Specification and Tool-based Analysis of an Aircraft Separation Minima

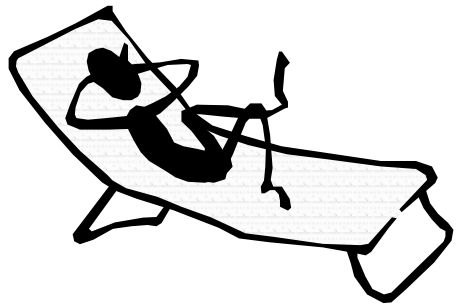
Nancy Day
University of British Columbia

Jeff Joyce, Gerry Pelletier
Hughes Aircraft of Canada

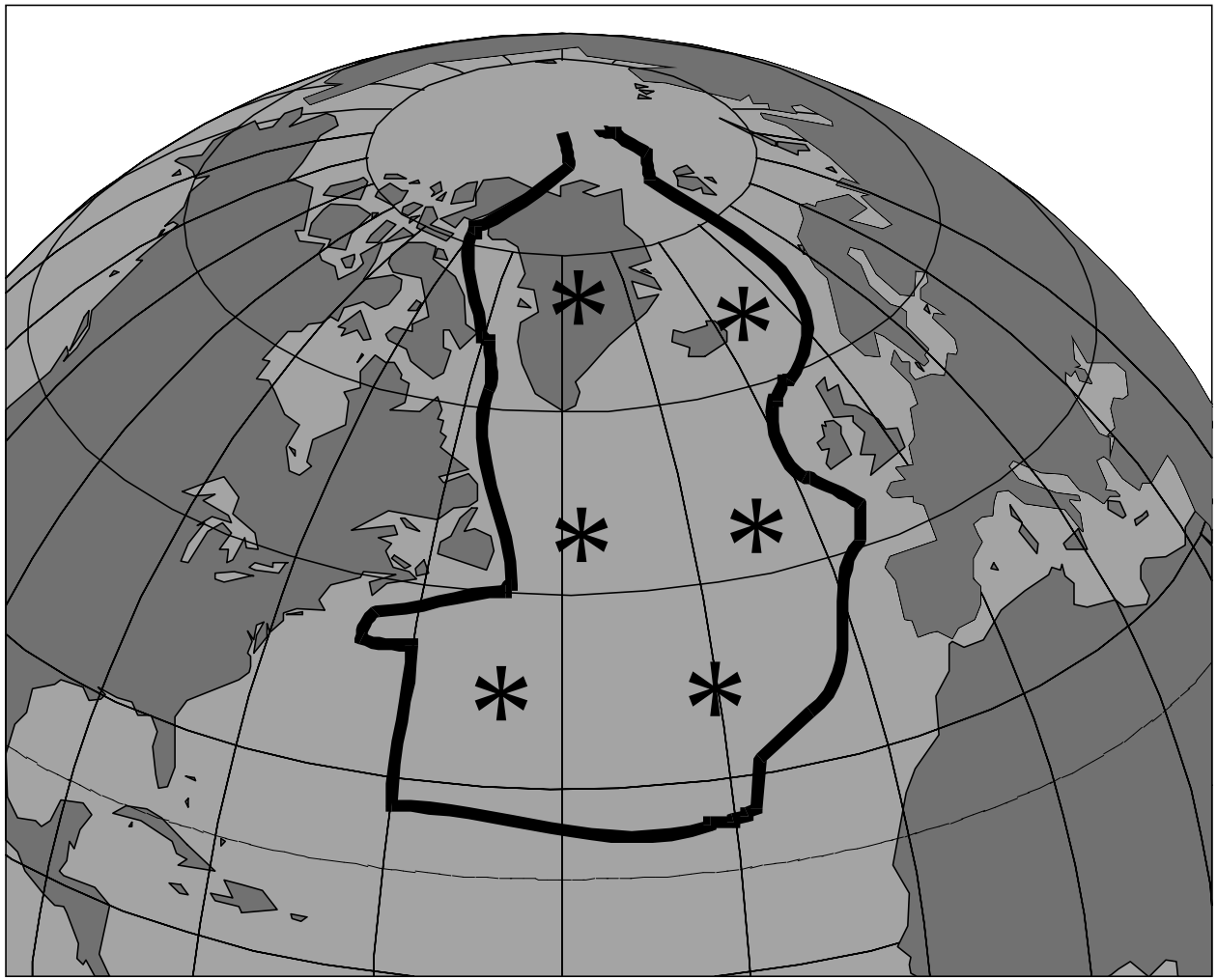
day@cs.ubc.ca
http://www.cs.ubc.ca/spider/day

