Lecture 2

Pete Manolios
Northeastern

Fundamental Questions

- How do we describe computational systems?
- How do we reason about computational systems?
- Well?

Fundamental Questions

The development of mathematics towards greater precision has led, as is well known, to the formalization of large tracts of it, so that one can prove any theorem using nothing but a few mechanical rules.

Kurt Gödel upon presenting his incompleteness theorems in 1931.

Either mathematics is too big for the human mind or the human mind is more than a machine.

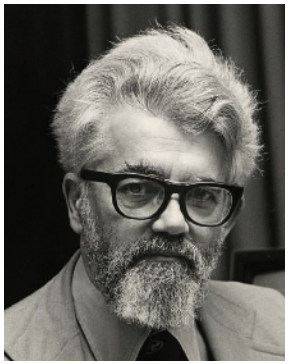
Kurt Gödel (an interesting project, Ankit?)

Mechanized Reasoning

Instead of debugging a program, one should prove that it meets its specifications, and this proof should be checked by a computer program.

"A Basis for a Mathematical Theory of Computation"
1961

John McCarthy

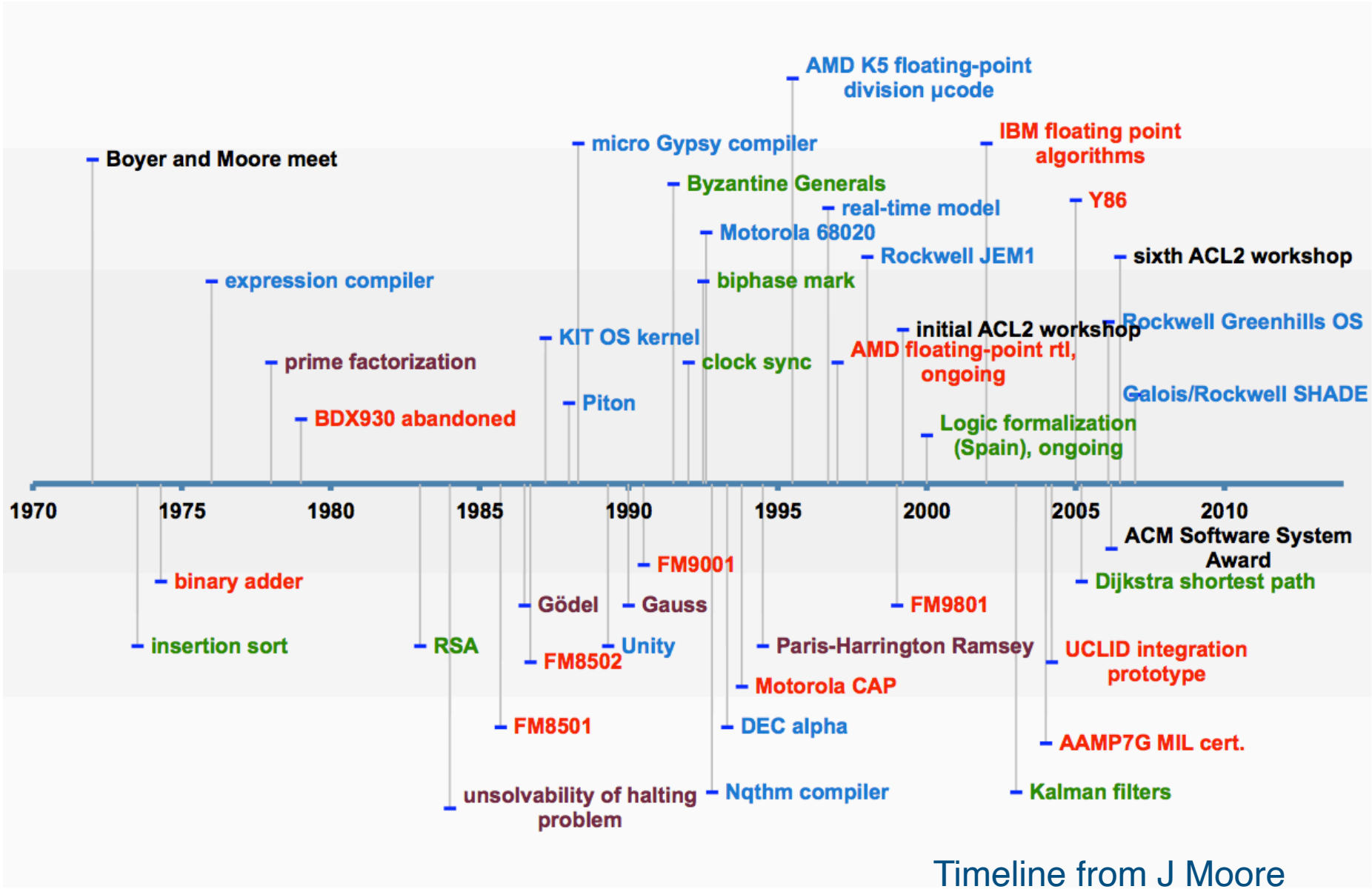


