cisa | Centre for Intelligent Systems and their Applications

THE UNIVERSITY of EDINBURGH informatics

A strategy language to facilitate proof re-use

Gudmund Grov
School of Informatics
University of Edinburgh

with contributions from: Alan Bundy, Lucas Dixon, Katya Komandantskaya & other AI4FM project partners

AI4FM

Motivation

- From expert provided proof(s)
- ... extract high-level proof strategy
- ... to be re-used for “similar” conjectures
- ... in order to increase proof automation

Proof illustration

[g1]

apply a; [g2]

apply b; [g3,g4]

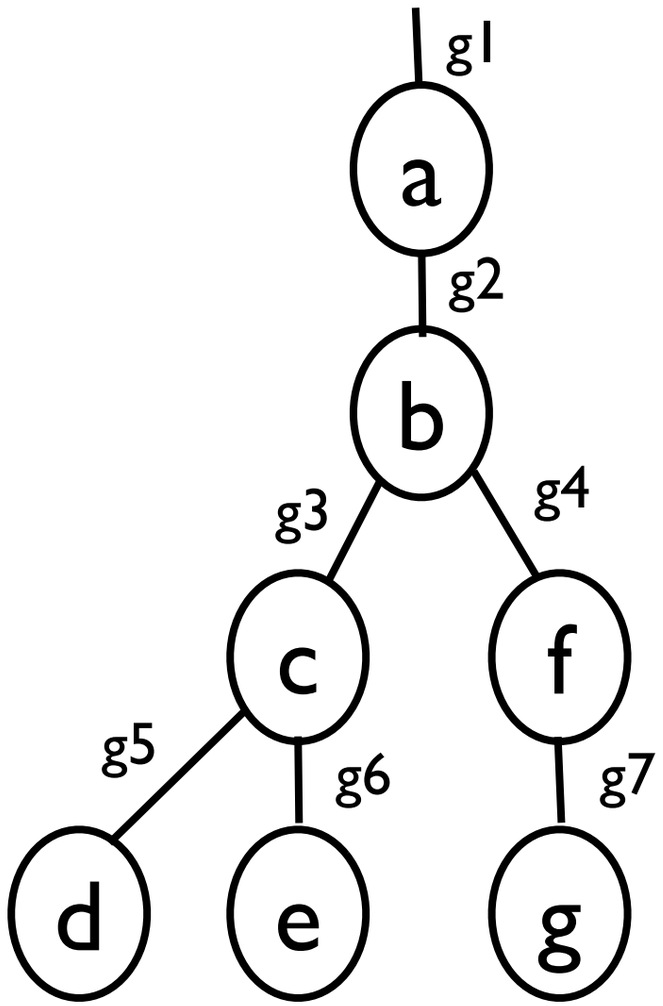
apply c; [g5,g6,g4]

apply d; [g6,g4]

apply e; [g4]

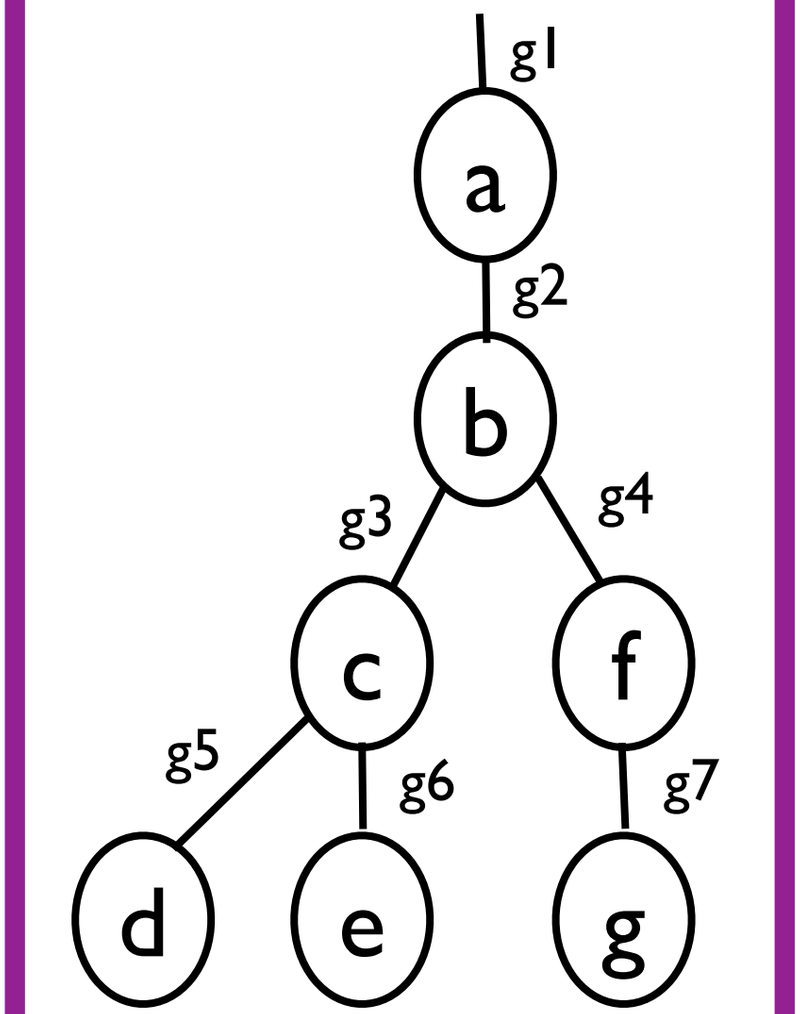
apply f; [g7]

apply g; []



