Formal Analysis of System Specifications

Nancy A. Day

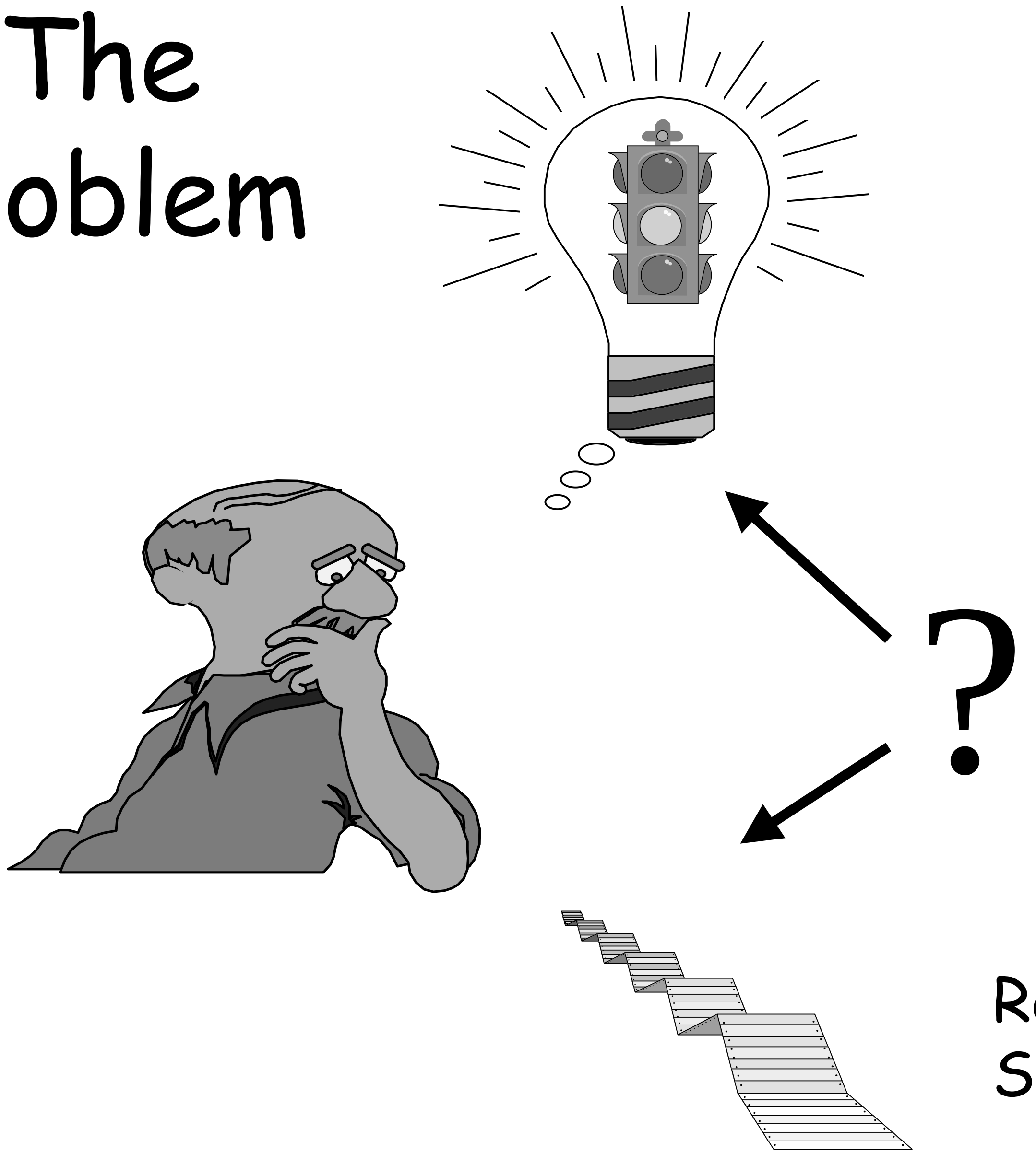
Supervisor: Dr. Jeff Joyce

Department of Computer Science
The University of British Columbia

day@cs.ubc.ca

http://www.cs.ubc.ca/spider/day

The Problem