Outline

1. system and desired analysis
2. specification
3. analysis results
4. summary

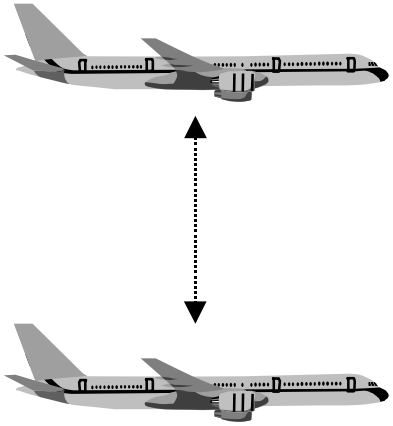


North Atlantic Region



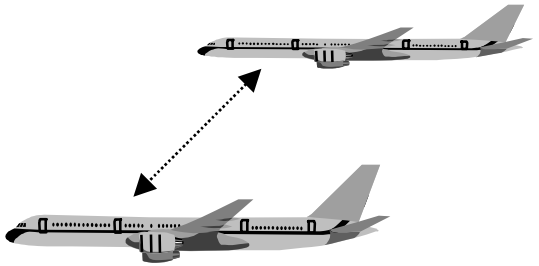
Separation Minima for NAT

rules for air traffic controllers



Vertical Separation

Lateral Separation



Longitudinal Separation

Existing Specification

- complex decision logic
- stateless

Form of the specification:

- informal natural language (64 pages)
  - thoroughly reviewed and in use
- pseudocode interpretation (9 pages)

Analysis

- completeness
  - are all combinations of inputs for two flights covered ?
- consistency
  - is it ever possible that the specification has multiple possible outcomes for the same inputs ?

2. Specification

ICAO Specification

3.1 Vertical Separation Minima

The vertical separation minima shall be:

a) 1000 feet below FL290; or

b) 2000 feet at or above FL290 except above FL450 between supersonic and between supersonic and any other aircraft where 4000 feet shall be used.

Pseudocode

```
BEGIN Vertical-separation-routine.

    Vertical separation is 2000 ft

    IF at least one of the aircraft is at or below FL280 THEN
            vertical separation is 1000 ft
    ENDIF

    IF both aircraft are above FL450 AND
            at least one aircraft is supersonic THEN
            vertical separation is 4000 ft
    ENDIF

END Vertical-separation-routine.
```

Case Studies: Checking Completeness and Consistency

- Traffic Collision Avoidance System (TCAS II)
  - UC-Irvine; UW; US FAA (Heimdahl and Leveson)
  - AND/OR tables
- Operational Flight Program for US Navy's A-7 Aircraft
  - Naval Research Lab (Heitmeyer et al.)
  - Software Cost Reduction (SCR) Method
  - multiple kinds of tables

Pseudocode

BEGIN Vertical-separation-routine.

Vertical separation is **2000 ft**

IF at least one of the aircraft is at or below **FL**280 THEN
vertical separation is **1000 ft**
ENDIF

IF both aircraft are above **FL**450 AND
at least one aircraft **is supersonic** THEN
vertical separation is **4000 ft**
ENDIF

END Vertical-separation-routine.