Proof dimensions

tactic level	goal level
	[g1]
apply a;	[g2]
apply b;	[g3,g4]
apply c;	[g5,g6,g4]
apply d;	[g6,g4]
apply e;	[g4]
apply f;	[g7]
apply g;	[]



proof level

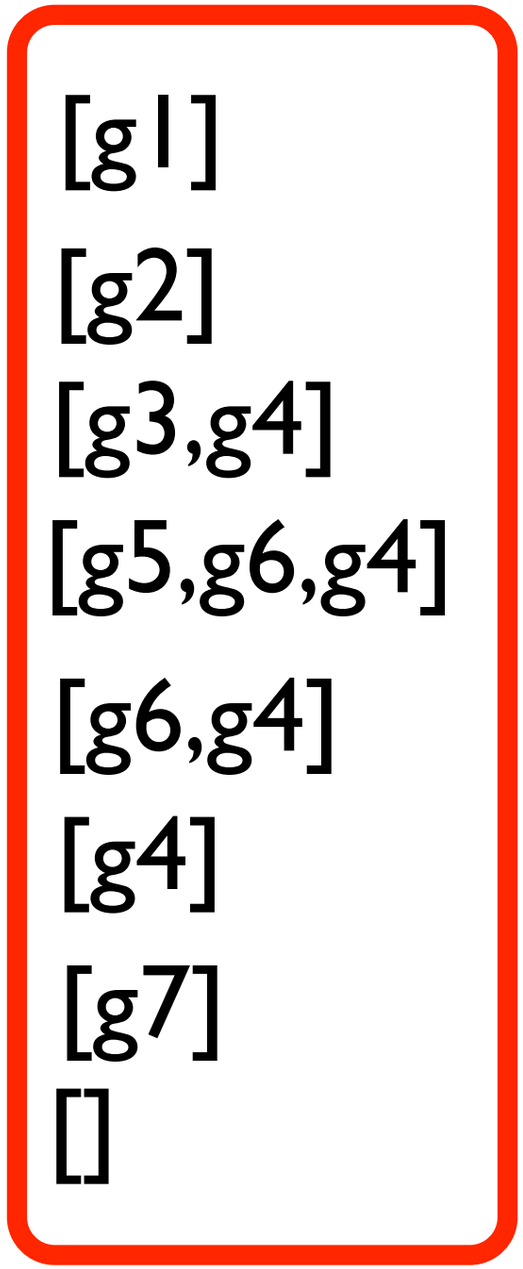
Proof dimensions

```
apply a;
apply b;
apply c;
apply d;
apply e;
apply f;
apply g;
```

tactic level

- combination of tactics only
- [+ tree structure?]
  - no information on goals
    - tactic applied to
    - result from applying tactic

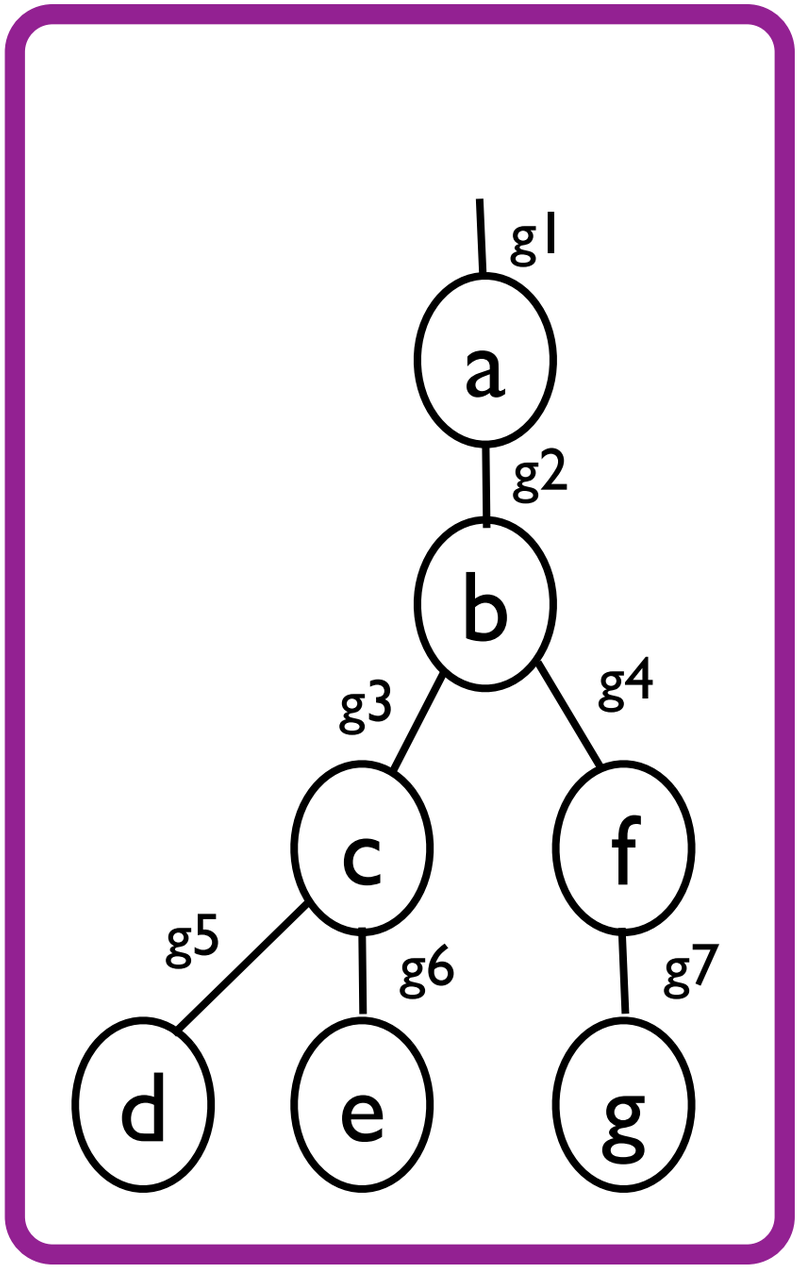
Proof dimensions



- goal (proof state) level
- look at goals only
- [+ tree structure?]

