Inverse Theorems

by Sven Nilsen, 2020

In this paper I present three inverse theorems found in Path Semantical Logic.

Similar to Normal[1], Abstract[2] and Constrained[3] Implication Theorems, there are three Inverse Theorems, which are proofs in Path Semantical Logic[4]:

  (a, b, c) (B, C):   Normal Inverse Theorem
  ¬a, b(B), c(C) => B=C

  (a, b, c) (B, C):   Abstract Inverse Theorem
  ¬a, b(B)=c(C) => B=C

  (a, b, c) (B, C):   Constrained Inverse Theorem
  ¬a, b(B)=>c(C) => B=>C

Here, the tuple `(a, b, c)` has level 1 and the tuple `(B, C)` has level 0.
The notation `b(B)` means `b=>B` where `B` is at a lower level.

The Inverse Theorems are so unusual that I checked in Propositional Logic[5] that they are a consequence of the core axiom[6], just in case the implementation[7] of Path Semantical Logic was incorrect. What I found was that not all cases need to be covered by the core axiom.

  a, b, c, B, C: Normal Inverse Theorem in Propositional Logic
  (a=b, a=>B, b=>C => B=C),
  (a=c, a=>B, c=>C => B=C),
  ¬a, b=>B, c=>C => B=C

  a, b, c, B, C: Abstract Inverse Theorem in Propositional Logic
  (a=b, a=>B, b=>C => B=C),
  (a=c, a=>B, c=>C => B=C),
  ¬a, (b=>B)=(c=>C) => B=C

  a, b, c, B, C: Constrained Inverse Theorem in Propositional Logic
  (a=c, a=>B, c=>C => B=C),
  ¬a, (b=>B)=>(c=>C) => B=>C

In the Constrained Inverse Theorem, only a single core axiom is needed in Propositional Logic.

References:

[1]   "Implication Theorem"
        Sven Nilsen, 2020
        https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/implication-theorem.pdf

[2]   "Abstract Implication Theorem"
        Sven Nilsen, 2020
        https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/abstract-implication-theorem.pdf

[3]   "Constrained Implication Theorem"
        Sven Nilsen, 2020
        https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/constrained-implication-theorem.pdf

[4]   "Path Semantical Logic"
        AdvancedResearch, reading sequence on Path Semantics
        https://github.com/advancedresearch/path_semantics/blob/master/sequences.md#path-semantical-logic

[5]   "Propositional calculus"
        Wikipedia
        https://en.wikipedia.org/wiki/Propositional_calculus

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

[7]   "Faster Brute Force Proofs"
        Sven Nilsen, 2020
        https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/faster-brute-force-proofs.pdf