Using Automated Theory Formation to Discover Invariants of Event-B models

Maria Teresa Llano[1]

Andrew Ireland[1] Alison Pease[2] Simon Colton[3] John Charnley[3]

[1]School of Mathematical and Computer Sciences
Heriot-Watt University

[2]School of Informatics
University of Edinburgh

[3]Department of Computing
Imperial College London

Rodin Workshop, September, 2010

Motivation

- The identification of invariants is a key aspect of the verification of formal models and the development of reliable systems.
- Discovering correct and meaningful invariants for a model represents a significant challenge.
- Increasing the level of automation in discovering invariants will:
  - Ripe productivity gains.
  - Increase the accessibility of formal modelling platforms such as Rodin.
- We investigate the use of Automated Theory Formation (ATF) techniques to reason about software requirements and suggest candidate invariants.

Outline

In this talk:

- Introduction to ATF and the HR system.
- Discovery of invariants in Event-B through ATF.
- Illustrative example.
- Roadmap.

ATF and HR

- ATF is a machine learning technique that builds theories about objects of interest within a given domain.
- HR (`http://www.doc.ic.ac.uk/~sgc/hr/`) is a system that implements ATF.
  - A set of background concepts describing the objects of interests are given to HR through a domain file.
  - New concepts are built from old ones using a set of generic production rules.
  - For each new concept HR calculates the set of examples and:
    - Estimates how interesting the concept is.
    - Looks for empirical relationships with existing concepts.
    - Forms conjectures when such relationships are found.

Concepts in HR

Concepts are composed of three components: a definition, a data table (or table of examples), and a categorisation.

Example: Concept of divisors of an integer (here the objects of interest are the integers from 1 to 10).

definition: $a \text{ is an integer} \wedge b \text{ is an integer} \wedge b/a$

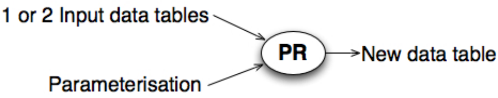
data table:

f(1) = [[1]]
f(2) = [[1], [2]]
f(3) = [[1], [3]]
f(4) = [[1], [2], [4]]
f(5) = [[1], [5]]
f(6) = [[1], [2], [3], [6]]
f(7) = [[1], [7]]
f(8) = [[1], [2], [4], [8]]
f(9) = [[1], [3], [9]]

$$\mathrm{f}(\underbrace{10}_{\mathrm{a}}) = \underbrace{[[1],[2],[5],[10]]}_{\mathrm{b}}$$

categorisation: [[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]]

HR Production rules (PR)



- Currently HR contains 22 PRs, among which are found:
    - **size**$<$ *columns_set* $>$**:** counts the number of appearances of the distinct tuples in the set of columns.
    - **split**$<$ *columns_set* $=$ *values_set* $>$**:** filters a data table according to the values given for the set of columns.
    - **negate**$<>$**:** finds the complement of a concept.

Example : Concept of prime numbers.

1. The size PR is applied in order to count the number of divisors of each integer of the first column.
2. The split PR is applied in order to get the integers that have only two divisors (the prime numbers).

Divisors	
1	1
2	1
2	2
3	1
3	3
.	.
.	.
10	1
10	2
10	5
10	10

$\mathit{size} < 1 >$

Number of divisors	
1	1
2	2
3	2
4	3
5	2
6	4
7	2
8	4
9	3
10	4

$\mathit{split} < 2 = 2 >$

Primes
2
3
5
7

PRs are first applied to background concepts, then more complex concepts are formed through their repeated application over existing concepts.

Making conjectures

- Each time a new concept is generated, HR checks to see whether a conjecture can be constructed.
- Types of conjectures:
  - **Equivalence conjectures:** if the new concept has the same data table as a previous concept.
  - **Implication conjectures:** if the data table of the new concept either subsumes or is subsumed by that of another concept.
  - **Non-existence conjectures:** if the data table for the new concept is empty.

Example of implication conjecture: *"All prime numbers are non-squares"*

$$\underbrace{(2 = |\{b : b|a\}|)}_{\text{prime}} \Rightarrow \underbrace{\neg(\exists b.(b|a \ \& \ b * b = a))}_{\text{non-square}}$$

Interestingness measures

- HR evaluates conjectures and concepts using various measures of *"interestingness"*.
- The heuristic search performed by HR is organised based on these measures (through an agenda mechanism).
- Examples of such measures:
  - *Applicability:* estimates the proportion of objects of interest for which the concept\conjecture is applied.
  - *Novelty:* evaluates how novel is the categorisation of a concept with respect to others.
  - *Comprehensibility:* measures how succinct the definition of a concept is.
  - *Variety:* measures the categorisations produced by a concept (less categorisation means less change).

Event-B models in HR

- Sets, constants, variables represent concepts of the domain.
- The list of examples for each of these concepts is obtained through the generation of animation traces (through the ProB animator).
- There is a concept of event (for which the examples are the events of the model).
- Each step of a trace represents the state of the system at a particular time.
- A step can represent a good or a bad state.
- States are also represented as concepts in the domain.