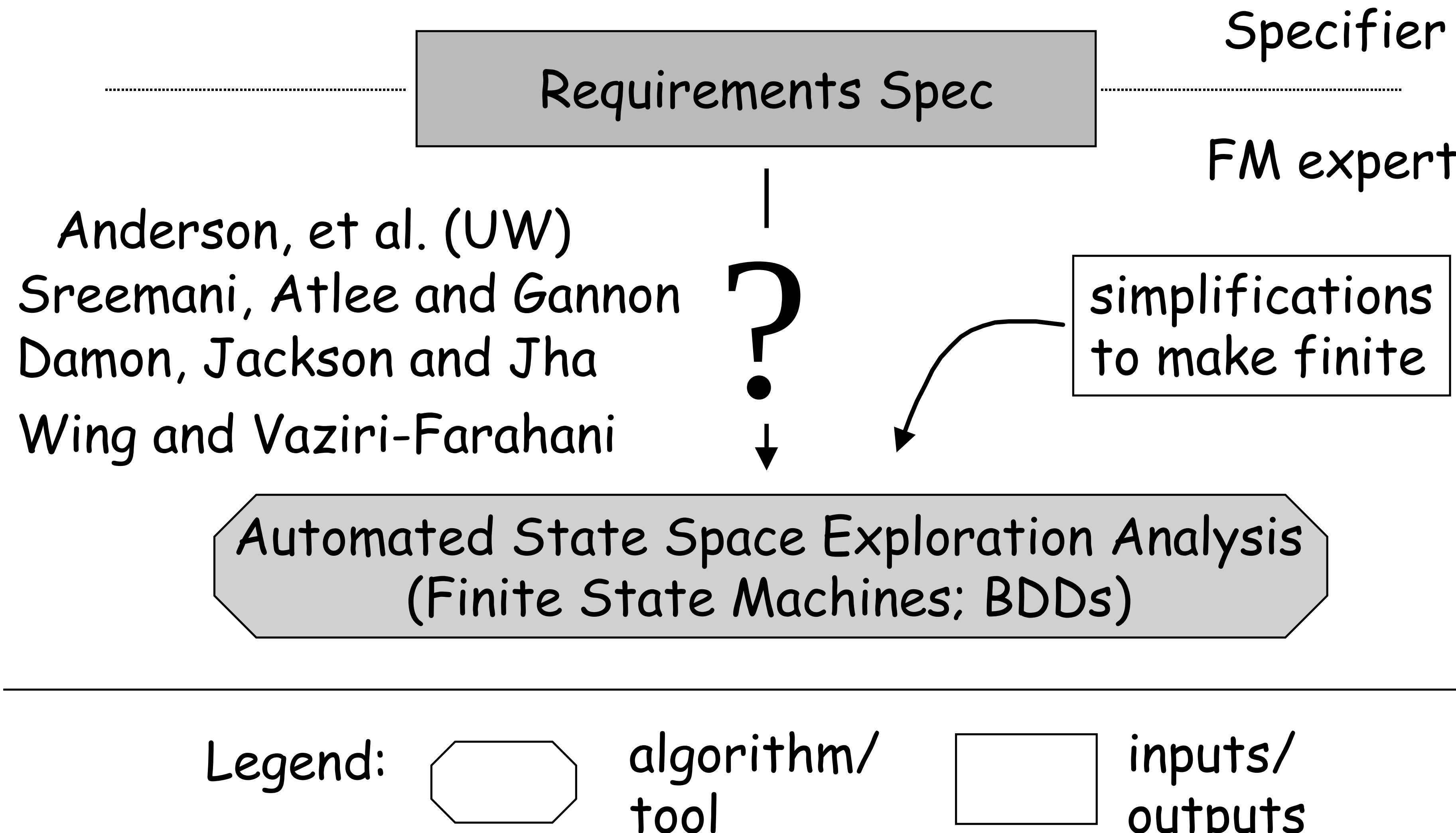
Are we building the right system ?

Requirements Specification

Analysis of Formal Req Specs

- parsing; typechecking
- simulation; symbolic simulation; prototyping
- completeness and consistency
- model checking
- ...

