Machine Minimization

ECE 152A – Fall 2006

Reading Assignment

- Brown and Vranesic
  - 8 Synchronous Sequential Circuits
    - 8.6 State Minimization
      - 8.6.1 Partitioning Minimization Procedure
      - 8.6.2 Incompletely Specified FSMs

Reading Assignment

- <u>Roth</u>
  - 15 Reduction of State Tables / State Assignment
    - 15.1 Elimination of Redundant States
    - 15.2 Equivalent States
    - 15.3 Determination of State Equivalence Using an Implication Table
    - 15.4 Equivalent Sequential Circuits
    - 15.5 Incompletely Specified State Tables

Elimination of Redundant States

- Row Matching
  - Recall CD player controller
    - Mealy implementation contained two sets of rows with same next state and output
    - Eliminate redundant states
- Row matching doesn't identify "equivalent states"
  - Row matching identifies "same state"
  - Equivalent states are the more general case

Equivalent States

- Definitions of equivalent states
  - *Roth : 2 states equivalent iff for every single input x, outputs are the same and next states are equivalent (as opposed to row matching)*
    - Pairwise comparison using implication table
  - *Kohavi : Iff for every possible input sequence the same output sequence will be produced regardless of whether S_i or S_j is the initial state*
    - Moore reduction procedure to find equivalence partition

Determination of State Equivalence using an Implication Table

- Find Equivalent Pairs

	NS		
PS	x=0	x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Determination of State Equivalence using an Implication Table

(1) Construct Implication Table for Pairwise Comparison

(2) First Pass

- Compare outputs
  - For states to be equivalent, next state and output must be the same
  - Put "X's" where outputs differ

Implication Table (first pass)

	A	B	C	D	E	F	G
B							
C	X	X					
D			X				
E	X	X		X			
F	X	X		X			
G			X		X	X	
H	X	X		X			X

PS	NS x=0	NS x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Determination of State Equivalence using an Implication Table

(3) One column (or row) at a time, find implied pairs

Implication Table (second pass)

	A	B	C	D	E	F	G
B	D-F C-H						
C	X	X					
D	A-D C-E	A-F E-H	X				
E	X	X	C-E A-D	X			
F	X	X	E-F B-D	X	C-F A-B		
G	B-D C-H	B-F H-H	X	A-B E-H	X	X	
H	X	X	C-E D-G	X	C-C A-G	C-F B-G	X

	NS		
PS	x=0	x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Determination of State Equivalence using an Implication Table

(3) One column (or row) at a time, find implied pairs (cont)

- Remove self implied pairs
  - A-D in cell A-D
  - C-E in cell C-E
- Remove same state pairs
  - H-H in cell B-G
  - C-C in cell H-E

Implication Table (second pass)

	A	B	C	D	E	F	G
B	D-F C-H						
C	X	X					
D	A-D C-E	A-F E-H	X				
E	X	X	C-E A-D	X			
F	X	X	E-F B-D	X	C-F A-B		
G	B-D C-H	B-F H-H	X	A-B E-H	X	X	
H	X	X	C-E D-G	X	C-C A-G	C-F B-G	X

Self-implied pairs

Same state pairs

Implication Table (second pass)

	A	B	C	D	E	F	G
B	D-F C-H						
C	X	X					
D	C-E	A-F E-H	X				
E	X	X	A-D	X			
F	X	X	E-F B-D	X	C-F A-B		
G	B-D C-H	B-F	X	A-B E-H	X	X	
H	X	X	C-E D-G	X	A-G	C-F B-G	X

Self-implied pairs

Same state pairs

Determination of State Equivalence using an Implication Table

(4) One column (or row) at a time, eliminate implied pairs

Implication Table (third pass)

B	~~D-F C-H~~ X						
C	X	X					
D	C-E	~~A-F E-H~~ X	X				
E	X	X	A-D	X			
F	X	X	~~E-F B-D~~ X	X	~~C-F A-B~~ X		
G	B-D C-H	~~B-F~~ X	X	~~A-B E-H~~ X	X	X	
H	X	X	C-E D-G	X	A-G	~~C-F B-G~~ X	X
	A	B	C	D	E	F	G

PS	NS x=0	NS x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Determination of State Equivalence using an Implication Table

(5) Next pass, one column (or row) at a time, eliminate implied pairs

(6) Continue until pass results in no further elimination of implied pairs

Implication Table (fourth pass)

B	~~D-F C-H~~						
C	X	X					
D	C-E	~~A-F E-H~~	X				
E	X	X	A-D	X			
F	X	X	~~E-F B-D~~	X	~~C-F A-B~~		
G	~~B-D C-H~~	~~B-F~~	X	~~A-B E-H~~	X	X	
H	X	X	~~C-E D-G~~	X	~~A-G~~	~~C-F B-G~~	X
	A	B	C	D	E	F	G