Building a table: step 1

FlightLevel A

FlightLevel B

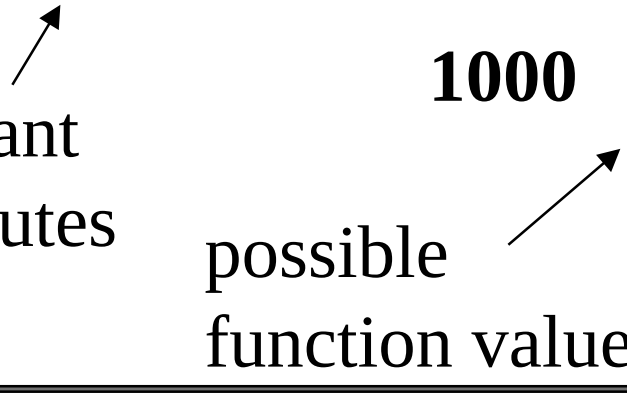
IsSupersonic A

IsSupersonic B

1000 **4000** **2000**

relevant attributes

possible function values



Pseudocode

BEGIN Vertical-separation-routine.

Vertical separation is **2000 ft**

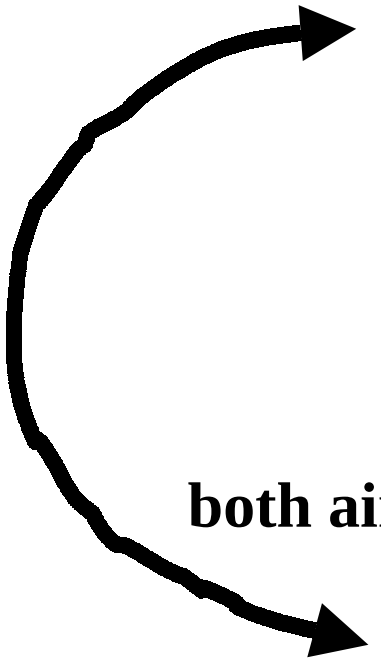
IF **at least one of the aircraft is at or below FL280** THEN
vertical separation is **1000 ft**
ENDIF

IF **both aircraft are above FL450** AND
at least one aircraft is supersonic THEN
vertical separation is **4000 ft**
ENDIF

END Vertical-separation-routine.

FlightLevel

at least one of the aircraft is at or below FL280



FlightLevel A <= 280
FlightLevel B - don't care

1000 ft

FlightLevel A - don't care
FlightLevel B <= 280

both aircraft are above FL450

FlightLevel A > 450
FlightLevel B > 450

4000 ft

Building a table: step 2

	1	2	3	
FlightLevel A	__ <= 280	●	__ > 450	
FlightLevel B	●	__ <= 280	__ > 450	
IsSupersonic A				
IsSupersonic B				
	1000	**1000**	**4000**	**2000**

● = “don’t care”

Pseudocode

BEGIN Vertical-separation-routine.

Vertical separation is **2000 ft**

IF **at least one of the aircraft is at or below FL280** THEN
vertical separation is **1000 ft**
ENDIF

IF **both aircraft are above FL450** AND
at least one aircraft is supersonic THEN
vertical separation is **4000 ft**
ENDIF

END Vertical-separation-routine.

