

## Selmer Bringsjord, **Logic, AI, and Pure Logic Programming: The Disastrous Disunion**

How We Got Both Today's Unintelligent, Black-Box AI, and Inelegant, Unverifiable Programming Languages

Relevant history being what is, humanity missed its first great chance to maintain the triadic union of logic, AI, and logic programming; and to, standing firmly upon that triad, reach great heights that are in fact now at best pipe dreams. In 1956, the pinnacle of AI at its very dawn was Logic Theorist (LT), an automated theorem prover created by Simon and Newell; soon thereafter the best AI on the planet was their even-better artificial reasoner: General Problem Solver (GPS) of 1959. LT and GPS exemplified the triad. But today, while cutting-edge automated reasoning is in fact impressive and soldiers on, the triad is broken apart and its pieces have been cast aside as remnants of a carcass; and AI is identified not with stunning automated reasoning, but rather with silly and humble black-box systems that for example recognize objects in pictures courtesy of massive amounts of data provided beforehand, and high performance in easy games (like Go) that is based on little more than what a calculator does. Hence all we have today is — and yes, this is an oxymoron if ever there was one — unintelligent AI. On the programming front, the history is a parallel one, and just as disastrous. Programming had a chance to embrace and build upon the triad of logic, AI, and computation, in order to program computers elegantly, powerfully, and transparently. Cordell Green pointed to how to use the triad in 1969, and while Prolog and its variants are somewhat related and have survived, they are both decidedly impure (contaminating declarative content with procedure), and, by any metric, if not carcasses, then certainly marginalized in favor of inscrutable code that perhaps even God can't verify.