Proof dimensions

- goals + tactic + tree structure
- proof term sub-level
  - the full details of proofs
  - ... but loses tactic structure

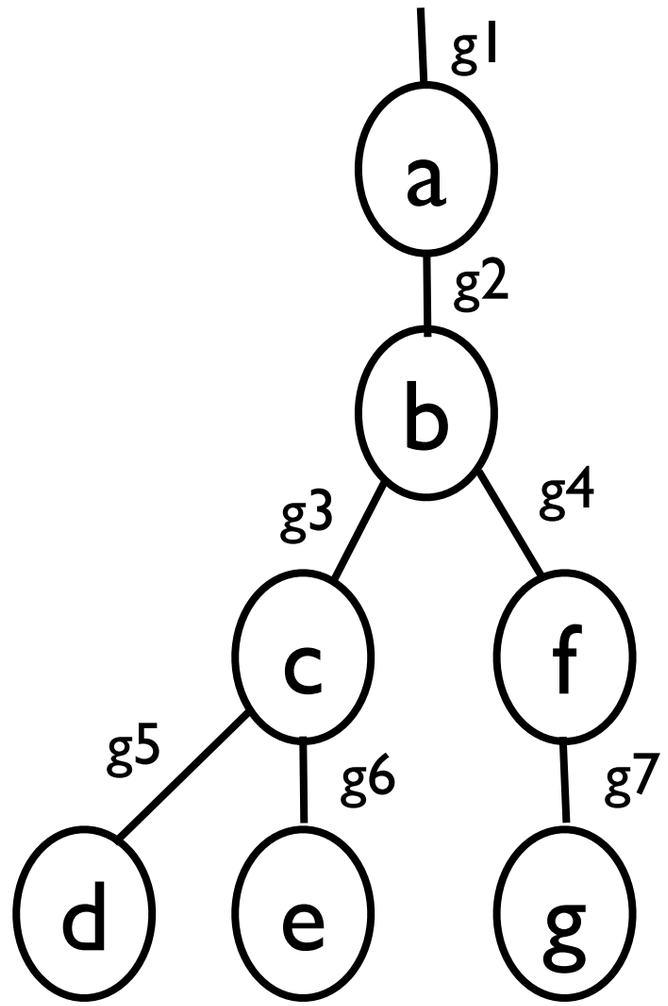


proof level

Proof dimensions

apply a;
apply b;
apply c;
apply d;
apply e;
apply f;
apply g;

[g1]
[g2]
[g3,g4]
[g5,g6,g4]
[g6,g4]
[g4]
[g7]
[]



tactic level ✗

goal level ✗

proof level ✓

Proof dimensions

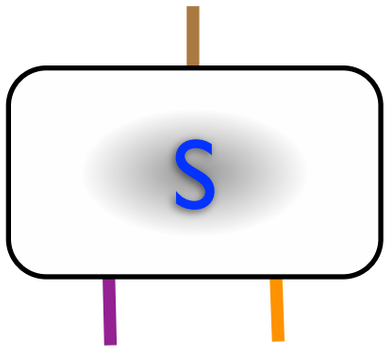
Other dimensions

- source of conjectures/lemmas
- additional user-inputs
- ...

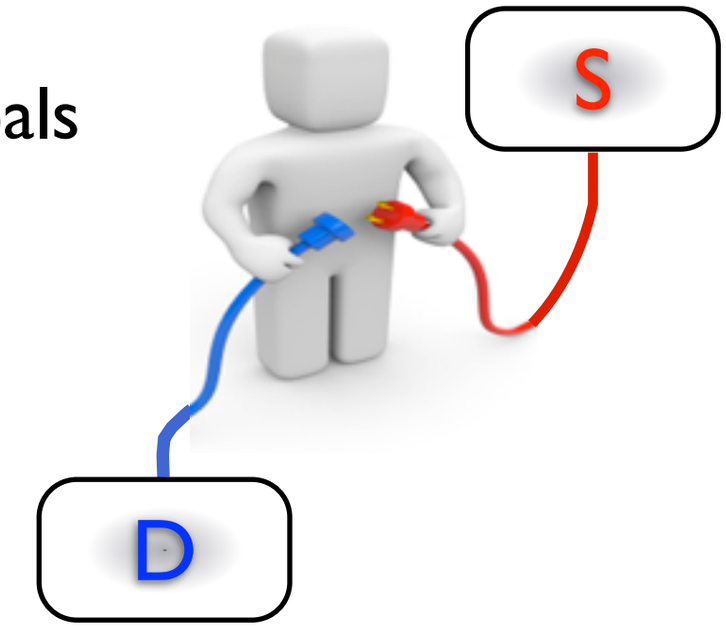
Need for abstractions

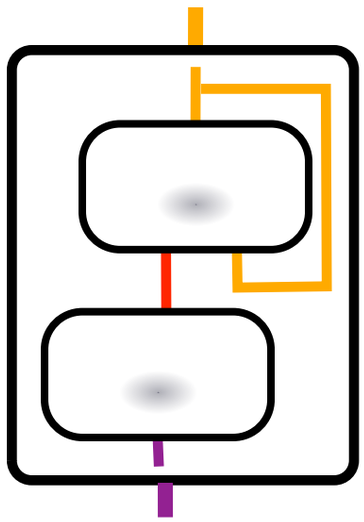
- Tactics are brittle
  - need I/O properties (as in methods)
  - ... to combine similar tactics
- Goals are complex & relational
  - only fragments of goal state needed

