Notes for Self-Programming Workshop slides. August 01, 2011.

SLIDE 1. Title. Emergent Inference, or How can a program become a self-programming AGI system?

SLIDE 2. Overview of Emergent Inference. In 1850, german physicist and physician Hermann von Helmholtz was studying vision, and discovered Unconscious Inference. A new knowledge base: partially ordered sets. Explanation for emergence in complex dynamical systems. A new type of inference: Emergent Inference. Indications that EI is the principle that makes the brain work.

SLIDE 3. Brain experiments. I start from a body of knowledge describing a system. For example, a problem statement given to an OO analyst. The analyst creates an OO design. The brain is viewed as a black box. Only the input/output are relevant.

SLIDE 4. The first experiment. The motivating experiment. All greeks are given input. Permutations, memory. Flux lines represent flow of information. Shrink to reduce transmission energy and biological resources. The “program” is actually a stream of experience received by the canonical system. Presumably, the same functional that controls the matrix also controls my high cognitive function.

SLIDE 5. The result from the first experiment. Discuss structures, self-programming, classes, objects. This is emergent inference. Problem of Physics solved, but I didn’t write a single line of code. The “refactored program” is a self-programmed program. This is the mathematical foundation for everything that follows. Cannot be ignored.

SLIDE 6. The importance of this discovery. The discovery has a vast importance and scope.

SLIDE 7. Claim and conjectures. The claim is a mathematical fact, not for discussion. Conjectures cannot be proved. However, it would not be correct to assume without proof that a structure observed in a complex system ***is not*** the natural structure of the system.

SLIDE 8. Representing systems as partially ordered sets. It is easy to represent systems as partially ordered sets. Much easier than actually solving the equations.

SLIDE 9. Traditional software development cycle. “Brain” means the human analyst providing the intelligence. When development is complete, the analyst is removed. All elements in red go away. Only the structures remain, but they are zombies, autistic: cannot be updated, sterotyped behavior, no intelligence, no emotions, no creativity. Structures that update other structures give the illusion of updating. All “thinking” happens in EI. There is no thinking in “Program”. When the analyst is gone, so is the thinking. The structures only give the illusion of thinking. When the analyst goes, intelligence goes. When the analyst is removed, the stream of experience also changes. It is now called “data.” This slide demonstrates that self-programming requires AGI.

SLIDE 10. Software Categorization. The analyst provides the EI. A structure is frozen when it can not be updated without human intervention. Frozen structures give the illusion of adaptation, but they follow rigid, invariable rules. Propose a progressive approach to AGI: start with hybrid programs, then expand the reach of EI. Blurred boundaries, hard boundaries.

Ben Goertzel: “... cognitive code that self-modifies in such a way as to affect the actual results it produces for given inputs.”

J. Storrs Hall: “The processes by which humans extend their own programming differ significantly ... from the ways in which ... a human would program a robot.”

Stefan Leijnen views a program as a constraint: “What a program produces is already implicitly present in its initial set of instructions.” “A program is in a maximally constrained state.” “A self-programming system must be capable of producing new constraints.” “Have we encountered a paradox?” YES!!!

David Ferrucci (Watson): “The way we program computers nowadays cannot create human intelligence.”

SLIDE 11. Traditional AI and AGI. A stage control program will not drive your car, and you can not play chess with a car driving program. This slide demonstrates the limitations in input range and output range of all programs. It also demostrates why integration or self-programming will not happen automatically. There is no refactoring, because all programs involved have been manually refactored by an analyst.

SLIDE 12. The brain. The brain is naturally integrated and does its own refactoring naturally.

SLIDE 13. Emergent Inference. This is fully feathered AGI. There are no limitations, no constraints. An I am doing it every day on my PC. EI is the only type of inference not man-made. Man-made inference always contains constraints imposed by humans, in the form of parameters, or procedures. The program cannot do anything outside these man-made parameters or procedures.

SLIDE 14. We need a principle of intelligence. George Cayley identified the lift force ~1800, thus establishing the ***principle*** for heavier-than-air flight and founded Aeronautical Engineering. Here is a brief history of failures at understanding automation.

SLIDE 15. Conclusions.