Boyer-Moore Theorem Provers

- 1970's
  - Edingurgh Pure Lisp Theorem Prover (1973)
  - A Computational Logic (1978)
- 1980's
  - NQTHM (1981)
  - ACL2 (1989) A Computational Logic for Applicative Common Lisp
- 1990's-Present
  - Kaufmman joins as developer
  - Workshops (16 already); huge regression suite
- 2000's:
  - ACL2 books
  - Development environments (ACL2 Sedan)
  - 2005 ACM Software System Award (Boyer, Kaufmann, Moore)

Boyer-Moore Theorems Proved

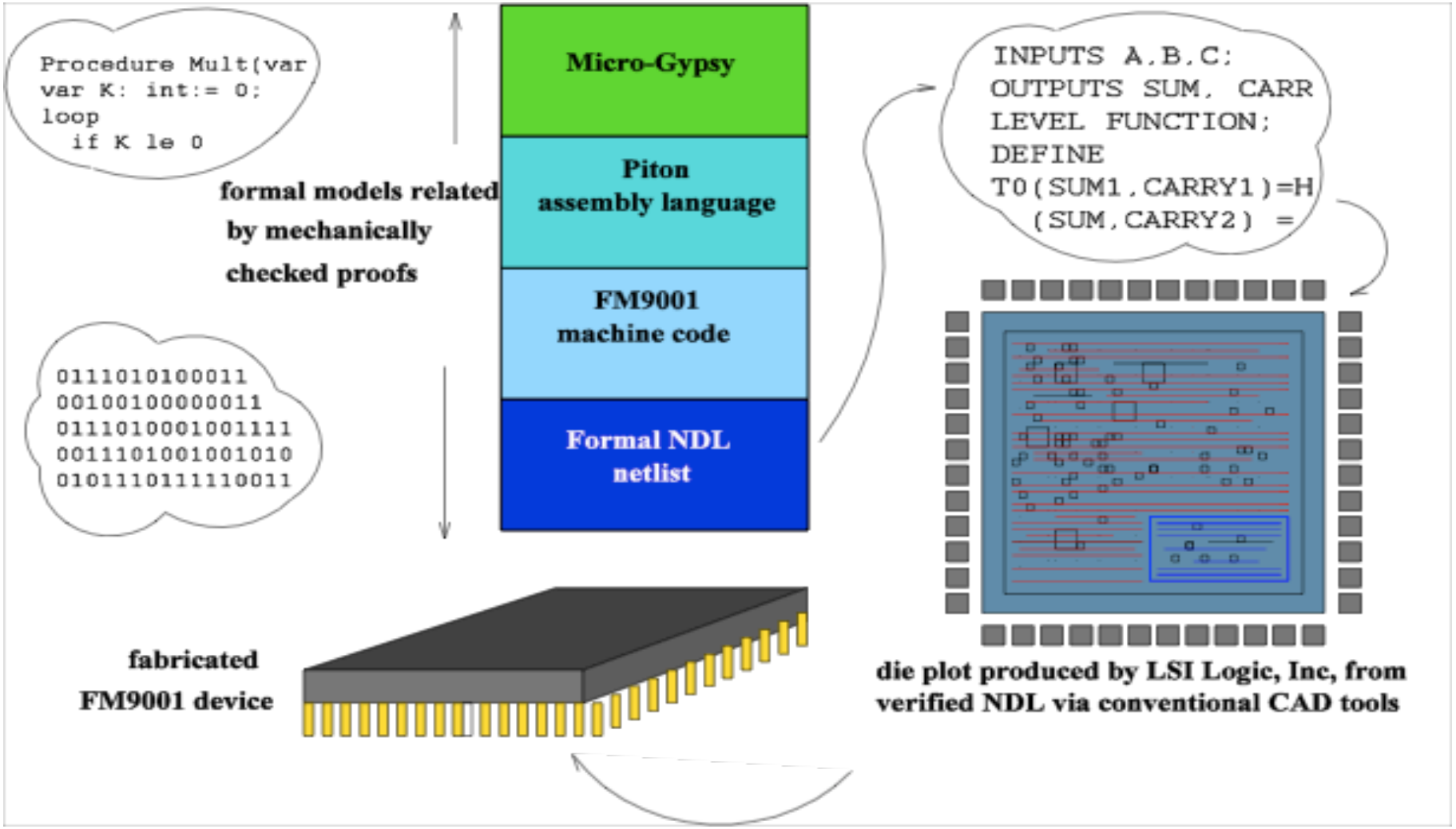
- 1970's: Simple List Processing
  - Associativity of append
  - Prime factorizations are unique
- 1980's: Academic Math & CS
  - Invertibility of RSA
  - Undecidability of halting problem
  - Gödel's First Incompleteness Theorem
  - Gauss' Law of Quadratic Reciprocity
  - CLI Stack:
    - Microprocessor
    - Assembler-linker-loader, Compiler, OS
    - High-level language



