Symbolic Computation: Course Outline

David McAllester

December 29, 2009

• Inference as a model of computation
  – Inference rules and inference relations. Transitive closure of a graph.
  – Turing universality of inference rules.

• Foundations of Mathematics
  – Boolean expressions. Syntax, semantics, inference rules, soundness and completeness.
  – Type variables, pairing, and recursive types. An implementation of the natural numbers.
  – Existential types as an abstraction barrier. The natural numbers with the implementation hidden. The Integers.
  – Functions. Equality on functions. The ordered field of real numbers (with implementation hidden). The field of complex numbers.
  – Vector Spaces. The dual of a vector space.
  – Automorphisms: The symmetry group of an abstract object.
  – Parametricity. The non-existence of a natural isomorphism between a vector space and its dual.
  – Type isomorphism: the many ways to define a group.

• A survey of existential types.
  – Groups and permutation groups. Every group is isomorphic to a permutation group.
  – Boolean Algebras and Fields of sets. Every Boolean algebra is isomorphic to a field of sets.
– Hilbert spaces. Every separable Hilbert space is isomorphic to \( \ell_2 \).
– Inner product spaces. Special relativity. Physical Units.
– The type theory of PCA and CCA.
– Manifolds.
– Measure Theory.

• Symbolic Algebra
  – Polynomial Arithmetic
  – Groebner Bases
  – Automatic Differentiation
  – Symbolic Integration

• Type Theory and Natural Language
  – Montague Grammar and Montagovian Logic
  – Modalities of Natural Language (possibility, likelihood, knowledge, ability, and permission)
  – Tense and Aspect