Gecode

Christian Schulte[1]
Guido Tack[2]
Mikael Z. Lagerkvist[1]

[1] KTH – Royal Institute of Technology, Sweden
[2] Saarland University, Germany

Gecode

- Generic constraint development environment
- Open source C++ library
- Key aspects
  - open: programming interfaces
  - free: BSD-style license
  - portable: whatever hardware/software environment
  - accessible: extensively documented
  - efficient: competetive performance (space/time)

Open Platform for...

- Research
  - extensibility
  - openness
- Education
  - modern free platform for teaching CP
- Deployment
  - do whatever you want
- Efficiency
  - be relevant

Gecode Architecture

- Generic kernel
  - kernel core
  - generic abstractions (branching, propagators, ...)
- Modules
  - one per variable domain (as many as you want)
  - search engines
  - modeling support
  - ...

Kernel

- Small and generic
  - coordinates constraint propagation and search
  - advanced propagation organization [CP 2004]
  - no special pets
- Provides interfaces for
  - variable domains
  - generic propagators [CP 2005, Recent Advances in Constraints 2006]
  - generic branchings (labelings)
  - search engines
- Around 3000 lines of code
  - core just 1400 lines

Search

- Search based on recomputation
  - expressive for programming search
  - adaptive and batch recomputation for efficiency
- Standard engines
  - depth-first search
  - limited discrepancy search
  - branch-and-bound optimization
  - DFS restart optimization
- Parallel search to be available soon
  - portability and licensing issues keep us

FD Integers

- Use generic kernel interfaces
- Standard constraints
  - arithmetic, Boolean, and linear constraints
  - reified versions of the above
- Global constraints
  - all-different, global cardinality, count, element, regular, lexicographic ordering, inverse, sortedness, cumulatives
  - typically supporting various consistency levels

FD Sets

- Bounds and cardinality reasoning
  - complete BDD-based solver underway
- Standard constraints
  - set relations and operations
- Global constraints
  - convexity, distinctness, atmost, selection, FD integer connection
- Compiler for generating propagators from formulas [CP 2006]

Modelling

- Modelling not primary goal…
  - designed for being interfaced to
  - …still, we want to write nice examples ☺
- Natural representation of expressions

```
post(this, x + 3*y >= z);
```

- Matrices of variables

```
for (int i = 0; i < 9; ++i)
  distinct(this, sudoku.row(i)),
  distinct(this, sudoku.col(i));
```

Efficiency

- Efficient in terms of
  - time
  - memory
- Efficiency consequence of
  - simplicity
  - informed design decisions
  - choosing the right algorithm/data structure
  - *not* from hacking
- See benchmarks at Gecode website
  - Gecode 1.3.1 consistently faster than ILOG Solver 6.0
  - typically also less memory

Quality: Systematic Testing

- Extensive test-suite for all constraints in the system
- Indispensable
  - for users
  - for reproducible research
- Randomized tests with relatively good coverage
- We found *many* bugs using this
- Major bugs found by users afterwards
  - one since December 2005

Interfaces

- Gecode/J
  - Java interface used for education
- AliceML
  - dialect of Standard ML
  - Gecode-bindings as a standard library, used for education
- GeOz
  - project to integrate Gecode into the Mozart/Oz environment (AVISPA Group)
- Possibly (planned): Python, ECLiPSe, G12

Gecode/J

- Complete interface in Java
  - modeling, propagators, branchings, etc
  - provides barrier-free and complete approach
- Used in education
  - KTH, Sweden
  - Uppsala U, Sweden
  - UCL, Belgium
  - American U, Egypt (planned)
  - Saarland U, Germany (planned)
  - U Freiburg, Germany (planned)

Contributions

- Most important contribution: fully open design
  - particular: program new variable domains
- Other contributions
  - organization of propagation
    [Schulte, Stuckey, CP 2004] [Schulte, Stuckey, CoRR, submitted, 2006]
  - views and iterators for generic propagators
    [Schulte, Tack, CP 2005]
  - automatic generation of set propagators
    [Tack, Schulte, Smolka, CP 2006]
  - search based on recomputation
  - systematic tests

Use Cases

- For users with background in CP
- Integrate CP technology
  - companies (small): cheap access
- Extend CP
  - QeCode: quantified constraints [Lallouet ea, CSCP 2006]
  - new variable domains CP(Graph), CP(Map): [Dooms ea, CP 2005] [Zampelli ea, CP 2005]
- Realistic experimentation platform
  - randomization in tail assignment [Otten ea, CP 2006]
  - abstractions for non-deterministic search [Michel ea, CP 2006]
- So far, dominated by academia

Relation: ILOG Solver

- Close relative
  - library in C++
- Key differences: ILOG Solver ...
  - provides sophisticated solver agnostic modelling abstractions
  - implements many more constraints and methods of search
  - has extensive tutorial documentation and debugging tools
  - offers professional support
  - is closed, few implementation techniques published

Relations: Choco

- Also allows to program new variables
- Open source
- Java
- Choco queues events, Gecode queues propagators
- Choco has explanation-based solving

Near Future

- More constraints, please contribute
- Tutorial to complement reference documentation
- Powerful abstractions for incremental propagation techniques
- Parallel search abstractions for shared memory machines
- BDD-based finite set solver
- New techniques for combination schemes not compromising propagation
- …

System Information

- Currently developed by
  - Christian Schulte (head, KTH – Royal Institute of Technology, Sweden)
  - Guido Tack (Saarland University, Germany)
  - Mikael Z. Lagerkvist (KTH – Royal Institute of Technology, Sweden)
- Open source, BSD-style license
- Version 1.3.1 released on October 25th
  - 50 kloc, 24 klod
  - some 100 serious users (maybe many more, as included in Debian, Ubuntu, etc)
- Available from http://www.gecode.org