Timeline from J Moore

Slides by Pete Manolios for CS4820

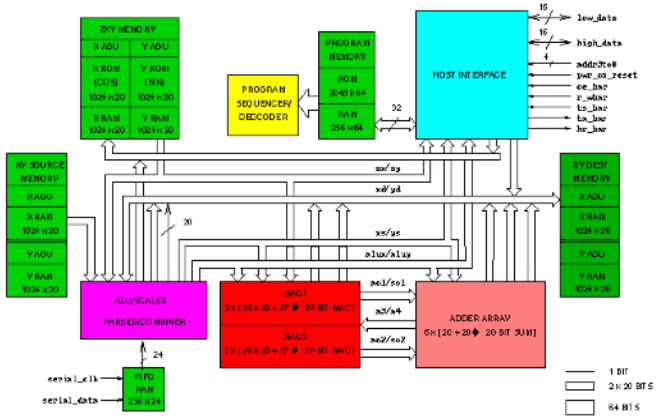
CLI Stack

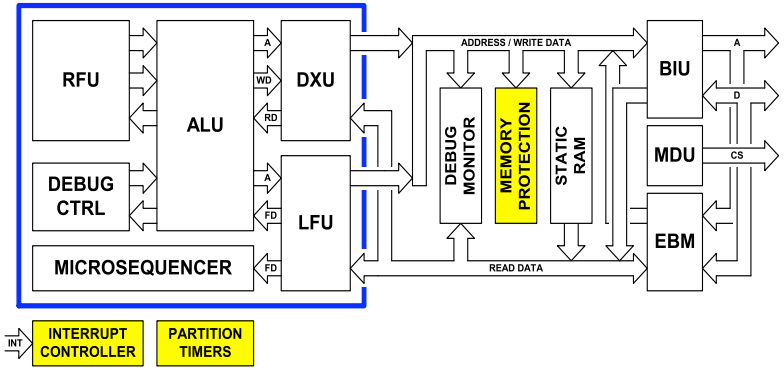


Slides by Pete Manolios for CS4820

Industrial Applications

- FDIV AMD Floating Point, IBM ...
- Motorola CAP DSP
  - Bit/cycle-accurate model
  - Run fasters than SPW model
  - Proved correctness of pipeline hazard detection in microcode
  - Verified microcode programs
- Rockwell Collins JEM1
- Rockwell Collins AAMP7
  - MILS EAL-7 certification from NSA for their crypto processor
  - Verified separation kernel
- Centaur: Media Unit





Slides by Pete Manolios for CS4820

So What?

Mechanized reasoning for commercial systems ✔

- Scalability to industrial problems
- Tool maturity
- Human talent
- Repeatability
- Time to market
- ROI vs other methods

What's Next?

