Implikacija v tQCA

Gregor Jeraj, Urša Levičnik, Zahir Mujanović, Sašo Skube

13. januar 2009

1 Uvod

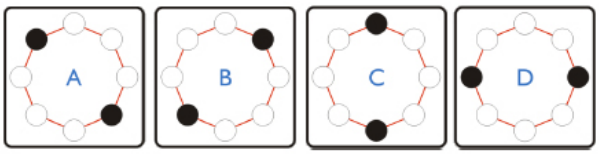
Tema druge seminarske naloge je realizacija trojiške implikacije v tQCA.

1.1 Trojiški sistem

Hayes v svojem članku obravnava trojiški sistem kot alternativo dvojiškemu, ki ga uporabljajo stroji in desetiškemu, ki ga ljudje uporabljamo za seštevanje. Ugotovi, da je najbolj kompaktni tisti zapis, ki bi imel za osnovo število e (osnovo naravnega logaritma), ki znaša 2.718. Ker pa to število ni celo, predlaga zaokorožitev na 3. Torej je trojiški sistem (sistem z bazo oz. radiksom 3) tisti, ki je nabolj kompakten.

1.2 tQCA

Trojiški sistem je osnova za spremembo Lentove celice. Do sedaj je bila to celica s štirimi pikami in dvema možnima stanjema, sedaj pa obravnavamo celico, ki ima 8 pik in 4 možna stanja: 0 in 1, ki ju že poznamo, ter dve novi stanji. Stanja se sedaj označujejo s črkami: A, B, C in D.



Slika 1: 4 stanja tQCA celice

Stanja A, B in C smo definirali tako:

$$A = 0$$

$$B = 1$$

$$C = \frac{1}{2}$$

Vrednosti D nismo definirali, ker je ne potrebujemo, tu je navedena le ker obstaja kot možno stanje v tQCA. Na vhodu in izhodu je to neželena vrednost.

1.3 Trojiška implikacija

Problem trojiške logike je, da logičnih funkcij ne moremo obravnavati na dvojiški način. Čeprav to hočemo pa imamo problem s tretjo (vmesno) vrednostjo, ki se ne obnaša tako kot hočemo. Na to temo je pisalo že veliko avtorjev, zato je tudi idej kakšni naj bi bili primitivi v trojiški logiki veliko.

Primitivi v trojiški logiki:

	Łukasiewicz	Bochvar	Kleene	Heyting	Reichenbach
a b	$\wedge\ \vee\ \Rightarrow\ \Leftrightarrow$	$\wedge\ \vee\ \Rightarrow\ \Leftrightarrow$	$\wedge\ \vee\ \Rightarrow\ \Leftrightarrow$	$\wedge\ \vee\ \Rightarrow\ \Leftrightarrow$	$\wedge\ \vee\ \Rightarrow\ \Leftrightarrow$
0 0	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
0 $\frac{1}{2}$	0 $\frac{1}{2}$ 1 $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	0 $\frac{1}{2}$ 1 $\frac{1}{2}$	0 $\frac{1}{2}$ 1 0	0 $\frac{1}{2}$ 1 $\frac{1}{2}$
0 1	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0	0 1 1 0
$\frac{1}{2}$ 0	0 $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	0 $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	0 $\frac{1}{2}$ 0 0	0 $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$
$\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ 1 1	$\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ 1 1	$\frac{1}{2}$ $\frac{1}{2}$ 1 1
$\frac{1}{2}$ 1	$\frac{1}{2}$ 1 1 $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ 1 1 $\frac{1}{2}$	$\frac{1}{2}$ 1 1 $\frac{1}{2}$	$\frac{1}{2}$ 1 1 $\frac{1}{2}$
1 0	0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0
1 $\frac{1}{2}$	$\frac{1}{2}$ 1 $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ 1 $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ 1 $\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{2}$ 1 $\frac{1}{2}$ $\frac{1}{2}$
1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

Za potrebe te seminarske naloge smo upoštevali Łukasiewiczevo tabelo.

