

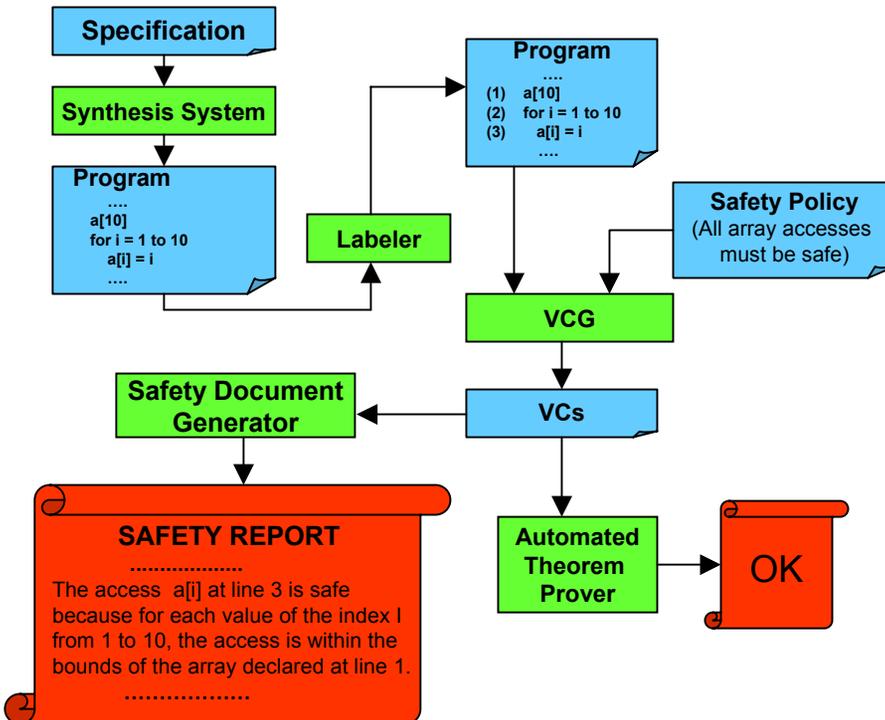
Automating the Documentation and Certification of NASA Software



PROBLEM

Traditional certification techniques are laborious and time-consuming. Formal certification uses theorem provers to automatically generate certificates of correctness in the form of formal mathematical proofs, but

- Can we trust the theorem-prover?
- How do we make these proofs human-readable?
- How do we relate these proofs to the program?



SOLUTION

Generate textual explanations of code safety from auto-generated proof obligations and trace these back to the program. These proof obligations refer to an explicit safety policy that can be varied. For example:

- Array-bounds safety,
- Variable initialization-before-use,
- Variable write limits for volatile memory, ...

TECHNOLOGY

- We have developed a generic safety document Generator that automatically generates explanations of program safety from verification conditions (VCs), formulas produced by a Verification Condition Generator (VCG).
- It is generic in the sense that new safety policies can easily be added to the system.
- A significant step in the direction of merging formal certification with traditional certification.

Explanation of Accomplishment

- **POC:** Ewen Denney (ASE Group, Code IC, edenney@email.arc.nasa.gov)
- **Collaborator:** Ram Prasad Venkatesan (Univ. Illinois at Urbana-Champaign)
- **Funding:** ITSR, QSS Summer Intern Program
- **Background:** The ASE group is developing automated program synthesis systems for the application domains of data analysis (AutoBayes) and state estimation (AutoFilter). We have previously extended these systems with an automated certification capability, based on mathematical logic, for various safety policies. However, it is very difficult for humans to interpret the resulting proofs and then relate them to the original program. We have addressed this by incorporating a *safety documentation* feature.
- **Accomplishment:** We have developed a tool that can automatically generate textual explanations of safety with respect to a given safety policy for auto-generated code. Our tool currently generates safety documents for two safety policies: **array-bounds safety and variable initialization-before-use**. Our framework, however, is generic in that new safety policies can easily be incorporated. Another increment over previous work is that we now provide a mechanism to trace proofs of program safety back to the program itself. A paper describing the work was presented 07/13/2004 at *Algebraic Methodology and Software Technology 2004* (AMAST'04) in Stirling, Scotland.
- **Shown:** The code generated by the synthesis engine is labeled with statement numbers and this is fed to the verification condition generator (VCG) along with a safety policy. The VCG generates verification conditions (VCs), which are passed to the document generator. The safety document generator then generates textual explanations of safety from these VCs. Finally, these VCs are checked automatically by a theorem prover.
- **Benefits:** This technology has the potential to increase confidence in the use of code generators within and outside NASA. Auto-generated code, in addition to a certificate of correctness (w.r.t. user-defined notions of safety) will come with a document containing human-readable explanation as to why it is correct. This approach is a significant step in the direction of merging formal certification with traditional certification.