A box & wire language



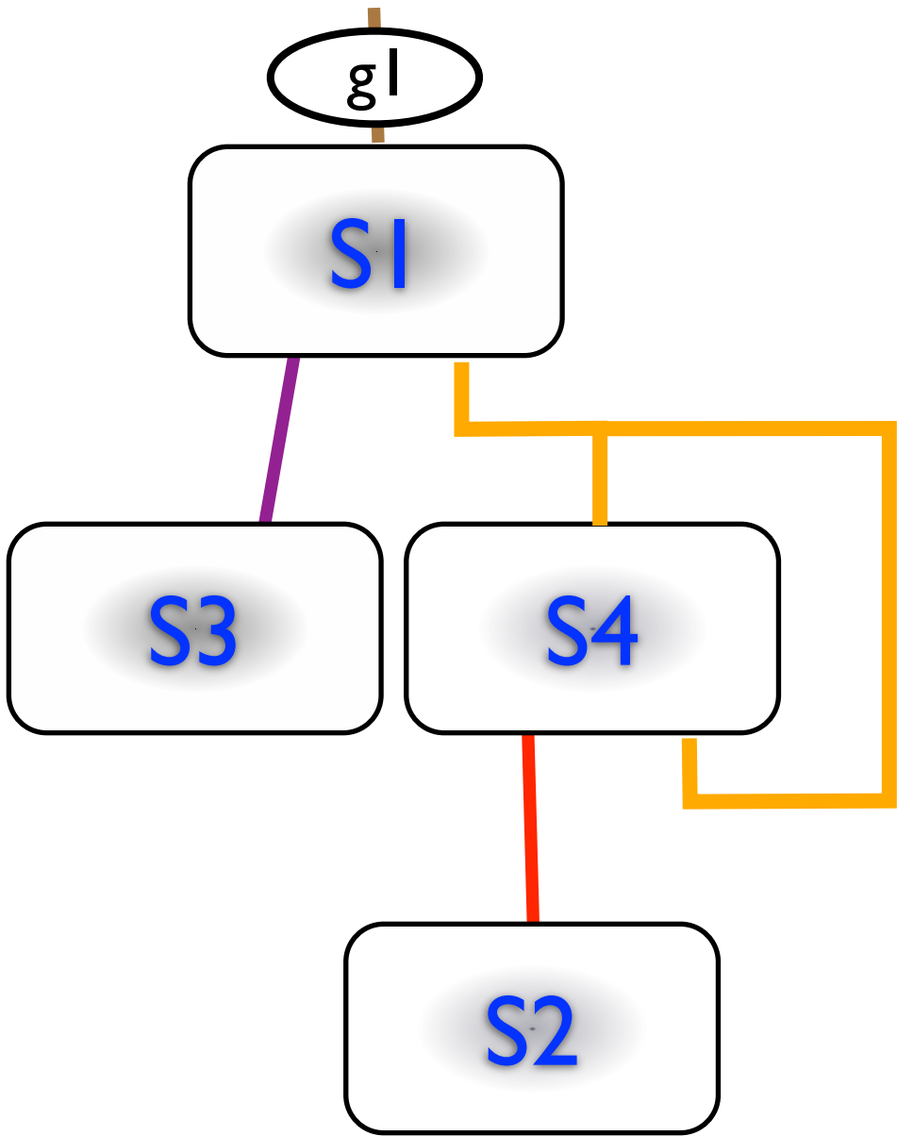
- A **wire** is an abstraction of a family of goals
  - provides a typing mechanism for goals
- A **tactic** is encapsulated by a box
  - explicit input and output "wires"
- Wire "typing" ensures correct **plugging**





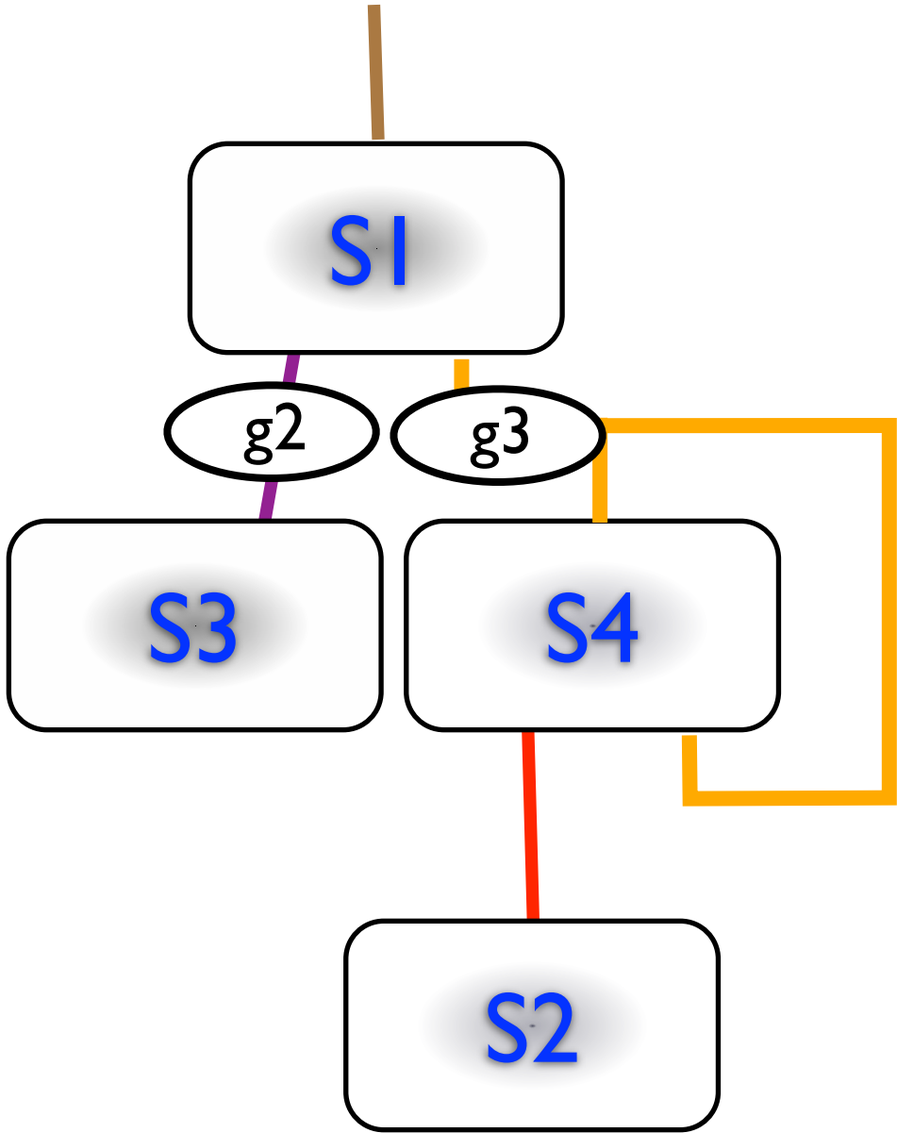
- Common pattern abstracted by **hierarchies**
- **Parameterised** over evaluation and search
- **Evaluation** by sending ***goal nodes*** on the wires

