Declarative platform-agnostic dependently typed build and deployment configuration languages for operating systems

SEMC, Nexus Aurora

***Abstract*—In the modern computing industry, operating systems are one of the most widely-configured parts of a project. Their management of numerous key computers components - such as input/output, memory, and packages - makes them a beehive for never-ending configuration. Without a unified system, information about current configuration can be lost over time. New members looking at a deployed OS are very unlikely to catch every modification at first glance, software updates are likely to introduce new configuration requirements, and making even small changes in general can be a complete mess. This paper introduces Salo: a declarative, dependently-typed configuration language that eliminates this issue entirely. With a language such as Salo, changes to configuration are centralized in one set of declarations, solving all of the previously-mentioned issues with operating system maintainership - and more.**

***Index Terms*—Language design, dependent types, configuration languages, type theory.**

I. Introduction

TODO: Intro

II. Language

A. Syntax

TODO: Syntax

B. Type System

Naming conventions:

- Lowercase Latin letters (e.g. x, y): variables
- Capital Latin letters (e.g. A, B): types
- Capital Greek letters (e.g. Γ, Δ): contexts

1) Zero type:

$$\frac{}{\vdash \varnothing : \text{Type}}$$

$$\frac{\Gamma \vdash p : \varnothing, \quad \Gamma \vdash C : \text{Type}}{\Gamma \vdash \text{abort}(p) : C}$$

2) Unit type:

$$\frac{}{\vdash \text{unit} : \text{Type}}, \quad \frac{}{\vdash () : \text{unit}}$$

$$\frac{\Gamma \vdash C : \text{Type}, \quad \Gamma \vdash c : C, \quad \Gamma \vdash p : \text{unit}}{\Gamma \vdash \text{triv}(p, c) : C}$$

$$\frac{\Gamma \vdash C : \text{Type}, \quad \Gamma \vdash c : C}{\Gamma \vdash \text{triv}(\text{tt}, c) \equiv c : C}$$

3) Product types:

$$\frac{\Gamma \vdash A : \text{Type}, \quad \Gamma \vdash B : \text{Type}}{\Gamma \vdash A \times B : \text{Type}}$$

$$\frac{\Gamma \vdash a : A, \quad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B}$$

$$\frac{\Gamma \vdash C : \text{Type}, \quad \Gamma \vdash p : A \times B, \quad \Gamma, x : A, y : B \vdash c : C}{\Gamma \vdash \text{unpack}(p, c) : C}$$

$$\frac{\begin{array}{c}\Gamma \vdash C : \text{Type}, \quad \Gamma \vdash a : A, \\ \Gamma \vdash b : B, \quad \Gamma, x : A, y : B \vdash c : C\end{array}}{\Gamma \vdash \text{unpack}((a, b), c) \equiv c[a/x, b/y] : C}$$

4) Sum types:

$$\frac{\Gamma \vdash A : \text{Type}, \quad \Gamma \vdash B : \text{Type}}{\Gamma \vdash A + B : \text{Type}}$$

$$\frac{\Gamma \vdash a : A, \quad \Gamma \vdash B : \text{Type}}{\Gamma \vdash \text{inl}(a) : A + B}, \quad \frac{\Gamma \vdash A : \text{Type}, \quad \Gamma \vdash b : B}{\Gamma \vdash \text{inr}(b) : A + B}$$

$$\frac{\begin{array}{c}\Gamma \vdash C : \text{Type}, \quad \Gamma \vdash p : A + B, \\ \Gamma, x : A \vdash c_A : C, \quad \Gamma, y : B \vdash c_B : C\end{array}}{\Gamma \vdash \text{case}(p, c_A, c_B) : C}$$

$$\frac{\begin{array}{c}\Gamma \vdash C : \text{Type}, \quad \Gamma \vdash a : A, \\ \Gamma, x : A \vdash c_A : C, \quad \Gamma, y : B \vdash c_B : C\end{array}}{\Gamma \vdash \text{case}(\text{inl}(a), c_A, c_B) \equiv c_A[a/x] : C}$$

$$\frac{\begin{array}{c}\Gamma \vdash C : \text{Type}, \quad \Gamma \vdash b : B, \\ \Gamma, x : B \vdash c_A : C, \quad \Gamma, y : A \vdash c_B : C\end{array}}{\Gamma \vdash \text{case}(\text{inr}(b), c_A, c_B) \equiv c_A[a/x] : C}$$

5) Anonymous function types:

$$\frac{\Gamma \vdash A : \text{Type}, \quad \Gamma \vdash B : \text{Type},}{\Gamma \vdash A \to B : \text{Type}}$$

$$\frac{\Gamma, x : A \vdash y : B}{\Gamma \vdash \lambda(x)(y) : A \to B}$$

$$\frac{\Gamma \vdash f : A \to B, \quad \Gamma \vdash a : A}{\Gamma \vdash f\ a : B}$$

$$\frac{\Gamma, x : A \vdash b : B, \quad \Gamma \vdash a : A}{\Gamma \vdash \lambda(x)(b)a \equiv b[a/x] : B}$$

C. Standard Library

TODO: Standard library

III. Operating System Building

TODO: Building an OS.

IV. Operating System Deployment

TODO: Deploying an OS.

V. Conclusion

TODO: Conclusion

Acknowledgment

TODO: Acknowledgment

References

[1] E. Dolstra, M. de Jonge, and E. Visser, *Nix: A Safe and Policy-Free System for Software Deployment*, Utrecht, Netherlands: Utrecht University, November 14-19 2004.