Computer-aided reasoning for the masses

Teach freshmen how to reason about programs

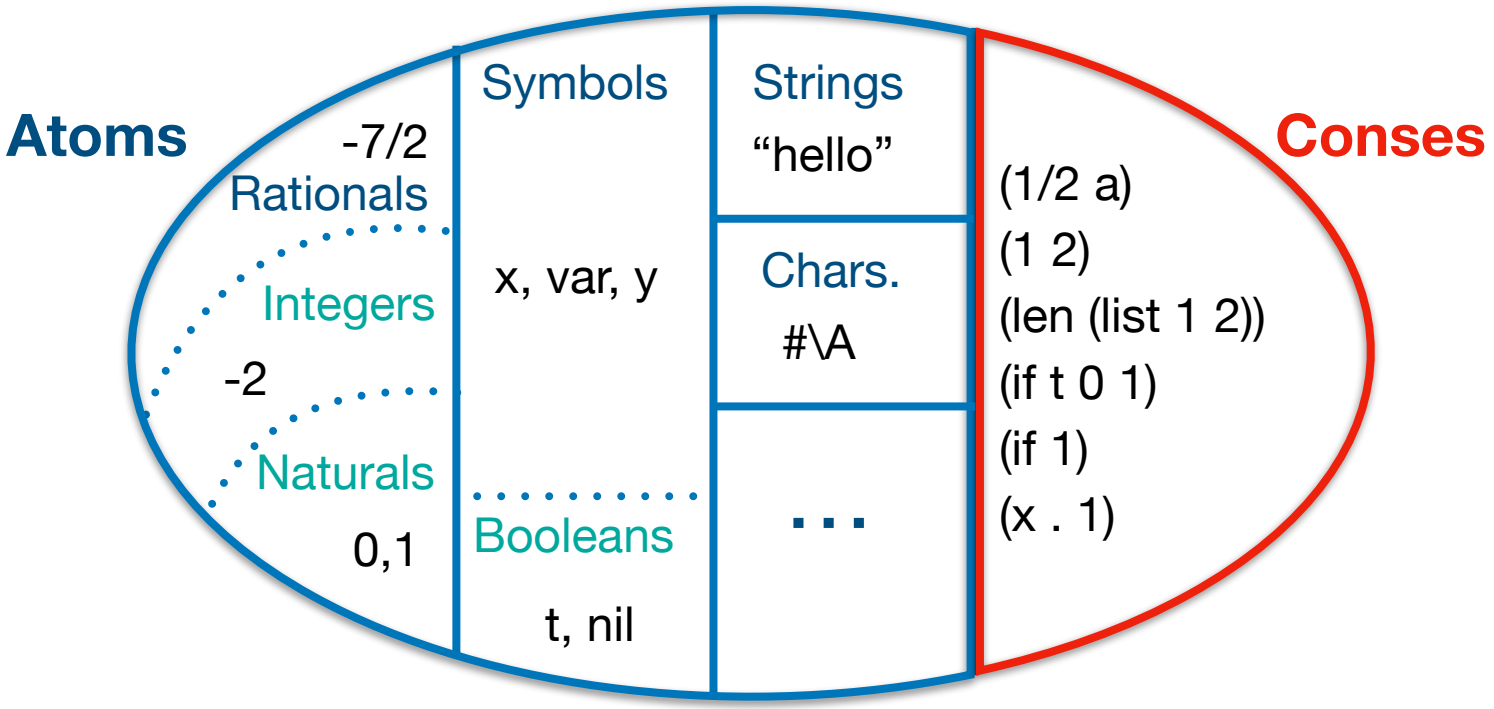
Slides by Pete Manolios for CS4820

ACL2 is ...

- A programming language:
  - Applicative, functional subset of Lisp
  - Compilable and executable
  - Untyped, first-order
- A mathematical logic:
  - First-order predicate calculus
  - With equality, induction, recursive definitions
  - Ordinals up to ε0 (termination & induction)

ACL2 Universe

All = Conses ∪ Atoms



Lists = Conses ∪ { ( ) }

True-lists = $\cup_{i \in \mathbb{N}} L_i$

$L_0 = \{ ( ) \}$, $L_{i+1} = L_i \cup \{(\texttt{cons x l}) : \texttt{x} \in \texttt{All}, \texttt{l} \in L_i\}$

Slides by Pete Manolios for CS4820

Data Definitions

- Allow us to write recognizers & enumerators for subsets of the universe
- Singleton types
- Recognizers
- Enumerated Types
- Range Types
- Product Types
- Records
- Constructors & Accessors
- Listof Combinator
- Union Types
- Recursive Types
- Mutually Recursive Data Types
- Custom types

DEMO

Questions?

- HWK 1 due Sunday
- Next week:
  - Definitions
  - PL
  - Equational Reasoning
  - Read material before class