjunction of the Boolean values of the clauses is realized by 6.
- If the result obtained in the previous step is [1] then rule 7 provides the ball G which remains alone after the elimination, by 8, of all the remaining 1s and 0s. Note that G can eliminate only 1s and 0s; therefore at this moment the entire C should be evaluated.
- Any other move provides as result the ball b which once present cannot be eliminated.

It is easy to see that the problem is in NP and the reduction can be done in log space.

Word problems can also be defined using Dijkstra-like games.
Let us define the *uniform word problem for commutative semigroups* [1]. A *semigroup presentation* S is given by a set of equations of the form $a_i = b_i, 1 \leq i \leq n$, where a_i, b_i, are words over an alphabet V. A presentation S for a commutative semigroup is a set of equations with the property that for every $x, y \in V$ we have in S the equation $xy = yx$. We define $a \rightarrow b$ if $((a = ca_id, b = cb_id)$ or $(a = cb_id, b = ca_id))$ and $a_i = b_i$ is an equation of the semigroup presentation. Let $\rightarrow^*$ be the reflexive transitive closure of $\rightarrow$ Put $a \equiv b$ if $a \rightarrow^* b$.
The *uniform word problem for commutative semigroups* is the following:
Given: A semigroup presentation S and words a, b.
Question: Is $a \equiv b$ in S?
Cardoza, Liptom and Meyer [1] showed that this problem is complete for EXPTIME.

Theorem 7 *The unpredictability problem for the urn* $[2, 2]$*-game with reversible moves is* EXPTIME*-complete. (Reversible games are games in which each move has its reverse move also legal.*

Proof First of all, without loss of generality we can reduce the problem to $[2, 2]$-games by considering the semigroup presentation as a recursive-enumerable commutative grammar, and reducing it to normal form [3].

Now to obtain the result we can just simulate the equations of the semigroup presentation with rules in the game. We can start with an urn containing a. If b is reached after a sequence of moves, a special final ball is produced. Any wrong attempt generates a bad final ball. The details are similar to that of the previous proof.

4 Acknowledgements

The author wants to thank Professor E. W. Dijkstra for comments on a previous draft and encouragements. It is a pleasure to thank Lew Robertson and Jacques Weinstein for suggestions and inspiring conversations about the problem.

References

[1] E. Cardoza, R. Lipton, A.R. Meyer, *Exponential Space Complete Problems for Petri Nets and Commutative Semigroups*, ACM Annual Symposium on Theory of Computing, 1980, pp. 1-5
[2] E. Dijkstra, *Why Correctness Must Be a Mathematical Concern*, The Correctness Problem in Computer Science, Boyer and Moore eds., Academic Press 1981
[3] J. Hopcroft, J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979