AI-DSL for autonomous interoperability

Nil Geisweiller

SingularityNET & OpenCog Foundations

1. What?
   - Provide formal description of an AI
2. Why?
   - Inform why to use a certain AI
   - Enable interoperability between AIs
3. How?
   - Dependent Types (Idris, AGDA, Coq, Liquid Haskell)
   - Probabilistic Logic Networks (PLN), OpenCog Hyperion
   - Natural Language Processing (NLP)!? (SingularityNET!?)
   - . . .

* Brainscape:
- OpenCog:
- MOSES:
- infer type signature of the candidates to evolve given the fitness function, and define program spaces fulfilling such type signature.
- Evolve candidates using search guided by reasoning, with a formalized local search and fitness function (SNET presentation)
- Learning via Reasoning: OpenCog Pattern Miner, mining patterns fulfilling a specification.
- Planning via Reasoning: discover plans (to control agent in env) provably probabilistically correct.
- Magic Haskell.
- Gluing mismatched programs. Need bridgers: bridgers need formal specification so they can be part of the network.
- Verified Stack-Based Genetic Programming via Dependent Types.
- NLP: because often the only available specification is the documentation.