DM820
Advanced Topics in Programming Languages

Peter Schneider-Kamp

petersk@imada.sdu.dk

http://imada.sdu.dk/~petersk/DM820/

MULTI-PARADIGM & CONSTRAINT PROGRAMMING

Programming Paradigms

- Actor programming
- Concurrent programming
- Constraint programming
- Dataflow programming
- Declarative programming
- Distributed programming
- Functional programming
- Generic programming
- Imperative programming
- Logic Programming
- Metaprogramming
- Object-oriented programming
- Rule-based programming
- Visual programming

Multi-Paradigm Languages

- Many languages use more than one paradigm:
  - Java: imperative, object-oriented, reflective, generic
  - Python: imperative, object-oriented, reflective, functional
  - C#: imperative, object-oriented, functional, reflective, generic
- These combinations are quite straightforward
- Multi-paradigm languages combine less obviously combinable programming paradigms:
  - Curry: constraint, functional, logic, concurrent
  - Oz: imperative, object-oriented, functional, logic, constraint, distributed, concurrent
- How to combine functional and logic programming?
- How to integrate constraint programming?

Declarative Programming

- Imperative programming:
  - commands for HOW to achieve a goal

- Declarative programming:
  - Express WHAT exactly should be achieved

  - Functional programming (lambda calculus)
  - Logic programming (predicate calculus)
  - Constraint programming (constraint satisfaction)

Constraint Programming

- Just provide constraints on a solution
- Let the computer figure out how to find solutions
- Example (constraint logic programming):

```
puzzle([S,E,N,D] + [M,O,R,E] = [M,O,N,E,Y]) :-
        Vars = [S,E,N,D,M,O,R,Y],
        Vars ins 0..9,
        all_different(Vars),
        S*1000 + E*100 + N*10 + D +
        M*1000 + O*100 + R*10 + E #=
        M*10000 + O*1000 + N*100 + E*10 + Y,
        M #\= 0, S #\= 0,
        label(Vars).
```

Example: Curry

- Belongs to the class of constraint functional logic languages
- Syntax very close to Haskell
- Named after mathematician Haskell B. Curry
- Implementation developed in Portland, Aachen, Kiel (PAKCS)
- Applications:
  - Bibliographic database, Wine manager, Recipe database
  - Music composition
  - Web-based learning, Web server scripting
  - Ecological simulation
  - Course assignment, Study program management
  - Graph grammar parsers
  - …

Hands-On

- Functional programming
- Logic programming
- Mixed functional-logic programming
- Send + more = money