IsSupersonic

at least one aircraft is supersonic

IsSupersonic A = T
IsSupersonic B - don’t care

4000 ft

IsSupersonic A - don’t care
IsSupersonic B = T

Building a table: step 3

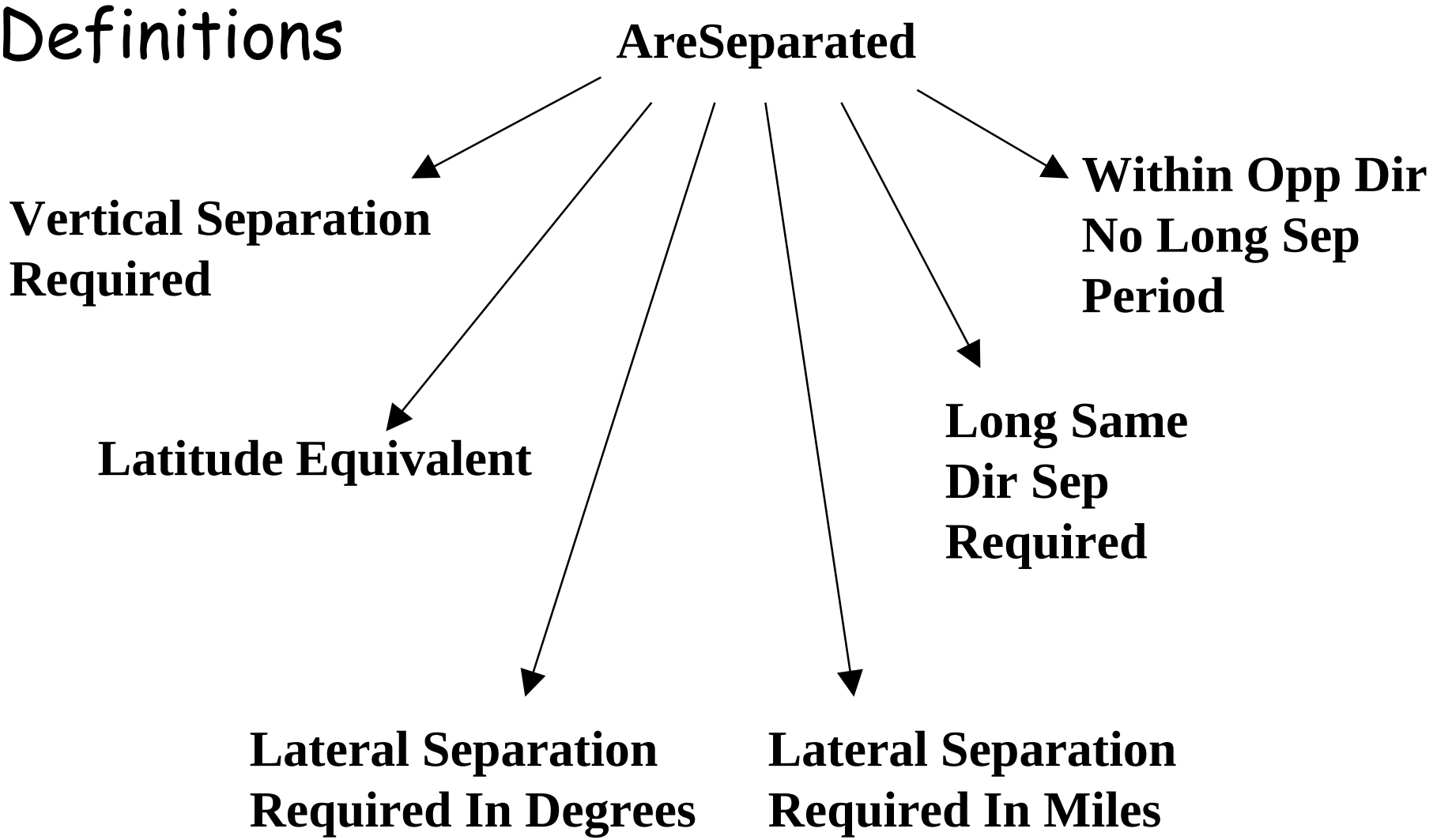
	1	2	3	
FlightLevel A	__ <= 280	●	__ > 450	
FlightLevel B	●	__ <= 280	__ > 450	
IsSupersonic A	●	●		
IsSupersonic B	●	●		
	1000	**1000**	**4000**	**2000**

Completed Table

	1	2	3	4	**Def**
FlightLevel A	__ <= 280	●	__ > 450	__ > 450	
FlightLevel B	●	__ <= 280	__ > 450	__ > 450	
IsSupersonic A	●	●	__ = T	●	
IsSupersonic B	●	●	●	__ = T	
VerticalSeparation Required (A,B)	**1000**	**1000**	**4000**	**4000**	**2000**

(page 16 of tech report)

Definitions



(page 15 of tech report)

Formal Specification

- combination of:

  - function and predicate tables

  - ASCII based predicate logic
    - » defined types, functions and predicates types
    - » primitive types, functions and predicates

