A note on Rubinstein's "Why are certain properties of binary relations relatively more common in natural language?"

Beard, Rodney

University of Alberta

2001

A note on Rubinstein's "Why are certain properties of binary relations relatively more common in natural language?" *

Rodney Beard
Department of Marketing, Business Economics and Law
School of Business
University of Alberta T6G 2R6
Canada
e-mail: Rodney.Beard@ualberta.ca

*I would like to thank Ariel Rubinstein and Sharon O'Donnell for helpful comments. All errors are my own.

Abstract

This note examines the complexity of complete transitive binary relations or tournaments using Kolmogorov complexity. The complexity of tournaments calculated using Kolmogorov complexity is then compared to minimally complex tournaments defined in terms of the minimal number of examples needed to describe the tournament. The latter concept is the concept of complexity employed by Rubinstein [6] in his economic theory of language. A proof of Rubinsein's conjecture on the complexity bound of natural language tournaments is provided.

JEL Classification: C60,D89

1 Introduction

Ariel Rubinstein's work on the applications of binary relations in natural language raises interesting questions both for the economics of language and for the uses of natural language in everyday economic activity. In particular, many forms of economic interaction involve the use of natural language. Certain types of legal contracts such as employment contracts, sales contracts, etc. employ natural language in their formulation.

Rubinstein analyses complete and transitive binary relations or tournaments [7]. He argues that tournaments that are less complex than others are more likely to give rise to behavioral regularities. Rubinstein suggests measuring the complexity of the tournament in terms of the minimum number of examples necessary to describe the tournament. An alternative approach has been suggested by the philosopher Johann van Benthem [1] this employs Rissannen's [5] idea of the minimal description length of a model. This idea is related to Kolmogorov complexity. Van Benthem's suggestion is interesting because it provides a ready way of calculating the complexity of a tournament that is well grounded in information theoretic principles. This approach has to the best of the author's knowledge not been constructively explored.

In this note Rubinstein's theory is further developed within the framework of Kolmogorov complexity theory. A proof of Rubinstein's conjecture on the complexity bounds is suggested and some further results relating Rubinstein's concept to Kolmogorov complexity is addressed. This note may also be interpreted as a partial response to van Benthem's suggestion for using minimal description length as criteria.

In section 2 the theoretical framework and some essential definitions and notations from Kolomogorov complexity theory that are employed in the derivation of results. In section 3 Rubinstein's conjecture and a proof of the conjecture are presented. In section 4 an algorithm for computing the Kolmogorov complexity of a tournament is suggested and two propositions relating Rubinstein and Kolmogorov complexity are presented and discussed. Finally conclusions are drawn in section 5 and possible extensions suggested.

2 Background

One approach to the complexity of a tournament would be to pose the question: How does one determine a winner? Another would be to ask "How do we rank the elements of a tournament?" Each of these questions could in principle be examined from an algorithmic perspective. In other words one can identify an algorithm that computes the winner of a tournament and one that ranks the elements of a tournament. An example of this can be found in social choice theory, imagine an algorithm that determines the Condorcet winner and another algorithm that ranks societal alternatives. Each of these algorithms takes a certain number of steps to compute. The complexity of a tournament could then be measured in terms of the number of steps it takes to solve each of these tasks. If the number of steps it takes to compute a winner is bounded by some polynomial function of the steps.

The complexity of a tournament may be approached from two perspectives: on the one hand the complexity of a tournament could be examined as a graph theoretic entity in terms of complexity classes (P or NP). Secondly, a tournament may be approached from perspective of Kolmogorov complexity using Turing machines. From an algebraic perspective tournaments are complete asymmetric binary relations. Alternatively, tournaments may be represented as a graph. In terms of graph theory a tournament is a directed graph such that for any two vertices u and v either (u, v) or (v, u) but not both is an arc of the tournament [2, p. 326.].

Before proceeding to the main contribution of the paper it is first necessary to introduce some basic notation. Most notation follows Rubinstein [**?**] and new notation is only introduced as necessary. Fistly, define an alphabet Ω as a set of symbols representing objects. A set of I agents is assumed to communicate in natural language about these objects using the alphabet. The reader is referred to Rubinstein [**?**] for examples.