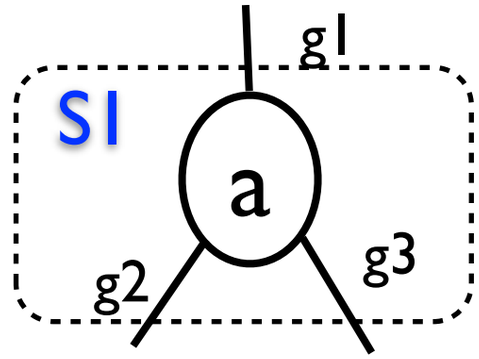
Evaluating a strategy



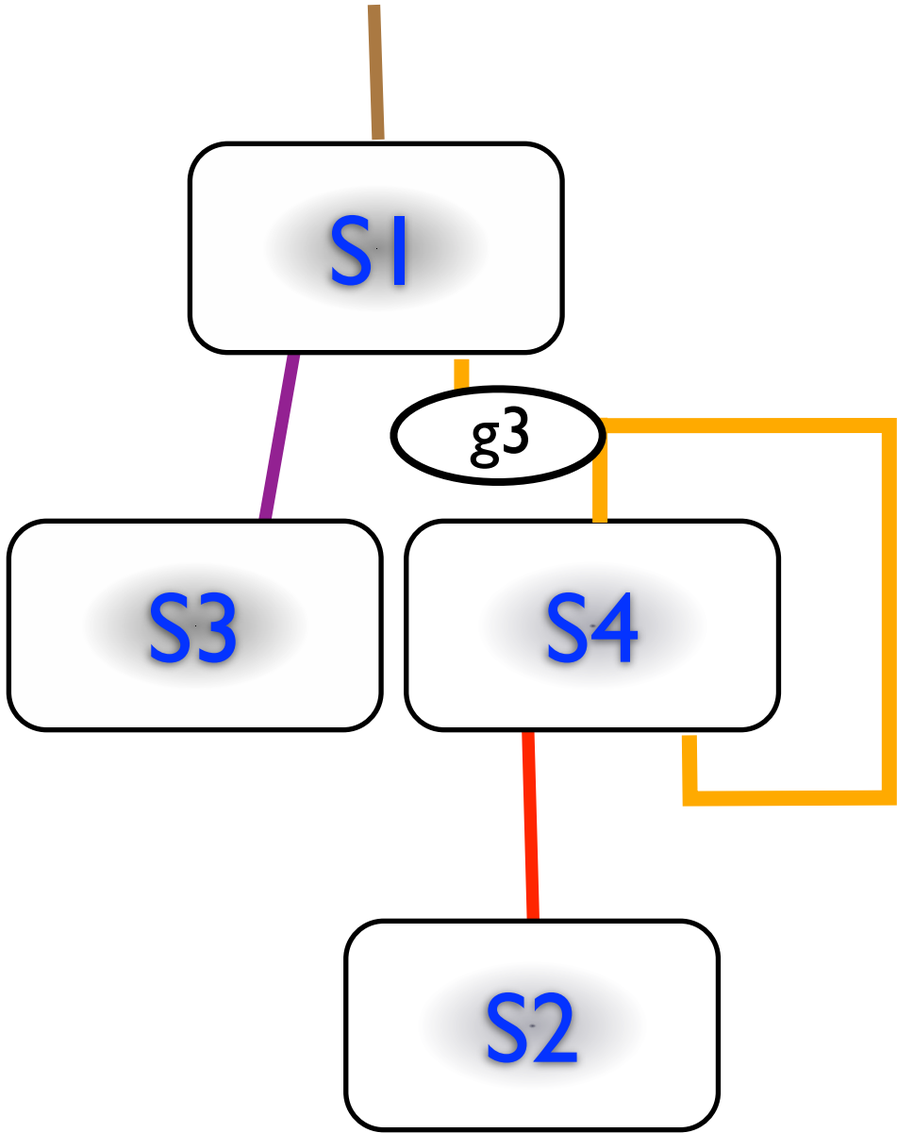
gl

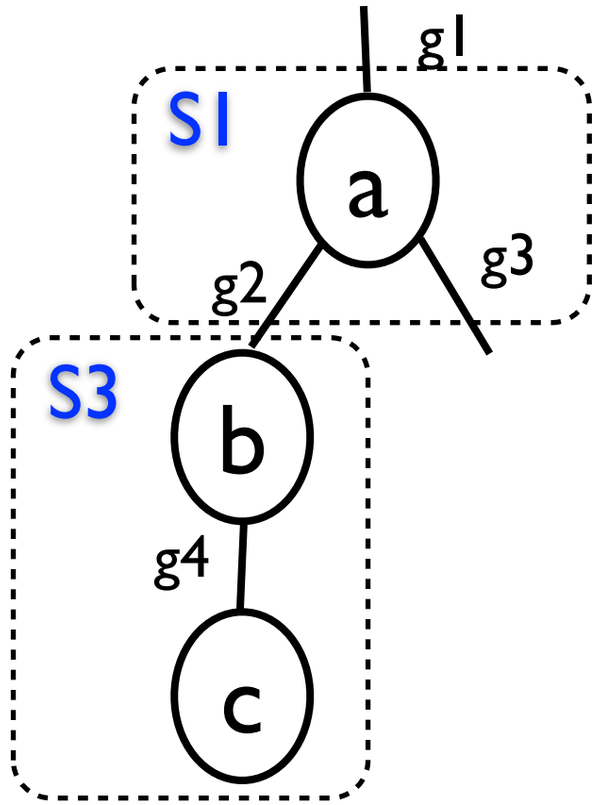
Evaluating a strategy



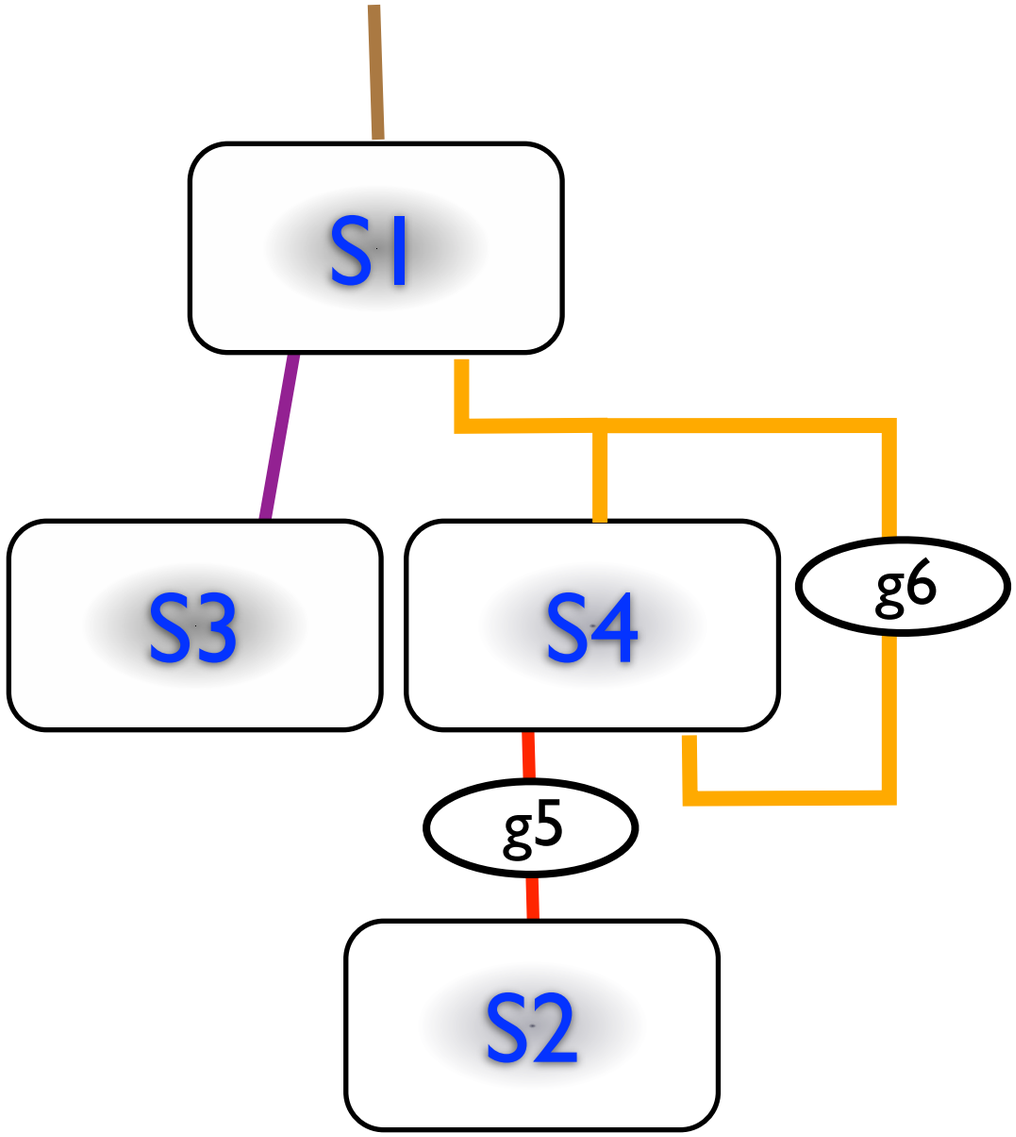


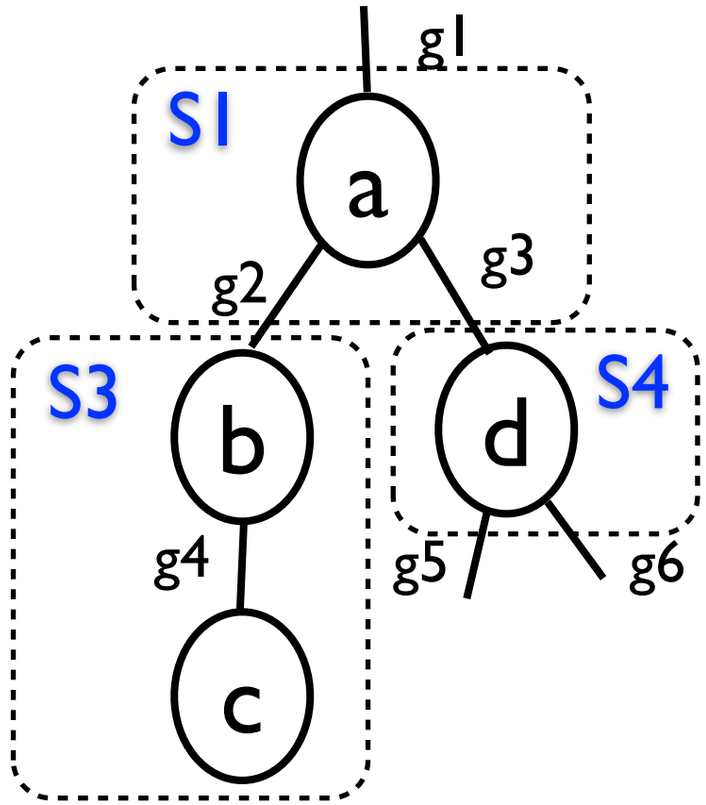
Evaluating a strategy



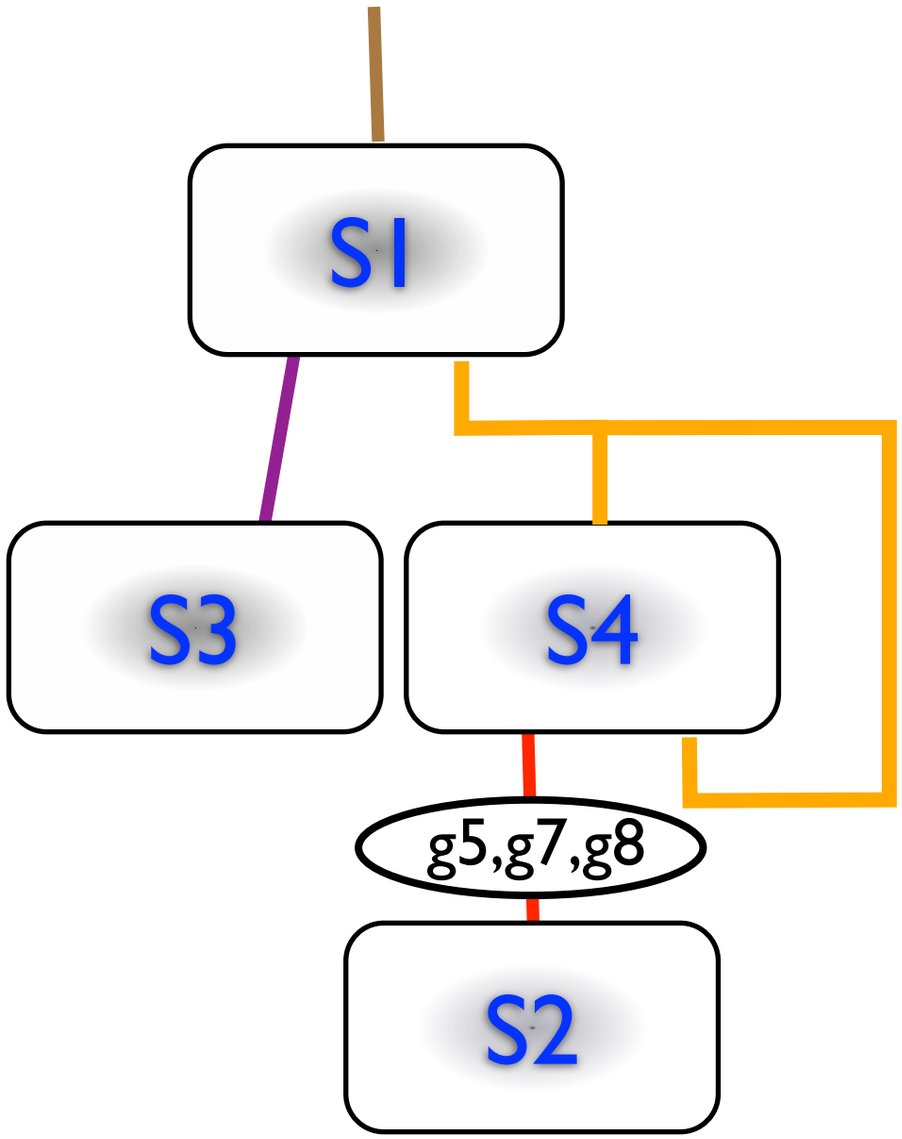


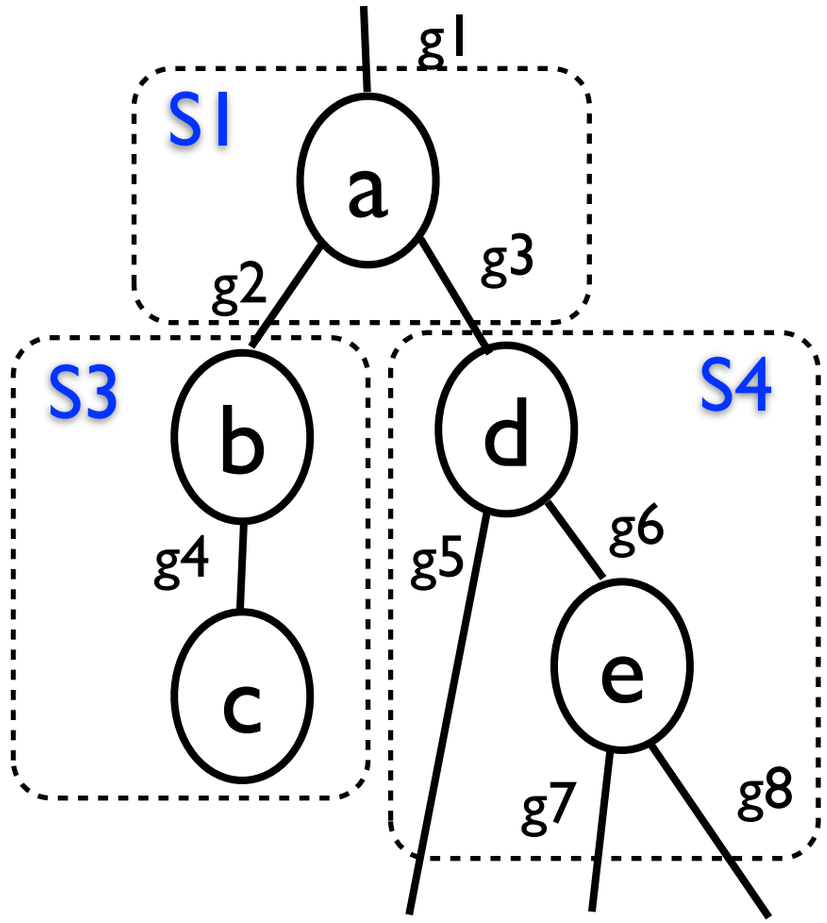
Evaluating a strategy



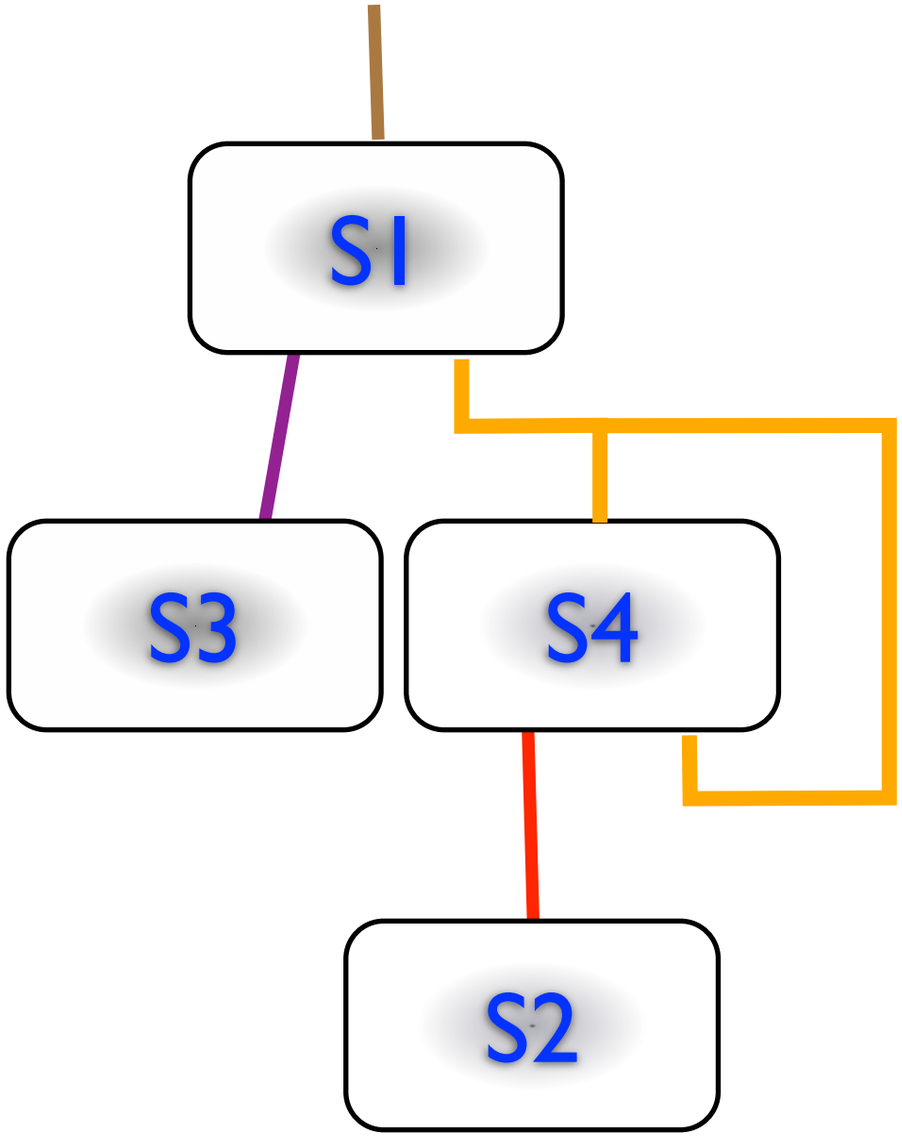