Primitives

:flight;

FlightLevel : flight -> num;

IsSupersonic: flight -> bool;

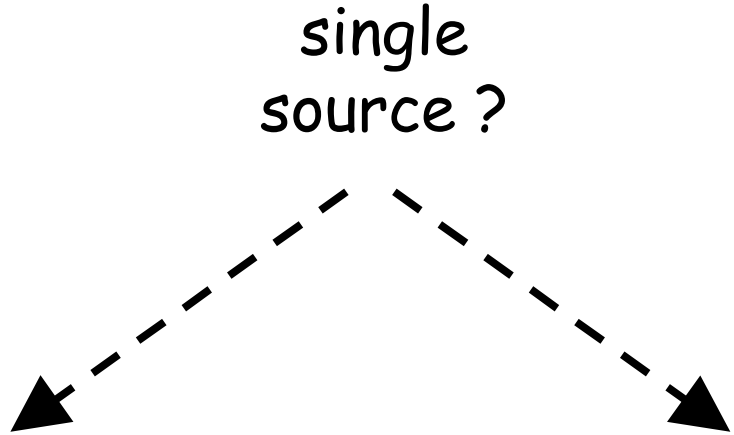
ensure terms are always used consistently

(pages 27-29 of tech report)

Size of the formal specification

- 15 tables
- 16 definitions
- 47 primitive functions and predicates
- formal spec is 300 lines
- 16 page document
  - (see pages 14 - 29 in tech report)

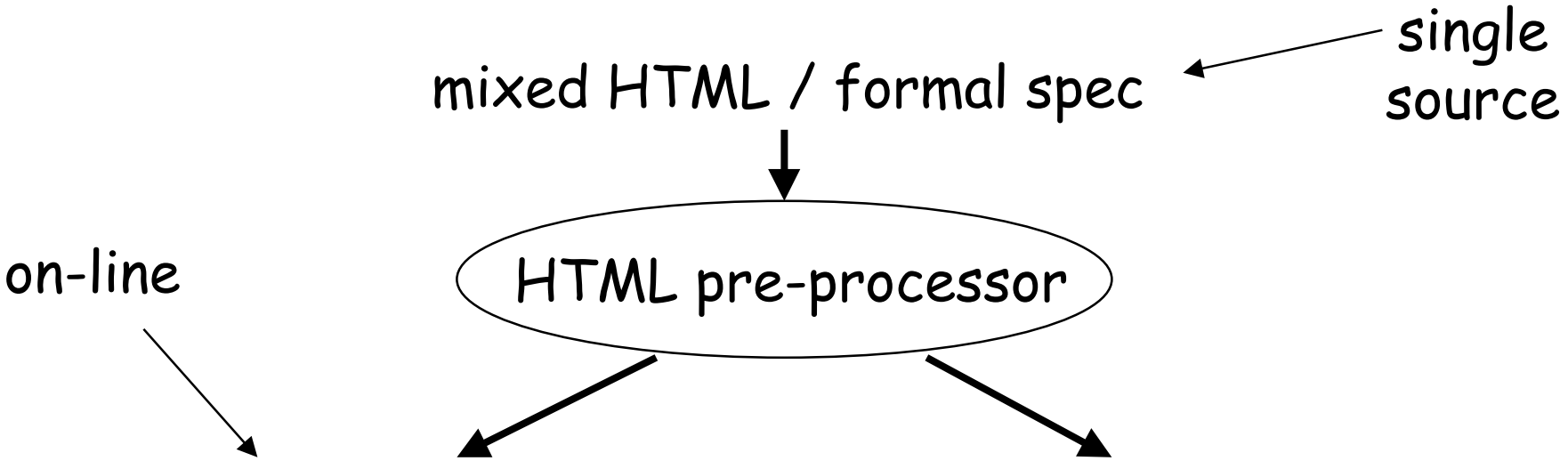
Presentation of the Specification



HTML
- cross-ref use to defn
- tables in formal spec
as HTML tables

formal spec (tool input)
- bottom-up order

Presentation of the Specification



3. Analysis Results

Definitions of Completeness and Consistency of a Table

- completeness:
  - default cases
- consistency:
  - no two columns with differing result values for the function overlap
- with respect to possible values for row entries

Example of Completeness Analysis Results

(page 30 of tech report)

Fusion - Version 1.0

...

