Formal Methods at *Galois* and *Free & Fair*

DeepSpec Kick-off Meeting

Joe Kiniry and Rob Dockins
Galois* and *Free & Fair

June 2016

Galois in a Nutshell

FREE & FAIR | galois |

- *Galois* is a private R&D employee-owned corporation with offices in Portland, OR and Alexandria, VA (next to DARPA)
- Galois was co-founded around 16 years ago by John Launchbury; he remains our Chief Scientist but is currently on leave as Director of I2O at DARPA
- Galois has around 60 employees, around 50 of which are researchers, most of which are computer scientists and mathematicians who do formal methods and more
- most of Galois's work is on BAAs, SBIRs, IDIQs, etc. for various federal agencies; a growing portion is for industry
- Galois's is small and flexible when it comes to collaborations and IP arrangements (licensing & spin-outs)
- Galois has spun-out five companies, including *Free & Fair*

Free & Fair in a Nutshell

FREE & FAIR | galois |

- *Free & Fair* is a public benefit, employee-owned corporation
- Free & Fair's is to be the "Red Hat of Digital Elections"
- Free & Fair is developing high assurance, open source elections technologies for public good
- around a dozen current and past Galois employees have worked for Free & Fair over the past two years
- Free & Fair is developing several products in the digital elections space, including a supervised voting system (but no IV)m all of which are based on peer-reviewed research
- Free & Fair's principles focus on transparency and trust—all pricing is simple and public, all technology is open source and freely available, all proposals are public
- Free & Fair is breaking and reinventing the elections market

Formal Methods at Galois

FREE & FAIR | galois |

- formal methods of various kinds is woven into the majority of proposals and projects at Galois
- our core wheelhouses are programming languages, formal verification, cyber-physical systems, and cryptography
- the most common use of formal methods is in programming language design and realization (DSLs and EDSLs)
- automated solvers, mainly SAT and SMT, are used often
- logical frameworks (mainly Coq, Isabelle, and PVS) are used when appropriate, primarily to mechanize models or semantics and to reason about systems that are being built using a correct-by-construction approach
- prototypes created for customers are most often written in Haskell and have formal models built using other systems

Formal Methods at *Free & Fair*

FREE & FAIR | galois |

- *Free & Fair*—as the child of Galois and my research group—uses formal methods for its product development
- Free & Fair uses a larger set of approaches and technologies than Galois, including not only LFs and advanced PLs and verification tools, but also old school formal methods (e.g., B, BON, OBJ, RAISE, TLA+, VDM, and Z), model finders and checkers of various kinds (e.g., Alloy and FDR), unusual PLs (e.g, Eiffel, Haskell, and SPARK), protocol specification and verification tools (e.g., F* and Tamarin)
- Free & Fair, like Galois, works directly with the federal agencies responsible for the certification of cryptographic modules and elections systems on helping define next generation certification standards using formal methods

Why do Elections need Formal Methods?

FREE & FAIR | galois |

History	**Opportunity**
Threat	**Assurance**

Technology and Products

FREE & FAIR | galois |

- polling place process analysis
- digital ballot distribution
- remote ballot marking
- voter-verifiable vote-by-mail
- electronic poll books
- supervised voting systems
- optical scan voting systems
- tabulation
- risk-limiting auditing
- election auditing

Our First Case Study: The F&F Tabulator

FREE & FAIR | galois |

- a tabulator computes the outcome of an election by reading cast vote records and writing an election result
- cast vote records are digital interpretations of (typically) paper ballot records
- votes are tabulated according to an election scheme
- there are dozens of elections schemes in use around the world; the U.S.A.'s plurality election scheme is a rarity
- other common election schemes are rank choice voting, approval voting, proportional representation through single transferable vote, list-based schemes, etc.
- non-plurality election schemes are notoriously hard to get right in election law, and every tabulation system ever rigorously examined has been flawed

Our First Case Study: The F&F Tabulator

FREE & FAIR | galois |

- the Free & Fair Tabulator computes the outcome of an election using either a plurality or the San Francisco variant of rank choice voting
- we have precisely translated, line for line, election law into its a formal model as a specification
- truths stipulated in law are encoded as either definitions or theorems, as appropriate, and some general theorems of social choice theory are also mechanized
- an implementation proven to conform to the specification is also encoded, and extract an implementation in Haskell
- the Haskell implementation is rigorously tested using sample election data and QuickCheck and is integrated with our optical scanning tabulator, OpenCount