syntelos

Sunday, 9 October 2022

CPL types

Declarations, definitions, and typedefs constitute the facilities of the C programming language (CPL) in proximity to mathematical type theory. This observation frames the distance between practical machine type theory and mathematical abstraction.

The mechanical conception of types served the metaphysical portability of compilers and their objective languages. The academic dimension of languages and systems is incremental. The practical dimension is spatiotemporal. Abstraction risks both spacetime and conception.

CPL types achieved requisite properties of binding and facility. The abstraction owed to formalization remains an open problem, today.

In C++, Bjarne Stroustrup proposed an exhaustion of the problem space. The automated reference type of C++ contradicts the abstract reference type of Java.

The former violates the meaning of the term in the case of the latter.

Spatial types distinct from temporal types is a hallmark of the historic situation. The CPL problem space is not as advanced. The C++/Java problem space represents an era of evolution. The popularity of these languages is representative of the era in which the relationship between the mechanical data type and a logical information type has been meandering through a variety of conceptual spaces. The formality of an abstract mathematical type is alluded to in many technology domains. Hardware description is where capital is most dependent on verification. Telephony is the historic concentration of investment in substance and significance.

The relationship between data and information is traditionally conceptual. A practice of comprehension is necessary to the exercise of comprehension necessary to understanding.

Formalizing the relationships between data and information occurs variously. In the physical sciences the measurement device and technology are the founding frame. In the metaphysical sciences a relatively anthropological frame is ultimately foundational to the problem space. These relationships have many expressions, from natural language to logical language. Bridging the gaps among mechanical and logical languages includes establishing logical expression in many cases.

To facilitate this understanding, we must recognize the mathematical distinction of data and logic. Mathematics handles these issues without much confrontation. Whether the mechanical object is data or information is rare, indeed.

/syntelos /science /computing /machinery: CPL types

John Pritchard, @syntelos

https://docs.google.com/document/d/1xyxOOX9Ea1v_uokVR3SnvgwYauWqLI07akFPpuUrtoY/edit?usp=drivesdk

This work is licensed under a Creative Commons Attribution- NonCommercial- ShareAlike 4.0 International License http://creativecommons.org/licenses/by-nc-sa/4.0/