PS	NS x=0	NS x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Determination of State Equivalence using an Implication Table

(7) Combine equivalent states (based on coordinates of cells, not contents)

- A ≡ D, C ≡ E in example
  - Equivalence is pairwise
    - A ≡ B, B ≡ C implies A ≡ C (transitive)

(8) Construct reduced state table

Determination of State Equivalence using an Implication Table

- Reduced State Table
  - * indicates change from original state table

PS	NS x=0	NS x=1	z
A	A*	C	0
B	F	H	0
C	C*	A*	1
F	F	B	1
G	B	H	0
H	C	G	1

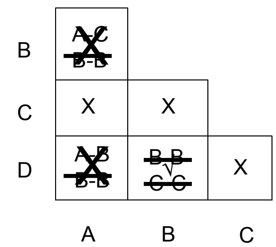
Determination of State Equivalence using an Implication Table

- Row Matching on an Implication Table
  - Mealy Machine outputs
    - Recall 101 sequence detector (direct Mealy conversion)

PS	NS,z x=0	NS,z x=1
A	A,0	B,0
B	C,0	B,0
C	A,0	D,1
D	C,0	B,0

Implication Table

- Same state pairs
- Eliminate implied pairs
- Matching rows
  - No implied pairs
  - B and D are "same state"



PS	NS,z x=0	NS,z x=1
A	A,0	B,0
B	C,0	B,0
C	A,0	D,1
D	C,0	B,0

Moore Reduction Procedure

- *States S_i and S_j of machine M are said to be equivalent If and only if, for every possible input sequence, the same output sequence will be produced regardless of whether S_i or S_j is the initial state*

Zvi Kohavi,
Switching and Finite Automata Theory

Moore Reduction Procedure

- *Two states, S_i and S_j, of machine M are <u>distinguishable</u> if and only if there exists at least one finite input sequence which, when applied to M, causes different output sequences depending on whether S_i or S_j is the initial state*
  - *The sequence which distinguishes these states is called a <u>distinguishing sequence of the pair (S_i,S_j)</u>*

Moore Reduction Procedure

- *If there exists for pair (S_i,S_j) a distinguishing sequence of length <u>k</u>, the states in (S_i,S_j) are said to be <u>k-distinguishable</u>*
  - *States that are not k-distinguishable are said to be <u>k-equivalent</u>*

Moore Reduction Procedure

- *The result sought is a partition of the states of M such that two states are in the same block if and only if they are equivalent*
  - *P_0 corresponds to 0-distinguishablity (includes all states of machine M)*
  - *P_1 is obtained simply by inspecting the table and placing those states having the same outputs, under all inputs, in the same block*
    - *P_1 establishes the sets of states which are 1-equivalent*

Moore Reduction Procedure

- *Obtain partition P_2*
  - *This step is carried out by splitting blocks of P_1, whenever their successors are not contained in a common block of P_1*
- *Iterate process of splitting blocks*
  - *If for some k, $P_{k+1} = P_k$, the process terminates and P_k defines the sets of equivalent states of the machine*
  - *P_k is thus called the <u>equivalence partition</u>*
    - *<u>The equivalence partition is unique</u>*

Moore Reduction Procedure

- Recall state table from earlier example

PS	NS x=0	NS x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

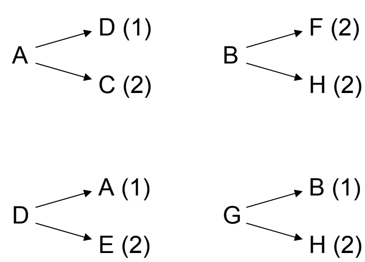
Moore Reduction Procedure

- P_0 = (ABCDEFGH)
- P_1 is obtained by splitting states having different outputs
  - P_1 =(ABDG)(CEFH)
    - Block 1 = ABDG, Block 2 = CEFH

PS	NS x=0	NS x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Moore Reduction Procedure

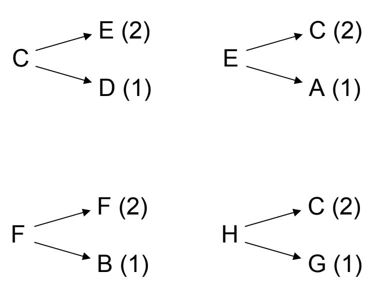
- Obtain P_2
  - Block 1 = ABDG, Block 2 = CEFH



