Uniform Formalisation and Verification of STV Algorithms

Milad K. Ghale Dirk Pattinson

Research School of Computer Science, ANU, Canberra

Abstract. We introduce a framework which offers uniform formalisation and verification of some properties for various Single Transferable Voting (STV) algorithms. These algorithms, although different from one another in some ways, have key properties in common which make them STV instances. By abstracting away common features of particular STV schemes, we obtain minimum requirements to accept a given, arbitrary scheme as a legitimate STV instance. We formally prove that any STV scheme which meets the expectations of the above conditions satisfies some mathematical properties, such as termination. We demonstrate extensibility and concision of our framework by modular formalisation of a divergent range of STV algorithms in the theorem prover Coq. Then, by the built-in mechanisms of Coq, for each of the modules, we extract a certifying executable programme into the Haskell programming language. The certificate produced upon each execution, is a visualisation of the trace of the computation carried out to obtain an end result for an election instance. It provides us with an independently checkable proof of tallying correctness to establish count-as-recorded subproperty of the universal verifiability index. Finally we show effectiveness of our approach by evaluating the executables on some real-size elections.

1 Introduction

Single Transferable Voting (STV) algorithms are preferential schemes for counting votes, where voters express which candidates they prefer more by numerically ranking them on election ballots. Despite variations in STV algorithms, closer examination of particular instances, reveals a common underlying structure existing in all of them. By abstracting these recurrent patterns away, we obtain a base which underpins what it takes for an algorithm to fit into STV class.