Panel on (C)LP teaching at ICLP 03

Manuel Hermenegildo

- Involved in (C)LP since 1983.
- Taught graduate and undergraduate courses since 1987
  (LP, CLP, formal logic, + some tutorials and Summer schools).
- (Co-)authored a relatively complete, freely set of slides on many (C)LP topics.
  `http://cliplab.org/~logalg`
- Currently at both:
  - Tech. U. of Madrid, Spain.
  - U. of New Mexico, US (and previously at U. of Texas at Austin, US).

  (So report on experiences both in Europe and US.)

Pedagogical objectives of (C)LP courses

- Learn two of the major programming paradigms (LP/CLP).
- Learn declarativeness, elegance, precision in programming.
- Recall and apply fundamental results in logic, automated reasoning.
- Enhance mathematical reasoning and problem-solving abilities.
- Show the usefulness of logic as a basis for other areas of CS

- *Important: communicate that these are useful, real-world languages.*
- Learn about typical application techniques / areas / possibilities.

(C)LP courses in area (Spain)

- Logic, LP, CLP, etc. at UPM in Madrid, Spain
  - Current situation:
    - 5 year master's level program.
    - 1st year: 1 semester of formal logic (400 students).
    - 2nd year: 1 semester of computational logic (400).
    - 3rd year: 1 sem. programming paradigms (400) + 1 sem. LP (elective, 200).
    - 4th year: 1 sem. advanced LP and CLP (elective, 100).
    - 5th year, several related electives: Fuzzy Logic, NLP, Verification, etc. (˜50).

    The LP/CLP electives are *very* popular!
  - Immediate future (2004/2005 Bologna process):
    - Moving to 4 year bachelor + Master/PhD (essentially US model).
    - Logic and elementary LP in bachelor (probably in 'Paradigms' course).
    - Advanced LP, CLP, Fuzzy Logic, NLP, Verification, etc. in Grad. program.

(C)LP courses in area (US)

- Logic, LP, CLP, etc. at UNM in Albuquerque, NM, US.
  - Bachelor:
    - CS261 Mathematical Found. of CS (*some* logic, proof, ...), CS401 (TCS).
    - *Required* course (CS451) Programming Paradigms:
      - Elementary logic programming.
      - Some use of logic programming (e.g., for interpreting SOS rules).
    - Use of Prolog in AI course (CS427).
  - Graduate program:
    - *Required* course (CS550) in programming language fundamentals (approx. 35 students):
      - Logic programming.
      - And now also advanced LP and CLP (I teach it!)
    - Electives in automated reasoning (CS537), verification (CS553/CS580), etc.
    - Some (voluntary) use in advanced AI (CS527/CS528)

Topics in existing courses

- *Formal logic course* (formal systems, 1st order pred. logic, completeness, decidability) + *Computational logic course* (natural deduction, resolution, knowledge representation.)
- LP course:
  - Greene's dream. Pure LP (using breadth-first search).
  - The Prolog Language. Programming in Prolog. DCGs. Incomplete data structures. etc. Efficient Prolog programming. Applications.
  - Advanced LP: syntactic/semantic extensions, higher order, relations+functions, records, object orientation...
  - Brief review of FO pred. logic, resolution. Implications of fundamental results.
  - LP semantics (model, fixpt). Equiv. of decl. and op. semantics. Program Equiv.
- CLP course:
  - Constraint Programming. CLP(R). CLP(Q). Finite domains.
  - Constraint Programming theory. Fundamental results.
  - Advanced Topics (e.g., implementation, static debugging, verification).

Personal view of an ideal curriculum

- I think the 5 year Spanish program is very good:
  - 1st year: 1 semester of formal logic.
  - 2nd year: 1 semester of computational logic.
  - 3rd year: 1 sem. programming paradigms + 1 sem. LP.
  - 4th year: 1 sem. advanced LP and CLP.
  - 5th year, several related electives: Fuzzy Logic, NLP,

  ...but it is a dying species.

Personal view of an ideal curriculum

- A proposal for a 4+2 model:
  - 1st year or 2nd year: 1 semester of *computational* logic.
    - Needed later for specification, verification, knowledge representation, etc.
  - 2nd and 3rd year: individual courses, 1 in FP and 1 in (C)LP.
  - 3rd and 4th year: electives in advanced LP and CLP. Fuzzy Logic, NLP, theorem proving, verification, etc.
- Other points of presence:
  - AI course.
  - Compilers (abstract machines, LP compilation).

Personal remarks

- Communicate that these are useful, real-world languages.
  - Teach first programming, on the computer (vs. teaching formal logic first?) (Otherwise they lose interest and get the wrong message!)
  - Do a project with, e.g., a WWW interface: gives 'real-world' feel.
  - Teach typical application techniques / areas / possibilities.
- Start with pure logic programming, but a system with *completeness* needed to avoid early confusion:
  - Used to do it with a breadth-first meta-interpreter.
  - Now use Ciao because of breadth-first and iterative deepening execution rules.
- Justify transition to Prolog (efficiency in memory, time); clarify difference between when programming and when doing theorem proving (programming=reasoning about/controlling cost).
- Use functions, higher-order: to show that the paradigm is not "lacking" anything.
- Say that these are "powerful, difficult tools for smart people to solve difficult problems" (true, and also works ego).

Personal remarks (Contd.)

- Other issues:
  - (C)LP elective courses *very* popular.
  - Paradigms course generally *counter-productive:* you need more time to 'get it', instructors typically not experts, eclipsed by FP and OO (normally have own course + part in paradigms course).
  - Preponderance of FP (specially in US): typically at least one course in addition to paradigms course.
  - There should be space for one LP course.
  - Need to get on ACM curriculum!!!

For more info:

`http://cliplab.org/~logalg`