Substructural Type Systems

John Albers

Substructural Type Systems Walker

Substructural Type Systems 101

@inbook{walker, place={Cambridge, MA}, title={Substructural Type Systems}, booktitle={Advanced Topics in Types and Programming Languages}, publisher={The MIT Press}, author={Walker, David}, pages={3–43}}

Summary: The first chapter in Pierce's Advanced Topics in Programming Languages gives a beginner friendly introduction to substructural type systems. The chapter opens with the discussions of Structural properties (i.e. Exchange, Weakening, and Contraction), and the results of tuning these properties. He also gives the motivation for substructural type systems as a way to control memory resources. Then Walker introduces an easy to follow linear type system. There are examples and exercises that show exactly what the type system offers. After developing an intuition about the type system, Walker introduces a type system that can be algorithmically checked. Finally, he goes on to show the operational semantics of a λ-calculus boolean language with this type system and discusses various extensions and variations.

Evaluation: The first 2 sections of the chapter are the most important. These sections introduce Substructural type systems very gently, and made the rest of the chapter much more understandable. The examples do a very good job motivating such type systems, but also justifying some of the rules of the type system that I found confusing or questioned. There is no research contribution here and rather acts as a introduction to substructural type systems

Linear Types Can Change the World! Wadler

Substructural Type Systems Academic 101

@inproceedings{Wadler_1990, title={Linear Types Can Change the World!}, booktitle={Programming Concepts and Methods}, publisher={North}, author={Wadler, Philip}, year={1990} }

Summary: This paper introduces linear type systems by using Linear Logic as a starting point. Wadler motivates Linear Type systems by showing how linear types would help solve problems in programming languages when dealing with resources. Wadler first gives a nod to Jean-Yves Girard who is credited with discovering linear logic. Then he introduces a type system with linear types (values that can be used exactly once) and unrestricted types (values who can be used any or none amount of times). He goes on to describe the linear calculus and the read/write problem. Finally, he discusses how to extend the linear calculus with arrays.

Evaluation: While this paper could be viewed as a the introduction of linear logic to programming languages, the paper demands a fairly deep background in logic. I found the paper quite difficult to read, but after reading Walker's chapter in Advanced Topics in Types and Programming Languages, the material became more grokable. Wadler's contributions are obviously very impressive, but this paper demands slow reading.

Practical Affine Types Tov and Pucella

Affine Types are Actually Usable!

@article{Tov_Pucella_2011, title={Practical affine types}, volume={46}, ISSN={03621340}, DOI={10.1145/1925844.1926436},number={1}, journal={ACM SIGPLAN Notices}, author={Tov, Jesse A. and Pucella, Riccardo}, year={2011}, month={Jan}, pages={447} }

Summary: This paper introduces Alms as a usable programming language that has affine types. Affine types were mentioned in Walker's chapter as types of values that could be used 0 or 1 times. There had been complaints in the past about the widespread and explicit annotation that must take place for such a language. The Author includes a short tutorial for programming in the language to show the language is indeed practical for programmers. Alms addresses concerns about explicit annotation by adding some type inference.

Evaluation: This paper gives a real example of a Substructural Type Systems in a usable language. The contribution of this work is to show that a language with support for affine types can be practical and is not just some academic hogwash. Alms can be used and experimented with. I found the most helpful section of the paper to be the part that talks about learning Alms. The later sections do tend to get pretty math heavy towards the end.