2 Realizacija

2.1 Simulator

Naša prva ideja je bila preiskati polje celic velikosti $N \times N$. Postopali smo na tak način, da smo napisali skripto, ki v tem polju generira neko povezano celično strukturo. To strukturo se nato da kot vhodni parameter tQCA simulatorju.

S simulatorjem smo si bolj malo pomagali. Ugotovili smo, da čim je polje $N \times N$ preveliko ($N \geq 4$) je število možnih kombinacij preveliko za zmogljivost naših procesorjev.

Simulator je sestavljen iz naslednjih delov:

- En del je za izdelavo kombinacij,
- en del za preverjanje povezanosti celic,
- en del za izdelavo simulacijskih datotek in zaganj,
- del za parsanje podatkov iz rezultatov
- in del ki primerja rezultate z funkcijami.

2.2 Drugačen pristop

Nato smo se lotili problema na drugačen način. Poskusili smo z logičnim pristopom iskanja Łukasiewiczeve implikacije. Odločili smo se, da tudi za druge logične operacije uporabimo Łukasiewiczev nabor. Izhajali smo iz dvojiške logike, pri kateri je $A1 \Rightarrow A2 = \neg A1 \vee A2$.

A1 A2	$\neg A1 \vee A2$	$A1 \Rightarrow A2$
0 0	1	1
0 1	1	1
1 0	0	0
1 1	1	1

Če enako logiko uporabimo v trojiškem naboru dobimo naslednje rezultate:

A1	A2	$\neg A1 \vee A2$	$A1 \Rightarrow A2$
0	0	1	1
0	$\frac{1}{2}$	1	1
0	1	1	1
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1
$\frac{1}{2}$	1	1	1
1	0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
1	1	1	1

Opazimo, da za vse kombinacije vhodnih podatkov dobimo ustrezen izhod pri Łukasiewiczevi trojiški logiki, moti nas le vhodni vektor $\frac{1}{2}, \frac{1}{2}$, pri katerem je rezultat $\frac{1}{2}$, želimo pa si rezultat 1, saj to zadostuje pravilom Łukasiewiczeve implikacije. Potrebujemo torej način kako iz $\frac{1}{2}$ dobiti vrednost 1. Pravilni rezultat bi dobili, če bi naredili *or* med funkcijama $\neg A1 \vee A2$ in funkcijo $F1$, ki je definirana tako, kot je predstavljeno v naslednji tabeli:

A1	A2	F1
0	0	0
0	$\frac{1}{2}$	0
0	1	0
$\frac{1}{2}$	0	0
$\frac{1}{2}$	$\frac{1}{2}$	1
$\frac{1}{2}$	1	0
1	0	0
1	$\frac{1}{2}$	0
1	1	0

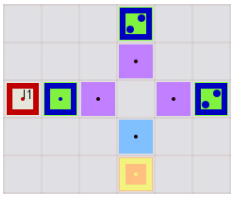
Zato moramo najprej iz funkcije izolirati vrednost $\frac{1}{2}$ in jo preslikati v funkcijo podobno F1. Vhodni vrednosti $\frac{1}{2}$ lahko izoliramo s funkcijo F2:

A1	A2	$F2 = (A1 \vee A2) \vee (\neg A1 \vee \neg A2)$
0	0	1
0	$\frac{1}{2}$	1
0	1	1
$\frac{1}{2}$	0	1
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	1	1
1	0	1
1	$\frac{1}{2}$	1
1	1	1

Potrebujemo le še funkcijo ki bo preslikala vrednosti funkcije F2 v F1. Funkcijo označimo z $\bowtie$ in je take oblike:

x	$\bowtie(x)$
0	1
$\frac{1}{2}$	1
1	0

Pri iskanju funkcije smo si pomagali z našim simulatorjem. S pomočjo programa smo v prostoru 3×3 našli funkcijo, ki ustreza našim zahtevam (slika 2.