>%include minima.s

...

>%comp VerticalSeparationRequired env

...

Invoking interval checker ...

Interval checker partitions the range into:

((FlightLevel A) > 450)

((280 < (FlightLevel A)) AND ((FlightLevel A) <= 450))

((FlightLevel A) <= 280)

...

Example of Completeness Analysis Results

(page 30 of tech report)

The following cases yield
the default value of 2000

Case 1

Row 1 : ((280 < (FlightLevel A)) AND ((FlightLevel A) <= 450))

Row 2: ((FlightLevel B) > 450)

Row 3: DC

Row 4: DC

DC = “don’t care”

	1	2	3	4	**Def**
FlightLevel A	__ <= 280	●	__ > 450	__ > 450	
FlightLevel B	●	__ <= 280	__ > 450	__ > 450	
IsSupersonic A	●	●	__ = T	●	
IsSupersonic B	●	●	●	__ = T	
VerticalSeparation Required (A,B)	**1000**	**1000**	**4000**	**4000**	**2000**

(page 16 of tech report)

Example of Consistency Analysis Results

(page 35 of tech report; table on page 17)

> %cons “LateralSeparation RequiredInDegrees”

Columns 1 and 3 conflict in the following:
Case 1
Row 1 : (((AllOf [A;B]) IsOutsideMNPSAirspace) = T)
Row 2 : (((AllOf [A;B]) (IsOnRoute Routes1)) = T)
Row 3: (((AllOf [A;B]) (IsOnRoute Routes2)) = F)
Row 4: DC
Row 5: (((AllOf [A;B]) IsSupersonic) = T)
Row 6: (((AllOf [A;B]) FlightLevelAbove275) = T)
Row 7: DC
Row 8: DC

Summary of Analysis Results

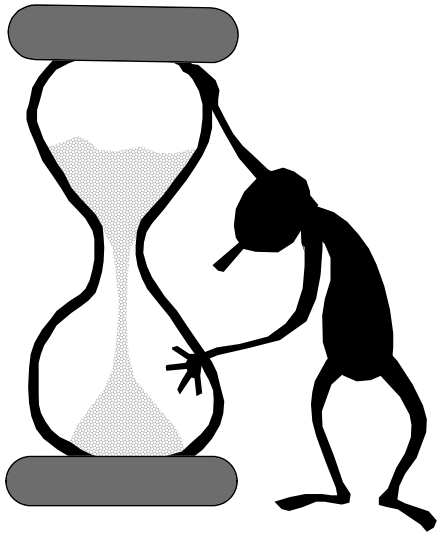
(page 11 of TR)

- completeness analysis found:
  - missing assumptions "everyone knew about" (domain knowledge)
- consistency analysis found:
  - three places where the requirements are inconsistent
- symmetry analysis found:
  - assumptions about the primitive terms

Analysis Method

- formal methods:
  - notation with an unambiguous syntax and semantics
  - common framework for definitions and tables
  - logical calculation
  - concise data structures

4. Summary



General Applicability ?

- decision logic (and/or):

  At least one of the following conditions is satisfied:

  ...

  iiii) All of the following are satisfied

  ...

- not just for requirements ...
  - design
  - software inspection
  - system test
  - ...
- modularity: multiple levels of tables

FormalWare

- formalized threads (Kendra Cooper)
- analysis of an aeronautical telecommunications network (N. Day)
- automatic test case generation (Mike Donat)
- safety analysis (Ken Wong)
- ...

http://www.cs.ubc.ca/formalWARE

Conclusions

- advantages of tables:
  - more "what" than "how"
  - concise
  - modular
- automatic analysis to help specifier
  - "push button" tools
  - iterative approach for review process
  - life cycle support

That's all!