Context



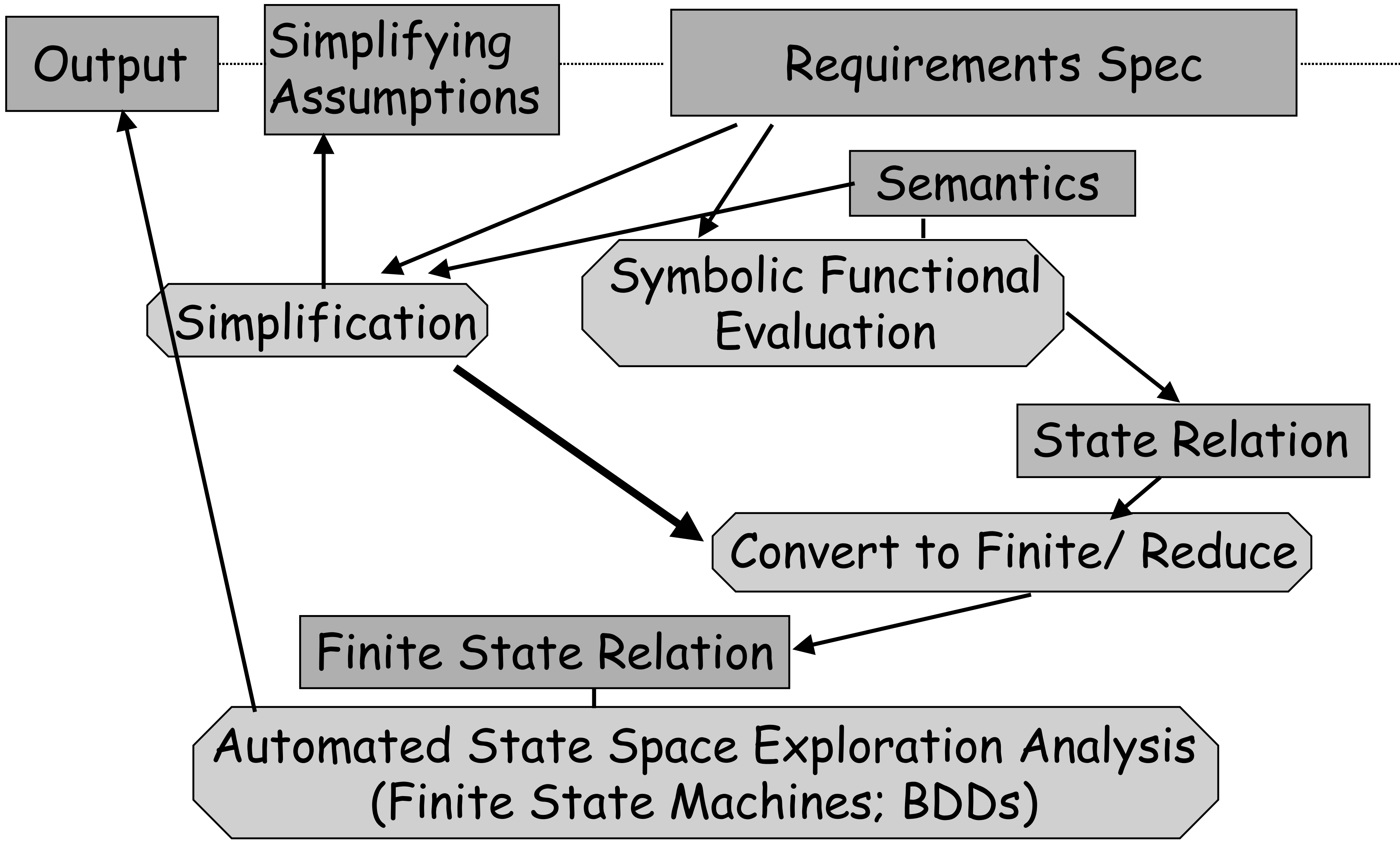
Specifier
Requirements Spec
FM expert
Anderson, et al. (UW)
Sreemani, Atlee and Gannon
Damon, Jackson and Jha
Wing and Vaziri-Farahani
?
simplifications to make finite
Automated State Space Exploration Analysis (Finite State Machines; BDDs)
Legend:
algorithm/ tool
inputs/ outputs

Observations

- different notations are used to describe different parts of the behaviour of the system
  - appropriate for application
  - life cycle (data encoding, code gen)
- simplifications used to make spec finite are often present in various parts of spec

Thesis Statement

Having an explicit machine-readable operational semantics for a notation within a common framework provides a systematic way to exploit inherent abstractions to carry out state-space exploration analysis.

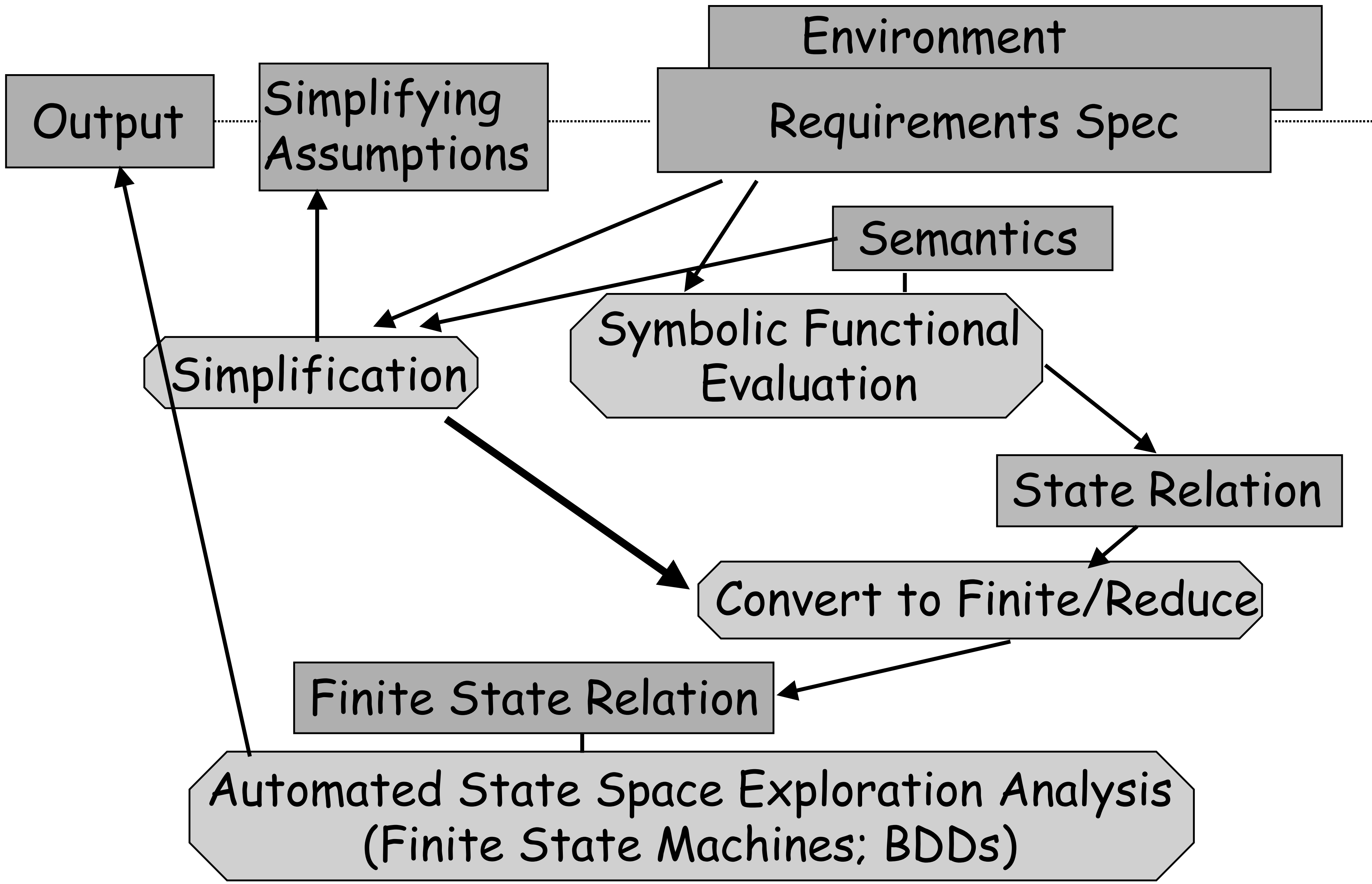

Output
Simplifying Assumptions
Requirements Spec
Semantics
Symbolic Functional Evaluation
Simplification
State Relation
Convert to Finite/ Reduce
Finite State Relation
Automated State Space Exploration Analysis (Finite State Machines; BDDs)

Example: Tabular Spec of Aircraft Separation Rules

Decision table: What is the vertical separation required between aircraft A and aircraft B ?

			Default
FlightLevel(A)	_< 280	_>450	
TypeOfAircraft(B)	_=Turbojet	_=Supersonic	
IsLevel(A)	_=T	.	
InCruiseClimb(A)	.	_=F	
Vertical_Separation (A,B)	1000	4000	2000

structure captures related elements in a row


Environment
Output
Simplifying Assumptions
Requirements Spec
Semantics
Symbolic Functional Evaluation
Simplification
State Relation
Convert to Finite/Reduce
Finite State Relation
Automated State Space Exploration Analysis (Finite State Machines; BDDs)

Example: Tabular Spec of Aircraft Separation Rules

Assumption: The following conditions are mutually exclusive and form a tautology:

(FlightLevel(A) < 280)
(FlightLevel(A) > 450)

However there is at least one entry which is a "don't care" entry and this covers all other cases.

Environment

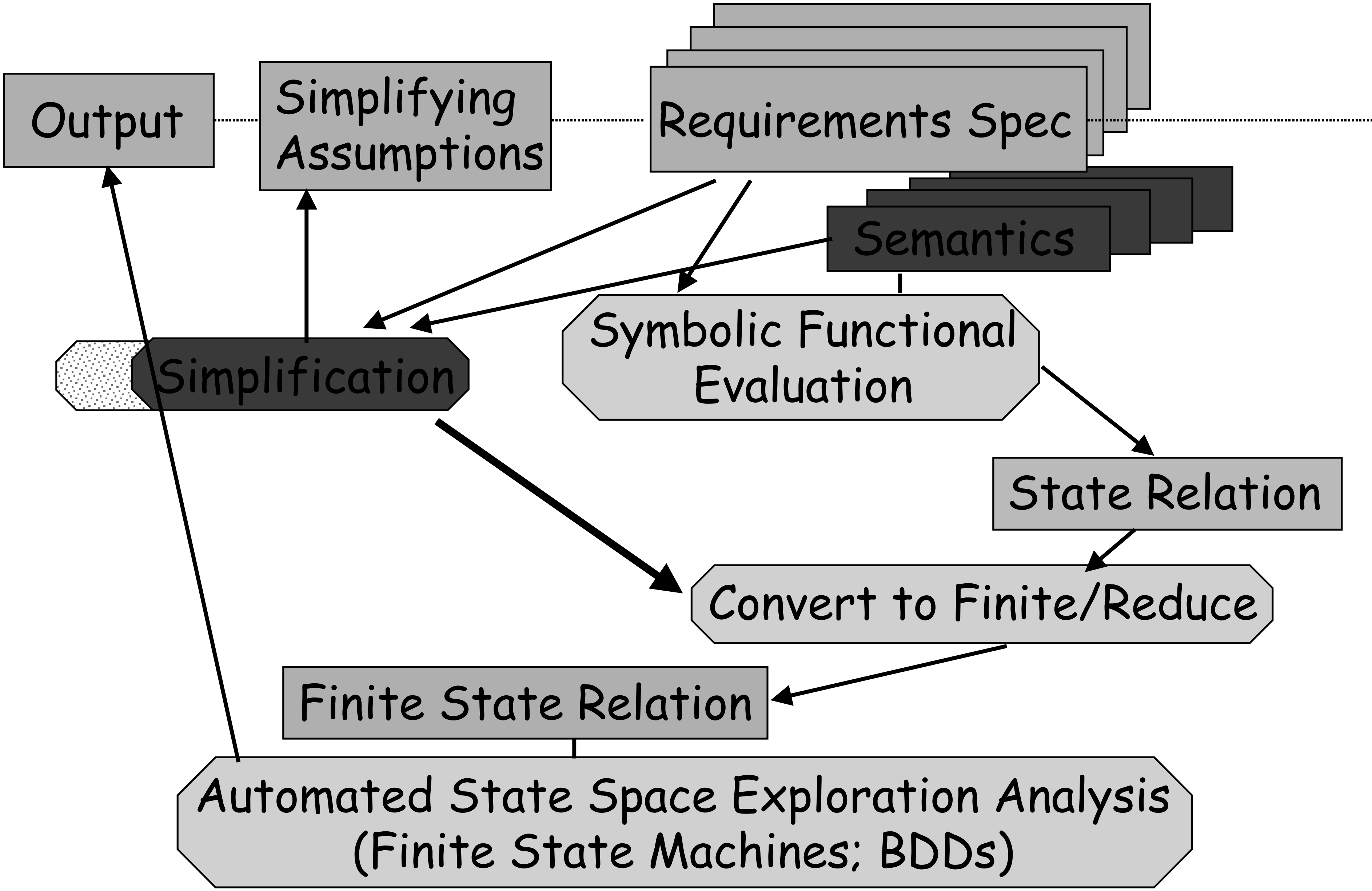
```
: typeOfAircraft := Turbojet | Supersonic | Other;

forall A:flight. IsLevel(A) ==>
        Not (InCruiseClimb(A));

forall A:flight. InCruiseClimb(A) ==>
        Not (IsLevel(A));
```

Analysis Results

- results produced at level of uninterpreted functions
- completeness analysis found:
  - missing assumptions "everyone knew about" (domain knowledge)
  - incorrect partitions
- consistency analysis found:
  - two places where the requirements were ambiguous


Output
Simplifying Assumptions
Requirements Spec
Semantics
Simplification
Symbolic Functional Evaluation
State Relation
Convert to Finite/Reduce
Finite State Relation
Automated State Space Exploration Analysis
(Finite State Machines; BDDs)

Advantages / Contributions

- use the explicit defn of semantics directly in analysis; also simulation, prototyping; analysis of semantics
- general framework for:
  - multiple notations; multiple analysis techniques
  - non-formal methods person; formal methods expert
- return results at correct level of abstraction
- exploit inherent abstractions

Current Status

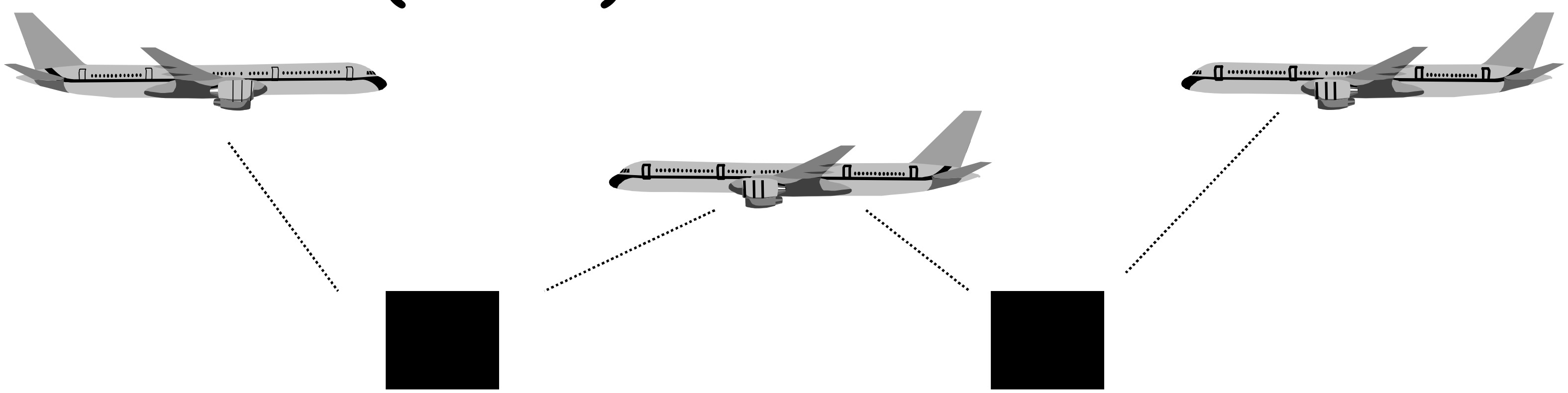
- mainly concentrating on how much can do for simplification engine for statecharts, tables, ASN.1 + functionality given in predicate logic
- working on more examples

Remaining Questions / Evaluation

- to what extend to the notations have to be operational to be used in this framework ?
- how much can structure reduce the size of the state space ?
- how to evaluate a general framework ?

More examples

- aeronautical telecommunications network (ATN)



  - statecharts; parameterization; ASN.1; uninterpreted functions, etc.
  - error states are particularly important