Formal Tools and Methods for Decision Tables

D. N. Hoover
Odyssey Research Associates, Inc.
301 Dates Dr.
Ithaca NY 14850-1326
Internet: hoove@oracorp.com

Overview

Honeywell's problem:

- ❐ Code involving choice of alternatives depending on complex conditions.
- ❐ Is the choice always unambiguous? (consistency)
- ❐ Is there always a choice? (completeness)

Raw material:

- ❐ Some Honeywell specs and code.
- ❐ Specs "nearly formal."
- ❐ Logical problem "nearly propositional" (some numerical comparisons).
- ❐ "The specs come from decision tables."
- ❐ Decision tables *are!* formal specs.

Decision Tables

Operational Procedure		Takeoff		Climb		Climb_ Int_ Level	Cruise
Input Variables	States						
Flightphase	climb cruise	climb	climb	climb	climb	climb	cruise
AC_Alt > 400	TRUE FALSE	TRUE	TRUE	*	*	*	*
compare(AC_Alt, ACC-ALT)	LT EQ GT	LT	LT	EQ GT	EQ GT	*	GT
Alt_Capt_Hold	TRUE FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
compare(Alt_Target, prev_Alt-Target	LT EQ GT	*	GT	*	GT	*	EQ

TableWise, a Decision Table Tool

After preliminary experimentation with PVS, we decided to build a specialized tool for the problem.

- ❐ Check completeness and consistency of decision tables
- ❐ Generate Ada code
- ❐ Generate English language specs like Honeywell's.
- ❐ Best of all, it is available by anonymous ftp from

    `air16.larc.nasa.gov`

directory

    `fm/ora`

TableWise—Internals

- Logic by D. N. Hoover.
- Interface by Zewei Chen.
- The logic is in the functional subset of SML.
- **That constitutes a formal design spec.**
- Code generation tested by verifying the generated code using ORA's Penelope Ada verification system.
- BDD-type algorithms.

Analysis Results

Overlapping scenarios		Takeoff.2 Climb_Int_Level.1	Climb.2 Climb_Int_Level.1
Input Variables	States		
Flightphase	climb cruise	climb	climb
AC_Alt > 400	TRUE FALSE	TRUE	*
compare(AC_Alt, ACC-ALT)	LT EQ GT	LT	EQ GT
Alt_Capt_Hold	TRUE FALSE	TRUE	TRUE
compare(Alt_Target, prev_Alt-Target	LT EQ GT	GT	GT

Analysis Results (cont'd)

Missing scenarios		0	1	2	3
Input Variables	States				
Flightphase	climb cruise	climb	cruise	cruise	cruise
AC_Alt > 400	TRUE FALSE	TRUE	*	*	*
compare(AC_Alt, ACC-ALT)	LT EQ GT	LT	LT EQ	GT	GT
Alt_Capt_Hold	TRUE FALSE	FALSE	*	TRUE	FALSE
compare(Alt_Target, prev_Alt-Target	LT EQ GT	*	*	LT GT	*

Structural Analysis

A way to localize flaws. There is always a structural flaw in an incomplete table.

Operational Procedure		Takeoff		Climb		Climb_Int_Level	Cruise
Input Variables	States						
Flightphase	climb cruise	climb	climb	climb	climb	climb	cruise
AC_Alt > 400	TRUE FALSE	TRUE	TRUE	*	*	*	*
compare(AC_Alt, ACC-ALT)	LT EQ GT	LT	LT	EQ GT	EQ GT	*	GT
Alt_Capt_Hold	TRUE FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
compare(Alt_Target, prev_Alt-Target	LT EQ GT	*	GT	*	GT	*	EQ

Structu

5 variable_structural flaws		0	1	2	3
Input Variables	States				
Flightphase	climb cruise	*	*	cruise	cruise
AC_Alt > 400	TRUE FALSE	FALSE	*	*	*
compare(AC_Alt, ACC-ALT)	LT EQ GT	*	*	LT EQ	*
Alt_Capt_Hold	TRUE FALSE	*	*	*	FALSE
compare(Alt_Target, prev_Alt-Target	LT EQ GT	*	LT	*	*

This idea is new, based on "Hoover's Theorem 'B'."

Continuing Work

- Assertions about decision tables.
- Decision tables including "behaviors," i.e. code implementing the action chosen.
- Generating tests from decision tables.
- Better code generation.
- Generating traceability information.
- Incorporate some of Parnas's ideas about tabular specifications.
- Tables as specification of a state transition (cf. Leveson's RSML).
- Structuring decision tables to compress information.
- Incorporate support for linear arithmetic in the checking.
- Solve the "industrial inference problem." (See Parnas, "Some Theorems We should Prove."