
Functions, Abstractions and Idealizations in the Explanation of Computing Systems Behaviour

Nicola Angius*^{†1}

¹University of Sassari – Italy

Abstract

The issue of explanation in Computer Science arises in connection with the problem of stating why a computational system displayed an execution which is not correct with respect to the system's requirements. In the process of designing, specifying, programming, and implementing computational artefacts, computer scientists are admittedly involved in multiple, hierarchical descriptions of the systems to be realized. Providing an explanans to an occurred miscomputation implies identifying the description(s) against which the artefact is not correct and tracing back the corresponding error state to fix the system accordingly. It is argued here that providing explanations in computer science is a pragmatic activity (van Fraassen 1980): depending on the context wherein the explanatory request is advanced, different contrast-classes showing allowed executions of the system under inquiry may be provided. For each of those contrast-classes, a relevance relation should be specified which expresses why the observed computation was executed instead of any other in the contrast-class. This talk is turned to show how, depending on the description pragmatically selected, the corresponding relevance relation provides explanantia that fulfil divergent models of scientific explanation. It is argued that semantic specifications in the form of some state transition system used in formal verification allow for nomological explanations of observed executions. Functional analyses of high-level language, assembly language, and code-machine language programs are shown to explain an observed miscomputation in terms of functional error states that can be multiply realized by different lower-level implementations. It is brought into question the thesis that architecture descriptions afford mechanical explanations of the system's failures. Descriptions of hardware architectures computing some given execution are acknowledged as mechanism sketches supplying functional explanations and focusing on state or combinatory elements introduced as functional units. Only the descriptions of working transistors implementing logic gates are capable of providing mechanical-causal explanations. However, it is argued here that those explanations concern the physical processes of an electric circuit, not the computational processes of a digital circuit.

*Speaker

[†]Corresponding author: