UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Proiectarea Logică

24. Echivalenta starilor

STARILE ECHIVALENTE DIN CIRCUITELE SECVENTIALE

Realizarea unui circuit secvenţial care implementează o anumită funcţionare nu are întotdeauna soluţie unică. Este, adesea, posibilă construcţia unor circuite secvenţiale care să aibă acelaşi comportament funcţional dar, care să prezinte complexităţi distincte şi costuri diferite.

Se consideră, tradiţional, că dintre toate circuitele secvenţiale care implementează aceeaşi funcţie este mai simplu şi realizabil cu costuri reduse acela care are un număr mai mic de stări. Acest motiv dar şi altele similare au impulsionat căutarea reducerii stărilor unui automat dar păstrând aceiaşi funcţionalitate.

Atunci când se construieşte diagrama sau tabelul de stări pentru un automat finit, ori atunci când se descrie un automat finit printr-un cod HDL, se întâmplă adesea să se introducă stări redundante. Stările redundante sunt stări a căror funcţie poate fi îndeplinită de alte stări ale aceleiaşi maşini. În fapt, stările redundante sunt stări echivalente ca funcţie în automatul respectiv şi, din acest motiv, este suficientă una singură.

Criteriul după care se stabileşte faptul că în automatul M starea S_a este redundantă în raport cu starea S_b este transformarea

şir de valori de intrare $\rightarrow$ *şir de valori de ieşire*

definită de automatul finit M, atunci când acesta este iniţializat într-una dintre cele două stări S_a ori S_b.

Utilizând descrierea automatului, presupus complet definit, se poate stabili inductiv, pornind dintr-o stare arbitrară S_w şi utilizând un şir oarecare de valori posibile de intrare ($u_1\ u_2\ ...\ u_q$), şirul de valori de ieşire produs ($v_1\ v_2\ ...\ v_q$):

$$(S_w\ ,\ u_1\ u_2\ ...\ u_q) \rightarrow (v_1\ v_2\ ...\ v_q)$$

Această corespondenţă este definitorie pentru funcţionarea unui automat finit, indiferent de secvenţa de stări parcursă prin automatul respectiv. Corespondenţa este, tradiţional, notată prin:

$$\lambda(\ S_w\ ,\ u_1\ u_2\ ...\ u_q) = (v_1\ v_2\ ..., v_q),$$

unde λ este funcţia ieşirii unui automat finit, aplicată în cazul de faţă iterativ:

$$\lambda(\ S_w\ ,\ u_1\ u_2\ ...\ u_q) = \lambda(\ S_w\ ,\ u_1)\ \lambda(\delta(\ S_w,\ u_1),\ u_2)\ ...\lambda(\delta(\ \delta\ \ ...\ (\delta\ (\delta\ (S_w,\ u_1)\ u_2)\ )...),\ u_q).$$

În vederea micşorării numărului de stări ale unui automat se investighează acele stări care realizează aceiaşi funcţie iar reducerea acestora conduce la o maşină cu stări mai puţine dar care funcţionează echivalent celei iniţiale. Se spune că, în general, două maşini finite sunt echivalente dacă, atunci când sunt iniţializate corespunzător şi li se aplică aceleaşi şiruri de valori de intrare se produc, întotdeauna, aceleaşi şiruri de valori de ieşire.

Două stări S_a şi S_b ale unei maşini M sunt distinctibile dacă şi numai dacă există cel puţin un şir finit de valori de intrare, care atunci când aceasta se aplică maşinii M, produce două secvenţe, şiruri, de valori de ieşirie diferite, după cum starea iniţială a maşinii este S_a ori S_b.

Şirul de valori de intrare care distinge stările S_a şi S_b se numeşte secvenţă de distingere a perechii de stări respective şi atunci când aceasta are lungimea q se spune că perechea de stări S_a şi S_b sunt q-distinctibile. Sunt de interes, în general, secevnţele cele mai scurte de distingere (prima apariţie a unor valori diferite în şirul de valori de ieşire poate fixa lungimea minimă de distingere).

Conceptul de q-distinctibilitate conduce la definirea q-echivalenţei şi respectiv a echivalenţei a două stări S_i şi S_j ale aceluiaşi automat M. Stările care nu sunt q-distinctibile se spune că sunt q-echivalente. Se poate remarca o proprietate simplă a stărilor q-echivalente, şi anume că două stări q-echivalente sunt p-echivalente pentru orice $p < q$. Stările care sunt q-echivalente pentru orice valoare (naturală) q se spune că sunt echivalente. Altfel spus, pentru stările echivalente nu există nici o secevenţă care să le distingă. Dacă stările S_a şi S_b sunt echivalente atunci echivalenţa lor se notează prin $S_a = S_b$.
Definiţia stărilor echivalente urmează aceste considerente.

Definiţia 1:

Două stări S_a şi S_b ale maşinii M se spune că sunt echivalente dacă şi numai dacă oricare ar fi şirul de valori posibile de intrare aplicat se produce aceiaşi secvenţă de valori de ieşire indiferent că starea iniţială a maşinii M este S_a ori S_b.

Se poate remarca, fără mari dificultăţi, că echivalenţa stărilor dintr-o maşină finită, aşa cum a fost aceasta introdusă, este o *relaţie de echivalenţă*. Aceasta se demonstrează probând reflexivitatea, simetria şi tranzitivitatea relaţiei.

Propoziţia 1:

Partiţionarea mulţimii stărilor în clase de echivalenţă este unică.
Demonstraţie: Prin reducere la absurd, se consideră două partiţionări P_1 şi P_2 în clase de echivalenţă a mulţimii stărilor unui automat M, astfel încât $P_1 \neq P_2$. Fie stările S_a şi S_b astfel încât acestea sunt în aceiaşi clasă în prima partiţie şi sunt în clase diferite în cea de-a doua partiţie. Deoarece cele două stări sunt în clase diferite în cadrul celei de-a doua partiţii, există un şir de simboluri de intrare care le fac distinctibile – ceea ce contrazice faptul că ar putea fi echivalente.

O consecinţă firească a acestei proprietăţi este partiţionarea mulţimii stărilor unei maşini finite în clase de echivalenţă (disjuncte). Două sau mai multe stări aparţin aceleiaşi clase de echivalenţă dacă sunt echivalente.

Un automat poate fi privit ca având pentru fiecare stare internă o clasă de echivalenţă asociată acelei stări, astfel încât ori de câte ori automatul tranzitează spre una dintre stările unei clase de echivalenţă nu are importanţă care dintre stările echivalente ale respectivei clase este activată.
Faptul că o clasa de echivalenţă a unui automat finit conţine cel puţin o stare face ca automatul cu cele mai puţine stări să fie cel ale cărui clase de echivalenţe conţin, fiecare, doar *o singură stare*.

Reducerea numărului de stări interne ale unui automat M revine la înlocuirea automatului M prin automatul N, în care cel de-al doilea automat are în fiecare clasă de echivalenţă (a stărilor) o singură stare. Având în vedere definţia criteriului de echivalare al stărilor se poate introduce definiţia automatelor finite echivalente.

Definiţia 2:

Fie M şi N două automate finite complet determinate având aceleaşi valori, simboluri, de intrare din mulţimea U.
Fie $(u_1\ u_2\ ...,\ u_q)$, cu $u_i \in U,\ 1 \leq i \leq q$ un şir oarecare de valori de intrare, de lungime arbitrară. Atunci stările $S_x \in M$ şi $S_y \in N$ sunt echivalente dacă şi numai dacă:

$$\lambda_M(\ S_x\ ,\ u_1\ u_2\ ...\ u_q) = \lambda_N(\ S_y\ ,\ u_1\ u_2\ ...\ u_q).$$

Definiţia care urmează introduce într-un cadru mai general echivalenţa a două maşini finite M şi N.

Definiţia 3:

Două automate finite M şi N sunt echivalente dacă pentru fiecare stare S din M există o stare echivalentă T din N şi reciproc. Atunci se notează echivalenţa celor două automate prin $M = N$.

Exemplul 1:

Se consideră circuitul secvenţial *A* definit prin tabelul 1 şi circuitul secvenţial *B* definit prin tabelul 2. Aceste două circuite sunt echivalente. Acest fapt este demonstrat prin aplicarea directă a definiţiei aşa cum se poate urmări în tabelul 3. Astfel, după cum rezultă din tabelul 3, în urma aplicării secvenţelor de intrări 00110 circuitului *A* aflat în starea *p*1 sau în starea *p*3, se obţine aceiaşi secvenţă de valori la ieşiri. Acelaşi lucru se constată aplicând secvenţa de valori de intrare 1110. Aplicând aceste secvenţe de valori ale intrării sau oricare altele se obţine acelaşi răspuns al circuitului secvenţial *A* aflat iniţial în starea *p*1 sau *p*3, ca şi pentru circuitul secvenţial *B*, aflat iniţial în starea *q*1. Prin urmare $p_1 = p_3$, $p_1 = q_1$ şi $p_3 = q_1$. În mod similar se poate arăta că stările p_2 şi q_2 sunt echivalente. □

Starea actuală \ Intrări	0	1
p_1	$p_3/0$	$p_2/1$
p_2	$p_1/1$	$p_2/0$
p_3	$p_1/0$	$p_2/1$

Tabelul 1. Circuitul secvenţial *A*.

Starea actuală \ Intrări	0	1
q_1	$q_1/0$	$q_2/1$
q_2	$q_1/1$	$q_2/0$

Tabelul 2. Circuitul secvenţial *B*.

Numărul curent	Intrări	Automatul finit *A* Starea actuală	Starea viitoare	Ieşirea	Automatul finit *B* Starea actuală	Starea viitoare	Ieşirea
1	0	p_1	p_3	0	q_3	q_1	0
2	0	p_3	p_1	0	q_1	q_3	0
3	1	p_1	p_2	1	q_3	q_2	1
4	1	p_2	p_2	0	q_2	q_2	0
5	0	p_2	p_1	1	q_2	q_1	1
1	1	p_1	p_2	1	q_3	q_2	1
2	1	p_2	p_2	0	q_2	q_2	0
3	1	p_2	p_2	0	q_2	q_2	0
4	0	p_2	p_1	1	q_2	q_1	1
...	...	...	...	...	...	...	...

Tabelul 3. Găsirea stărilor echivalente în cele două circuite secvenţiale prin aplicarea directă a definiţiei.

Problema minimizării numărului de stări ale unui circuit secvenţial revine la stabilirea claselor de echivalenţă ale mulţimii stărilor acestuia. Aplicarea definiţiei, ca în multe alte cazuri similare, nu oferă soluţia cea mai eficientă.

Partiţionarea stărilor unui circuit secvenţial în clase de echivalenţă

Există mai multe metode de calcul al claselor de echivalenţă peste stările unui automat cu stări finite şi complet definit.
Toate metodele utilizează atât definiţia stărilor echivalente cât şi câteva condiţii *necesare* de realizare a echivalenţei stărilor. Condiţia necesară cel mai des utilizată se referă la totalitatea valorilor de ieşire (scalare sau vectoriale) ale unei stări. Această totalitate rezumă în fapt valorile ieşirii circuitului, într-o stare dată, pentru toate valorile posibile ale intrărilor. Daca totalitatea valorilor de ieşire pentru o stare p diferă de totalitatea valorilor de ieşire pentru starea q atunci stările p şi q nu sunt echivalente, notat $p \neq q$.

În fapt, dat un circuit secvenţial A, se poate estima de la bun început o margine inferioară a numărului claselor de echivalenţă. Anume, se formează totalitatea valorilor ieşirii pentru fiecare stare şi se calculează cuantumul situaţiilor distincte. Aceasta este marginea inferioară a numărului claselor de echivalenţă.

Astfel circuitul secvenţial din tabelul 4 are o margine inferioară a numărului de clase de echivalenţă egală cu patru:
$\{1, 5\}$, $\{2, 4, 8\}$, $\{3, 6, 9\}$, $\{7, 10, 11, 12\}$ pentru totalitatea ieşirilor (0, 0, 0, 0), (0, 1, 0, 0), (1, 0, 1, 1) şi respectiv (1, 1, 1, 1).

Oricare circuit secvenţial are şi o margine superioară a numărului de clase de echivalenţă. Această margine este egală cu numărul de stări ale circuitului. Dacă un circuit nu are echivalenţe printre stările sale atunci această margine va fi atinsă, fiecare stare fiind unică membră a câte unei clase de echivalenţă.

Algoritmul Paull-Unger

Acest algoritm utilizează un tabel în care se înscriu, pentru fiecare pereche de stări care ar putea fi echivalente, implicaţiile echivalenţei respective sau condiţiile care ar trebui îndeplinite pentru ca respectiva pereche de stări să fie echivalente.
Tabelul are numărul de linii şi de coloane egal cu numărul stărilor mai puţin o unitate. Pe rânduri sunt notate în ordine crescătoare, de sus în jos, toate stările exceptând prima stare. Pe coloane, la baza tabelului, sunt notate în ordine crescătoare, de la stânga la dreapta, toate stările exceptând ultima stare.

Din raţiuni de simetrie, tabelul este suficient să fie completat doar jumătate. Tradiţional se completează jumătatea stângă.

Intrări / Starea actuală	1	2	3	4
1	2/0	3/0	2/0	4/0
2	3/0	7/1	6/0	1/0
3	2/1	1/0	7/1	8/1
4	3/0	10/0	3/0	1/0
5	2/0	6/0	2/0	8/0
6	2/1	1/0	11/1	4/1
7	1/1	4/1	3/1	5/1
8	6/0	11/1	6/0	5/0
9	2/1	5/0	11/1	4/1
10	1/1	2/1	6/1	5/1
11	1/1	2/1	6/1	5/1
12	1/1	3/1	2/1	1/1

Tabelul 4. Circuitul secvenţial al exemplului 2.

În elementul *i,j* al tabelului, corespunzător intersecţiei liniei stării *i* şi coloanei stării *j,* sunt înscrise condiţiile pentru ca respectivele stări să fie echivalente sau sunt notate orice alte remarci privitor la echivalenţa acestor stări.

În cadrul utilizării elementelor tabelului sunt utilizate trei reguli.

Prima regulă stabileşte că toate perechile de stări care au totalitatea ieşirilor diferită sunt stări incompatibile şi se barează celula tabelului corespunzătoare acestora. Perechile de stări pentru care coincide totalitatea ieşirilor pot fi echivalente dacă sunt satisfăcute condiţiile de echivalenţă ale stărilor corespunzătoare intrărilor specifice.

A doua regulă este iterativă şi aplică un procedeu de omogenizare a condiţiilor. Astfel toate condiţiile de incompatibilitate (corespunzătoare celulelor barate) sunt verificate în listele de condiţii de echivalenţă ale celulelor ne-barate. Prezenţa unei incompatibilităţi într-o condiţiile de echivalenţă a unei celule ne-barate conduce la bararea acesteia şi la trecerea perechi de stări corespunzătoare în categoria perechilor de stări incompatibile. La pasul iterativ următor această pereche va fi verificată ca prezenţă în toate celulele ne-barate dar cu condiţii înscrise ş.a.m.d.
Atunci când după o parcurgere completă a tabelului nu mai sunt descoperite noi

perechi de stări incompatibile se trece la aplicarea celei de-a treia reguli.
A treia regulă, şi ultima, extrage din tabel toate condiţiile de compatibilitate şi utilizând tranzitivitatea stabileşte clasele de echivalenţă.

2											
3											
4		7,10 3,6									
5	3,6 4,8										
6			7,11 4,8								
7											
8		3,6 7,11 1,5		3,6 10,11 1,5							
9			1,5 7,11 4,8			1,5					
10							2,4 3,6				
11							2,4 3,6			√	
12							3,4 3,2 1,5			2,3 2,6 1,5	2,3 2,6 1,5
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**

Tabelul 5. Aplicarea algoritmului Paull-Unger circuitului secvenţial din Exemplul 2.

Pentru simplitatea expunerii şi uşurinţa înţelegerii aplicării algoritmului se va apela la urmărirea exemplului 2.

Exemplul 2:

Se consideră circuitul secvenţial descris prin tabelul 4.
Construcţia, în conformitate cu algoritmul Paull-Unger, a tabelului de găsire a stărilor echivalente este prezentată în tabelul 5.