Discovering Invariants in Event-B Models

- States form the objects of interest in the theory, i.e., we are interested in concepts and conjectures that are associated with states.
- A conjecture implies a candidate invariant if it applies:
    - to all states, or
    - only to all good states, or
    - only to all bad states.
- Concepts can be considered as invariants, or parts of invariants, if they represent properties which remain unchanged in the domain: this is captured by HR's interestingness measure *variety*.

Example - Abrial's Cars on a Bridge

The model consists of an island connected to the mainland by a bridge. The bridge has one lane, i.e. cars can travel only in one direction.

Animation trace:

Maximum number of cars: 2

		Variables			
State	**Event**	**cti**	**coi**	**ctm**	**Desired case?**
s00	initialization	0	0	0	Yes
s01	ml_out	1	0	0	Yes
s02	ml_out	2	0	0	Yes
s03	il_in	1	1	0	Yes
s04	il_in	0	2	0	Yes
s05	il_out	0	1	1	Yes
s06	il_out	0	0	2	Yes
s07	ml_in	0	0	1	Yes
s08	ml_in	0	0	0	Yes

Note: cti = cars_to_island, coi = cars_on_island, ctm = cars_to_mainland

Example - Concepts

For each state, we supply HR with the concepts of state, cars going to the island, cars going to the mainland and good states (good). (Other concepts are not shown due to lack of space).

state(A)
s00
s01
s02
s03
s04
s05
s06
s07
s08

cti(A,B)	
s00	0
s01	1
s02	2
s03	1
s04	0
s05	0
s06	0
s07	0
s08	0

ctm(A,B)	
s00	0
s01	0
s02	0
s03	0
s04	0
s05	1
s06	2
s07	1
s08	0

good(A)
s00
s01
s02
s03
s04
s05
s06
s07
s08

Note: A = state, B = number_of_cars

Example - Split production rule

The split production rule is applied to extract the states for which the number of cars is 0.

cti(A,B)	
s00	0
s01	1
s02	2
s03	1
s04	0
s05	0
s06	0
s07	0
s08	0

split $< 2 = 0 >$

cti(A,0)
s00
s04
s05
s06
s07
s08

ctm(A,B)	
s00	0
s01	0
s02	0
s03	0
s04	0
s05	1
s06	2
s07	1
s08	0

split $< 2 = 0 >$

ctm(A,0)
s00
s01
s02
s03
s04
s08

Example - Negate production rule

HR then applies the negate rule to both concepts, producing the concept of the states for which the number of cars is different from 0.

cti(A,0)
s00
s04
s05
s06
s07
s08

negate $<>$

¬cti(A,0)
s01
s02
s03

ctm(A,0)
s00
s01
s02
s03
s04
s08

negate $<>$

¬ctm(A,0)
s05
s06
s07

Example - Formed conjectures

¬cti(A,0)
s01
s02
s03

⇒

ctm(A,0)
s00
s01
s02
s03
s04
s08

¬ctm(A,0)
s05
s06
s07

⇒

cti(A,0)
s00
s04
s05
s06
s07
s08

$$\Downarrow$$

$$\textbf{conj}_1\textbf{:}\ \forall A.((state(A) \land \neg cti(A,0)) \Rightarrow ctm(A,0))$$

$$\textbf{conj}_2\textbf{:}\ \forall A.((state(A) \land \neg ctm(A,0)) \Rightarrow cti(A,0))$$

In other words,

Invariant: $cti = 0 \lor ctm = 0$

Example conjectures with good and bad states

		Variables			
State	**Event**	**cti**	**coi**	**ctm**	**Desired case?**
s00	initialization	0	0	0	Yes
s01	ml_out	1	0	0	Yes
s02	ml_out	2	0	0	Yes
s03	il_in	1	1	0	Yes
s04	ml_out	2	1	0	Yes
s05	il_out	2	0	1	No
s06	il_in	1	1	1	No
s07	il_in	0	2	1	Yes
s08	il_out	0	1	2	Yes
s09	ml_out	1	1	2	No

$$\textbf{conj}_1\textbf{:}\ \forall A.((state(A) \wedge good(A) \wedge \neg cti(A,0)) \Rightarrow ctm(A,0))$$
$$\textbf{conj}_2\textbf{:}\ \forall A.((state(A) \wedge good(A) \wedge \neg ctm(A,0)) \Rightarrow cti(A,0))$$

OR

$$\textbf{conj}_3\textbf{:}\ \forall A.((state(A) \wedge \neg good(A)) \Leftrightarrow (\neg cti(A,0) \wedge \neg ctm(A,0)))$$

Results so far...

- We have analysed 4 models.
- Animation traces of around 30 states.
- 1000 formation steps.
- 10 - 15 seconds.
- 150 - 300 Concepts.
- 600 - 1800 Conjectures.

Ongoing and future work

- Analysis of larger models.
- Further analysis about measures of interestingness for the selection of the candidate invariants.
- Extend to deal with the discovery of gluing invariants.
- So far we have done theory formation, we are interested also in doing theory modification.
  - E.g. modify faulty events and faulty user-given invariants.
  - To this end we plan to apply Lakatos's philosophical theory of reasoning which uses counterexamples to trigger concept and conjecture development.
- Integration with Rodin, particularly with the ProB animator.

Summary

- We have applied ATF to reason about Event-B models.
- We have automatically discovered invariants of various Event-B models through the HR system.
- We believe that our results will provide a useful way of automatically reasoning about Event-B models.

Thanks!