



Formalization of



Sneha Popley

Computer Science

Dr. Rhonda Hatcher

Dr. Dick Rinewalt

Dr. Antonio Sanchez

Honors Committee

Dr. Jonathan Aldrich

Robert Simmons

School of Computer Science

Carnegie Mellon University



How?

- Summer Research at UPenn
- What next?
- Emailed Dr. Aldrich at Carnegie Mellon University
- Had NEVER met him and there were NO job openings



Motivation

- A flaw in TurboTax ¹
- Toyota recalls Prius ²
- The list goes on...

¹ Kocieniewski, David. *Taxpayer could have kept the \$600, but he put his country first.* The New York Times. April 6, 2010.

² The Associated Press. *Toyota promises prompt response on quality issues.* The New York Times. March 30, 2010.



Motivation

- Software plays an integral role in our lives
- Its reliability and accuracy is invaluable to us
- Operations are complex and depend on complicated data handling



A Solution

- Use a strongly formalized language to build software systems.
- My project involved formalizing one such language: **SASyLF**



What is Formalization?

- A formal definition provides foundation for
 - Building a community for users
 - Proving **properties** of the language and programs written in the language
- **It is essential to**
 - **A complete understanding of the behavior of the language**
 - **Convince others about soundness of design**
 - **Improve reliability of any program in the language**



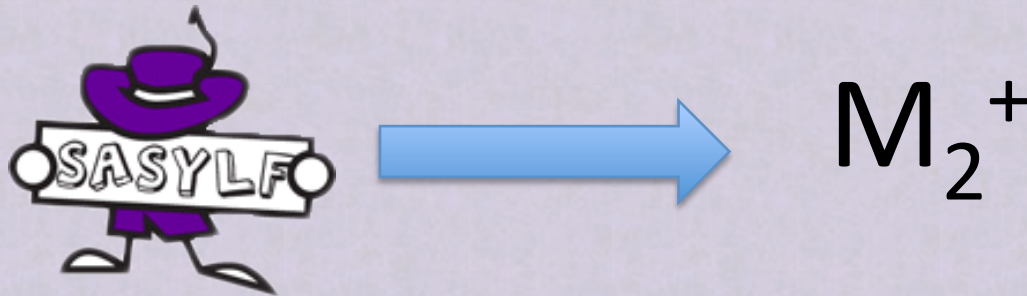
What is SASyLF?

- A programming language to teach graduate level computer science concepts
 - Gentle learning curve
 - Familiar syntax
 - Explicit notation
 - Minimal math context
 - Local checking and localized errors



Goal

- Show relationship between SASyLF and a representation of its underlying logic, M_2^+



- **Create a translation between the two languages**
- M_2^+ is a sound logic that proved as an inspiration for SASyLF, but it wasn't mathematically written down.



More on M_2^+

- M_2^+ is a sound logic to **represent and analyze** programming language concepts

- These concepts include theoretical, mathematical ideas behind the behavior of programming languages

$$\begin{aligned}\Sigma &= n:\text{type}, \\ & z:n, \\ & s:n \rightarrow n,\end{aligned}$$

Definition of natural numbers

- Examples: properties of integers, lists, groups, etc



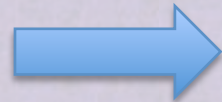
The Process

- Create an **abstract representation** of all possible programs in SASyLF
- **Compare it** to the abstract representation of all possible programs in M_2^+
- Write down equations that **relate** the two representations
- But, it can't be that simple...

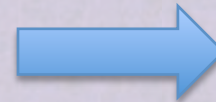


The Process

- SASyLF is **semantically too different** from M_2^+ to write down such equations
- So, we introduced a Core Calculus that acts as a transition between SASyLF and M_2^+

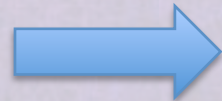


Core
Calculus

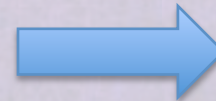


M_2^+

The Lion
King



Simba's
dad's death



Animation
Process



Conclusions

- Formal proof that SASyLF is based on a sound logic, M_2^+
- Basis for **further development** of SASyLF
 - More features
 - Added robustness
- Formal proof of the **reliability** of SASyLF
- Suitable to teach graduate-level computer science concepts
- Possibility of future formalization of SASyLF in other logic systems such as Beluga and Delphin



Questions ?