Rubinstein defines a binary relation as follows:

Definition 1. *(Binary relation with respect to f) $(f, \{a_i, Rb_i\}_{i\in I})$ defines the binary relation (R^* on $A \subset \Omega$) when*

- *f is a sentence in predicate calculus*
- *R^* is the unique binary relation on Ω satisfying f and $\forall i$ $a_i R^* b_i$ is true.*

The complexity $C(R^*)$ of the binary relation R^* may be computed in a number of ways. Rubinstein [7] suggests that the complexity of a tournament is given by the solution of the following optimization problem:

$$\min_{R^*\in T} C(R^*)$$

where T is the set of tournaments and $C(R^*) = \{I|R^* \in T\}$.

An alternative method of computing the complexity of a tournament has been suggested by Van Benthem [1]. This involves employing Solomonoff-Kolmogorov complexity theory and the minimum description length principle [5] to formalize the notion of a tournaments complexity. In the following this idea will be explored to some extent but the the theory will not be dveloped to the extent of introducing a minimum description length characterization of the complexity of tournaments. Instead it will be shown how Rubinstein's complexity concept may be formulated in terms of Solomonoff-Kolmogorov complexity before proceeding to prove Rubinstein's conjecture on the complexity bound of a tournament.

Definition 2. *Solomonoff-Kolmogorov complexity Given a message x the complexity of x is given in general by*

$$C_f(x) = \min\{l(p) : f(p) = n(x)\}$$

where p is some non-negative integer, $f(p)$ a partial function defined over the integers and $n(x)$ the size of x where $n : X \to N$, and $l(p)$ the length of p.

One way to think of this is to think of p being a program and $f(p)$ a Turing machine [3],[4, p. 94].

If $f(p)$ is considered to be a Turing Machine then the Kolmogorov complexity may be expressed as:

$$C(x) = \min \{l(T) + C(x|T) : T \in \{T_0, T_1, \ldots\}\} + O(1)$$

where T is a Turing machine.

To compute the Kolmogorov complexity of a tournament one need only write a program to compute the minimum number of binary relations needed to identify all objects in Ω. But this is just a minimal spanning tree problem for a digraph with equal weights on each edge with the one modification that cycles are allowed (see section 4). So the complexity of a tournament reduces to the problem of computing the Kolmogorov complexity of a minimal spanning algorithm for a tournament, i.e. the task is to find a minimal spanning subgraph that is both complete and asymmetric. The key point is that one no longer requires transitivity so that the digraph is not necessarily a tree.

Before moving on to a discussion of what such an algorithm might look like, a conjecture of Rubinstein regarding the lower bound of the complexity of a tournament is proved [6, 7].

3 Rubinstein's conjecture

The economics of language has lead to a number of interesting results including the following conjecture by Rubinstein [6, 7]:

Conjecture 1. *Rubinstein's Conjecture Let ϕ be a sentence in the predicate calculus language which includes a single name of a binary relation, R. then there exists n^* such that for any $|\Omega| \geq n^*$ and any tournament R which is defined by the sentence ϕ, $C(R) \geq |\Omega| - 1$.*

Note that $|\Omega|$ is the order of the tournament p and that $C(R)$ is the size q of the tournament. The first theorem of digraph theory relates these two concepts [2, p. 32]. This theorem states that given a digraph D of order p and size q, with $V(D) = \{\nu_1, \ldots, \nu_p$, then $\sum_{i=1}^{p} od\nu_i = \sum_{i=1}^{p} id\nu_i = q$, where $od\nu_i$ is the outdegree of vertex ν_i and $id\nu_i$ is the indegree of vertex ν_i. Clearly, the validity of Rubinstein's conjecture depends on the degree of the graph.

Proof. (of Rubinstein's conjecture) Call the first vertex the start vertex then

$$od\nu_1 + \sum_{i=2}^{p-1} od\nu_i = q$$

if all vertices labelled 2 to $p-1$ are identical in degree then

$$C(R) > |\Omega| - 1$$

To see this note that $C(R) = od\nu_1 + od\nu(|\Omega| - 1)$ so that the strict inequality holds for $od\nu > 1$ and $od\nu_1 = 0$ or for $od\nu_1 \geq 1$.

Furthermore, the out-degree of any vertex must be a non-negative integer so that this strict inequality will hold for all-graphs regular or otherwise, unless the arbitrarily chosen start vertes has out-degree zero and all other vertices have an out-degree at each vertex of at least 1. Note that strict equality only holds if one vertex has out-degree zero and all others have out-degree 1. So that for a graph with any non-negative out-degree at each vertex $C(R) \geq |\Omega| - 1$. Note that for a graph to be a tournament $n^* > 0$ (there are no tournaments of order zero). □

Note that that a strict equality will hold for the transitive case.

4 An algorithm for computing the Kolmogorov complexity of a tournament

Instead of using a Turing machine to compute the Kolmogorov complexity, a specific algorithm that could be implemented by a Turing machine will suffice.