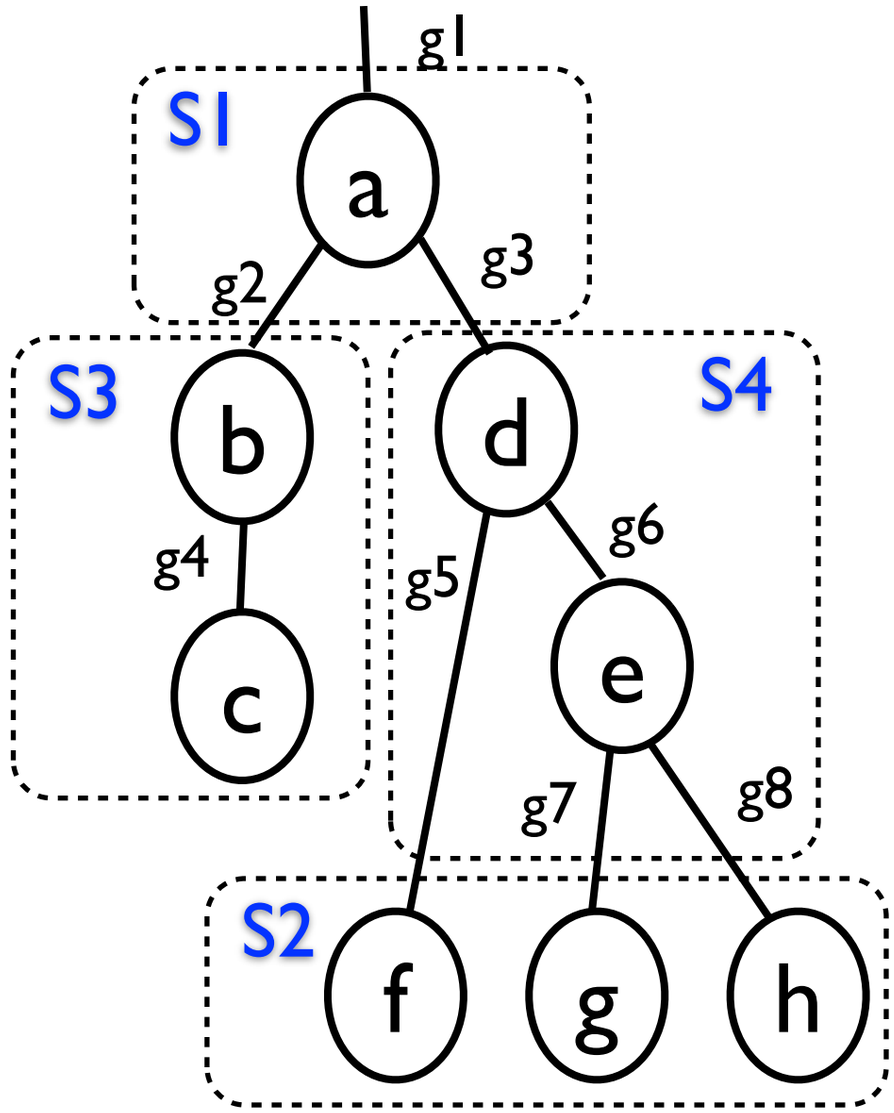
Evaluating a strategy





Evaluating a strategy





Wire details

- Abstracts relevant goal properties
  - "slicing/focus" of relevant parts of goal state
  - goal nodes must preserve these properties
- Must give **meaning** to wires to extract strategies
  - represented by a set of **features**
    - e.g: *has_induct_var*, *embeds_in_hyp*, *of_shape*

Extracting strategies

proofs (process) +++

translate proofs (process) into low-level strategies

initial strategies

strategy mapping/ refactoring + symbolic feature abstraction

feature vectors

(statistical) pattern recognition (clustering + families)

patterns/ families

symbolic strategy generalisation

strategies

strategies

strategies

Combination of symbolic & statistical methods

- ILP,HR,EBG,anti-unification... & neural nets, SVM,...
- hierarchical reinforcement learning + NLP seems relevant

Now... and beyond

- Develop underlying language theory
- Implementation in Isabelle/IsaPlanner
- Parsing of proofs
- Learning/generalisation experiments
- Conjecturing

- Make strategies/wire properties probabilistic?
- Require new generalisation/learning techniques?
- Go beyond Isabelle? to exchange proofs?
- Go beyond "LCF-style proofs"
  - (semi-implicit) waterfall model?
  - code proof outlines? reasoned modelling plans/critics?