Prima regulă a algoritmului vizează completarea iniţială a elementelor tabelului.

Toate elementele (căsuţele) tabelului care sunt barate reprezintă perechi de stări ne-echivalente sau, altfel spus, incompatibile. Astfel, starea 1 şi starea 2 din tabelul 4 nu sunt echivalente deoarece ieşirea lor pentru intrarea 2 diferă.

Există, pe de-altă parte stări evident echivalente. În tabelul 4 se remarcă, fără nici un dubiu, echivalenţa stărilor 10 şi 11 (celula respectivă este marcată cu caracterul √).

Acolo unde totalitatea ieşirilor coincide se pot formula condiţiile de echivalenţă. Astfel, pentru ca starea 11 să fie echivalentă stării 12 mai trebuie ca 2 = 3, 2 = 6 şi 1 = 5. Aceste condiţii sunt înscrise sub forma unor perechi (perechea 2,3 spre exemplu), în elementul tabelului corespunzător liniei 12 şi coloanei 11.

A doua regulă a algoritmului propagă toate incompatibilităţile celulelor barate din tabel şi în celulele ne-barate dar care utilizează condiţii explicite de echivalenţă.

Atunci când o celulă, ne-barată, conţine o condiţie de echivalenţă contrazisă de o incompatibilitate, aceasta se barează devenind incompatibilitate la rândul său şi putând propaga alte incompatibilităţi printre echivalenţele condiţionate.

Această a doua regulă se recomandă să se aplice ordonat în tabel (ca parcurgere) şi iterativ.

Astfel, dacă se parcurge tabelul 5 pornind din colţul dreapta jos, prima incompatibilitate corespunde perechii (9,12) dar, care nu apare menţionată ca şi condiţie de echivalenţă în nici un element al tabelei.

Nu acelaşi lucru se întâmplă cu perechea incompatibilă (3,4) care apare ca şi condiţie de echivalenţă a stărilor 7 şi 12. În consecinţă se barează celula (7,12), care avea dealtfel şi condiţia incompatibilă (2,3). Deoarece acest mod de barare este ulterior celui iniţial, celula (12, 7) în tabelul 5, pentru mai multă claritate a fost barată printr-o elipsă desenată punctat şi înscrisă în celulă.

Incompatibilitatea stărilor 2 şi 3 barează atât perechea (10,12) cât şi perechea (11,12) (aveau şi (2,6) a doua pereche incompatibilă dealtfel). Ambele celule din tabel sunt barate similar celulei (12,7) din tabelul 5.

În continuare nu mai apar incompatibilităţi şi se încheie de aplicat regula a doua. Tabelul conţine acum informaţii despre toate perechile de stări echivalente.

Aplicând cea de-a treia regulă se extrag perechile care s-au păstrat:

10 = 1, 7 = 11, 1 = 10, 6 = 9, 3 = 9, 4 = 8, 2 = 8, 3 = 6, 1 = 5 şi 2 = 4.

Se grupează aceste perechi în clase de echivalenţe utilizând tranzitivitatea relaţiei:

{10, 11, 7}, {6, 9, 3}, {4, 8, 2} şi {1, 5}. La aceste clase de echivalenţă formate din două sau mai multe stări se adaugă ultima clasă de echivalenţă a acestui circuit secvenţial conţinând o singură stare {12}. S-au obţinut, în total, 5 clase de echivalenţă ale circuitului descris prin tabelul 4, cu una mai mult decât marginea inferioară estimată iniţial.

□

În cazul circuitelor secvenţiale complet definite, partiţia mulţimii stărilor obţinută prin relaţia de echivalenţă a stărilor reprezintă unica posibilitate de obţinere a unui număr minim de grupe de stări echivalente, deoarece fiecare stare este inclusă într-o unică clasă (de echivalenţă).

Această partiţie are proprietatea că fiecare clasă de stări poate fi reprezentată printr-o singură stare din clasa respectivă. Prin urmare este suficient ca fiecare clasă să fie reprezentată printr-o unică stare în circuitul secvenţial minim.