A starting point for the development of an algorithm for computing the Kolmogorov complexity of a tournament is the following modification of Kruskal's algorithm for mimimal spanning tree problems [2, p. 88]:

1 Initialize the set $E \leftarrow \emptyset$ of edges of some graph (V, E)

2 Increment E pick any $\epsilon \in E$ (note all weights are equal and normalized to 1) such that $e \notin E$ and $E \cup \{e\}$ is complete and asymmetric but not necessarily acyclic, then let $E \leftarrow E \cup \{e\}$

- If $|E| \geq p-1$ where p is the order of the graph, then output E and output $|E|$, the size of the graph.

This algorithm computes the subdigraph which minimally spans a given set of vertices. The Kolmogorov complexity of this algorithm is then the minimum number of steps that the program needs to compute the size of the minimal spanning subdigraph.

In what follows the fact that any string can be suitably encoded is utilized to establish a relationship between Rubinstein and Kolmogorov complexity. This comparison is a partial response to van Benthem's suggestion of using the minimal description length criteria. This is because minimal description length is based on kolmogorv complexity. A detailed discussion of coding theory is however omitted, the reader is referred to Li and Vitanyi for this [4].

Proposition 1. *(Relationship between Rubinstein and Kolmogorov complexity) The Kolmogorov complexity $C(x)$ is less than or equal to the Rubinstein complexity $n(x)$ plus a constant.*

Proof. The algorithm suggested above computes the Rubinstein complexity for an arbitrary set of vertices by calculating $|E|$ the number of steps needed to do this is just $l(p)$. If the source alphabet is just the set of vertices then the size of the spanning digraph $|E| = n(x)$ which is the Rubinstein complexity of the tournament. Note that this is just the order of the digraph minus 1. On termination of the program $f(p) = n(x) = |E|$ and the minimum number of steps required to do this will be $l(p)$ or the Kolmogorov complexity. The order of the digraph is by definition $l(x)$. From the previous theorem it is known that $n(x) \geq l(x) - 1$. Finally, from [4, p. 100] theorem 2.1.2 it is known that $C(x) \leq l(x) + c$. Combining these facts, rearranging and substituting gives $C(x) \leq n(x) + 1 + c$. □

Note that $n(x) = |\Omega|$ by definition. Based on the proceeding result relating to the relationship between Kolmogorov complexity and Rubinstein complexity the following corollary may be stated.

Proposition 2.

$$C(R) \geq |\Omega| - 1 \geq C(x) - 2 - c$$

Proof. Follows by substitution and proposition 1 and proposition 2. □

Rissannen [5] has developed the idea of the minimum description length as a model selection criteria for statistical models. To relate the notion of a statistical model back to the preceeding discussion, graph theoretic analysis would need to be extended to a random graph setting. This implies a stochastic rank ordering. So that Benthem's suggestion of using minimal description length as an alternative complexity measure should not be taken too literally. So while theoretically a tournament could be constructed on a random graph and minimal description length could be employed to evaluate the complexity of binary relations. Rubinstein complexity is simpler and more intuitively appealing.

5 Conclusion

This note discusses the relationship between Kolmogorov complexity and Rubinstein complexity for tournaments. A number of results are proven showing the relationship between Kolmogorov complexity and Rubinstein complexity for tournaments. The main purpose of this paper has been to explain the connection between these two notions of complexity concept and to provide a proof of Rubinstein's conjecture regarding a complexity bound for binary relations in natural languages. Some possible extensions are to exlore the connection between Rubinstein's concept of indication friendliness and unique decodability in coding theory and to develop a more explicit comparison of minimal description length and Rubinstein complexity.

References

[1] J.van Benthem (2000) Comments on A. Rubinstein's "Economics and Language", in: A.Rubinstein, *Economics and Language*, Cambridge University Press, Cambridge, 2000.

[2] G. Chartrand, O.R. Oellermann *Applied and Algorithmic Graph Theory*, McGraw-Hill, New York 1993.

[3] J.E. Hopcroft and J.D. Ullmann, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, New York 1979.

[4] M. Li and P. Vitanyi *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag, Berlin 1997.

[5] J. Rissanen *Stochastic complexity in statistical inquiry*, World scientific, Singapore, 1989.

[6] A. Rubinstein (1996) Why are certain properties of binary relations relatively more common in natural language? *Econometrica* 64, pp.343-356.

[7] A. Rubinstein *Economics and Language*, Cambridge University Press, 2000.