Computational Equivalence

by Sven Nilsen, 2019

In two previous papers I showed that there are *formal* and *informal* associations[1], and we are asserting formal associations in formal languages used for theorem proving[2].

Computational equivalence can be thought of as a process where the equivalence class of a symbol is expanded dramatically. The first thing one should notice is that the core axiom of path semantics[3] only implies the associated equality. It is not an equivalence itself. However, it does not prevent you from defining an equivalence. This means that if the associated equality is asserted to be an equivalence, the non-circular assumption vanishes:

$$F_0(X_0) \ \land \ F_1(X_1) \ \land \ F_0 = F_1 \qquad \simeq \qquad X_0 = X_1$$
$$X_0(F_0) \ \land \ X_1(F_1) \ \land \ X_0 = X_1 \qquad \simeq \qquad F_0 = F_1$$

$F_0 <> X_0$ non-circular assumption vanishes since it is false bi-directional

This means that computational equivalence might be asserted, but not formally. In this context, formal associations might hold for some symbol relations but not others, but in the larger picture it is not provable formally that this holds for all symbols.

For example:

$$120943 + 291283 = 412226$$
$$412226 = 120943 + 291283$$

You might associate something with the number \`412226\`, but you can not prove that the same association holds for \`120943 + 291283\`. This intuition can be explained by computation makes it hard to associate formally, such that we do not actually know whether two statements in a formal language are equivalent or not.

However, there is a loophole:

By normalizing[4] a term, which for arbitrary sub-types means evaluating it, association gets easier:

$412226 = 412226$
“120943 + 291283” > “412226” adding back non-circular assumption

Still, one can not prove the association formally. To prove that we mean the same about \`120943 + 291283\` as \`412226\` formally, one must add a reflection axiom in addition to normalization:

$X = X$: true

The intuition is that in a formal system with only one symbol, the reflection axiom is a tautology, but for systems with more than one symbol, one must perform some computation to check for equality.

References:

[1] “Lifted Associations”
Sven Nilsen, 2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/lifted-associations.pdf

[2] “Asserting Formal Associations”
Sven Nilsen, 2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/asserting-formal-associations.pdf

[3] “Path Semantics”
Sven Nilsen, 2016-2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/path-semantics.pdf

[4] “Normalization”
Wikipedia
https://en.wikipedia.org/wiki/Type_theory#Normalization