	NS		
PS	x=0	x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Moore Reduction Procedure

- Obtain P_2 (cont)
  - Block 1 = ABDG, Block 2 = CEFH



	NS		
PS	x=0	x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

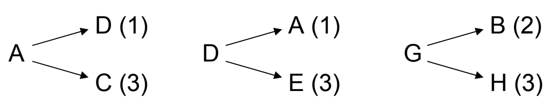
Moore Reduction Procedure

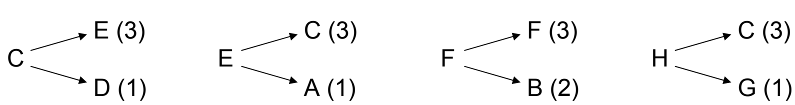
- Split B out of block 1
  - B is “2 distinguishable” from A, D and G
- No states of block 2 are “2 distinguishable”
- P_2 = (ADG)(B)(CEFH)
  - Block 1 = ADG
  - Block 2 = B
  - Block 3 = CEFH

Moore Reduction Procedure

- Obtain P_3
  - P_2 = (ADG)(B)(CEFH)

	NS		
PS	x=0	x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1





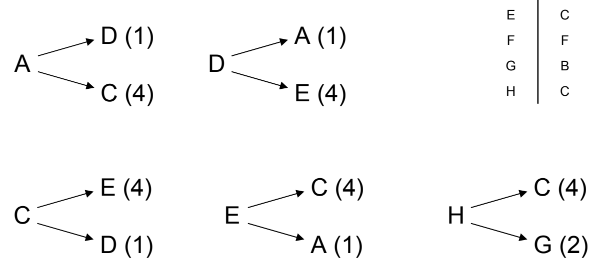
Moore Reduction Procedure

- Obtain P_3 (cont)
  - Split G from block 1
    - G is 3-distinguishable from A and D
  - Split F from block 3
    - F is 3-distinguishable from C, E and H
- P_3 = (AD)(G)(B)(CEH)(F)
  - Block 1 = AD, block 2 = G, block 3 = B, block 4 = CEH and block 5 = F

Moore Reduction Procedure

- Obtain P_4
  - P_3 = (AD)(G)(B)(CEH)(F)

PS	NS x=0	NS x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

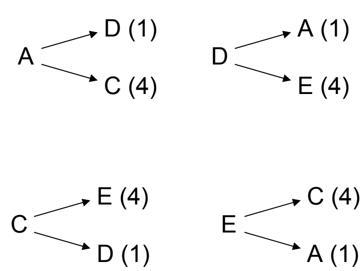


Moore Reduction Procedure

- Obtain P_4 (cont)
  - Split H from block 4
    - H is 4-distinguishable from C and E
- P_4 = (AD)(G)(B)(CE)(H)(F)
  - Block 1 = AD, block 2 = G, block 3 = B, block 4 = CEH, block 5 = H and block 6 = F

Moore Reduction Procedure

- Obtain P_5
  - P_4 = (AD)(G)(B)(CE)(H)(F)



PS	NS x=0	NS x=1	z
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

Moore Reduction Procedure

- Obtain P_5 (cont)
  - No blocks split from P_5
- $P_5 = P_4 = (AD)(G)(B)(CE)(H)(F)$
  - $P_5 = P_4$ = equivalence partition
  - Same result as implication table

Reduction of Incompletely Specified State Tables

- Use "modified row matching" to combine states

	NS		Z	
PS	x=0	x=1	x=0	x=1
A	-	B	-	-
B	C	D	-	-
C	A	-	0	-
D	A	-	1	-

A and C can be combined

A and D can be combined

C and D cannot (outputs differ)

Reduction of Incompletely Specified State Tables

- Using an Implication Table
  - State pairs are compatible, not equivalent
  - States must be “pairwise” compatible
    - ABC requires A-B, B-C and A-C
    - Compatible relationship is not transitive like equality
    - Compatible pairs must be grouped and included in reduced machine

Reduction of Incompletely Specified State Tables

- √ indicates “compatible pair”

A-C and A-D are compatible pairs
C-D are not compatible pairs

B	B-D		
C	√	A-C	
D	√	A-C	X
	A	B	C

A-B implies B-D; B-D implies A-C
→ requires ABCD grouping

B-C implies A-C; A-B implies B-D
→ requires ABCD grouping

B-D implies A-C
→ √

Reduction of Incompletely Specified State Tables

- Heuristic (non-deterministic) process
  - Requires "trial and error"
  - Not necessarily minimal

	NS		Z	
PS	x=0	x=1	x=0	x=1
AC	AC	BD	0	-
BD	AC	BD	1	-