Slika 2: Funkcija

Po simulaciji dobimo tak rezultat:

$$A == B(0.998585)$$
$$B == A(0.998585)$$
$$C == B(0.998585)$$
$$D == C(0.998687)$$

D-ja ne potrebujemo, zato ga ignoriramo. Vidimo, da se kritična vrednost C spremeni v B. To je točno to kar potrebujemo.

Če še enkrat obnovimo celoten postopek je naša implikacija zgrajena takole:

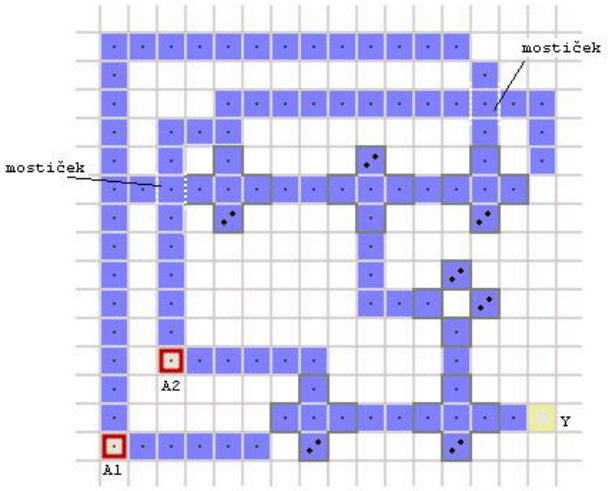
$$F_{implikacija} = (\neg A1 \vee A2) \vee \bowtie (F2(A1, A2))$$
$$F_{implikacija} = (\neg A1 \vee A2) \vee \bowtie ((A1 \vee A2) \vee (\neg A1 \vee \neg A2)$$

Na tak način lahko s funkcijami *and*, *or*, *not* izrazimo Łukasiewiczevo implikacijo.

A1	A2	$\neg A1 \vee A2$	F1	F2	$\bowtie (F2)$	$(\neg A1 \vee A2) \vee \bowtie (F2(A1, A2))$	$A1 \Rightarrow A2$
0	0	1	0	1	0	1	1
0	$\frac{1}{2}$	1	0	1	0	1	1
0	1	1	0	1	0	1	1
$\frac{1}{2}$	0	$\frac{1}{2}$	0	1	0	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	1
$\frac{1}{2}$	1	1	0	1	0	1	1
1	0	0	0	1	0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	$\frac{1}{2}$	$\frac{1}{2}$
1	1	1	0	1	0	1	1

Iz navedene tabele in enačb bi tako lahko sledila naslednja logicna tQCA struktura. V praksi zadeve nismo mogli preizkusiti, ker je struktura prevelika - vsebuje preveč celic. Pri takšni realizaciji ne bi mogli biti vsi vhodi na eni

in vsi izhodi na drugi strani, ampak bi moral biti vsaj en vhod v notranjosti strukture. Druga rešitev je uporaba večnivojske arhitekture, kot smo nakazali na naši skici (3).



Slika 3: Ideja celotnega vezja

3 Zaključek

Naša ideja o izdelavi simulatorja ki bi sam iskal funkcije je bila v teoriji dobra, v praksi, pa se ni preveč izkazala. Na koncu smo morali še vedno iskati funkcijo na roke.

Ker je implikacija precej zahtevna funkcija, nam kljub veliko truda funkcije ni uspelo najti. Čim smo si probali olajšati nalogo smo uporabili preveč celic in simulacija je bila prezahtevna. Če smo hoteli zmanjšati število celic, funkcija ni delovala kot smo hoteli. Na koncu smo v poročilo vključili našo idejno funkcijo, katere največji problem je, da potrebuje dvoplastnost.

4 Literatura

Hayes, Third Base, American Scientist 2001
Magdevski Z., Realizacija omejenega nabora trovrednostnih logičnih funkcij na osnovi struktur kvantnih celičnih avtomatov, Magistrska naloga, 2006
Mraz M., Prosojnice s predavanj predmeta ONT
Pečar P., Prosojnice iz vaj predmeta ONT