The Application of Deductive Synthesis to the Rapid Assembly and Re-Assembly of Grid Applications

Alan Bundy, Bin Yang & Alan Smaill

School of Informatics
University of Edinburgh
EPSRC Research Project

- Grant Holders: Alan Bundy, Alan Smaill.
- Project Student: Bin Yang.
- Start Date: 1st Jan 2003.
- Duration: 3 Years.
- Part of UK e-Science Programme.
e-Science and the Grid

- Data-intensive sciences: particle physics, genomics, Earth satellite monitoring.
- Distributed, high-performance computing; high-bandwidth communications.
- Need for rapid assembly and re-assembly of Grid applications.
Typical Grid Application

Data → Processing
Data → Processing
Data

Processing → Visualisation
Data Transformation

Process 1  Data transformation  Process 2
Deductive Synthesis

Specification: \( \forall inputs \, \exists output \, \text{spec} (inputs \, , \, output) \)

Constructive Deduction

Proof

Extraction

Program: \( f \, \forall inputs \, \text{spec} (inputs \, , \, f (inputs)) \)
Application to Grid Assembly

- Specify each Grid service.
- Specify Grid application required.
- Prove specification in constructive logic.
- Extract required Grid application from constructive proof.
- Possibility of user interaction.
Application to Grid Re-Assembly

- Grid application breaks in service...
- ...because key Grid service fails.
- Reprove (possibly modified) specification in real time from available services.
- Possibility to use analogy from original proof for total automation.
- Extract revised application.
Simple Example

Process 1

\[ \forall input . \text{spec}_1 (input , f_1 (input )) \]

Process 2

\[ \forall input . \text{spec}_2 (input , f_2 (input )) \]

User specification:

\[ \forall input . \exists x . \exists output . \text{spec}_1 (input , x) \land \text{spec}_2 (x , output ) \]

Extracted program:

\[ \lambda x . f_2 (f_1 (x)) \]
Conclusion

- Deductive synthesis uses constructive proof to assemble complex objects.
- e-Science requires rapid assembly and re-assembly of Grid applications from Grid services.
- Deductive synthesis can be applied to these tasks.
- Need to explore appropriate formalisms for specification.
- Need to explore automation of proofs.