Minimizarea funcțiilor logice

realizat de Florin Sabău

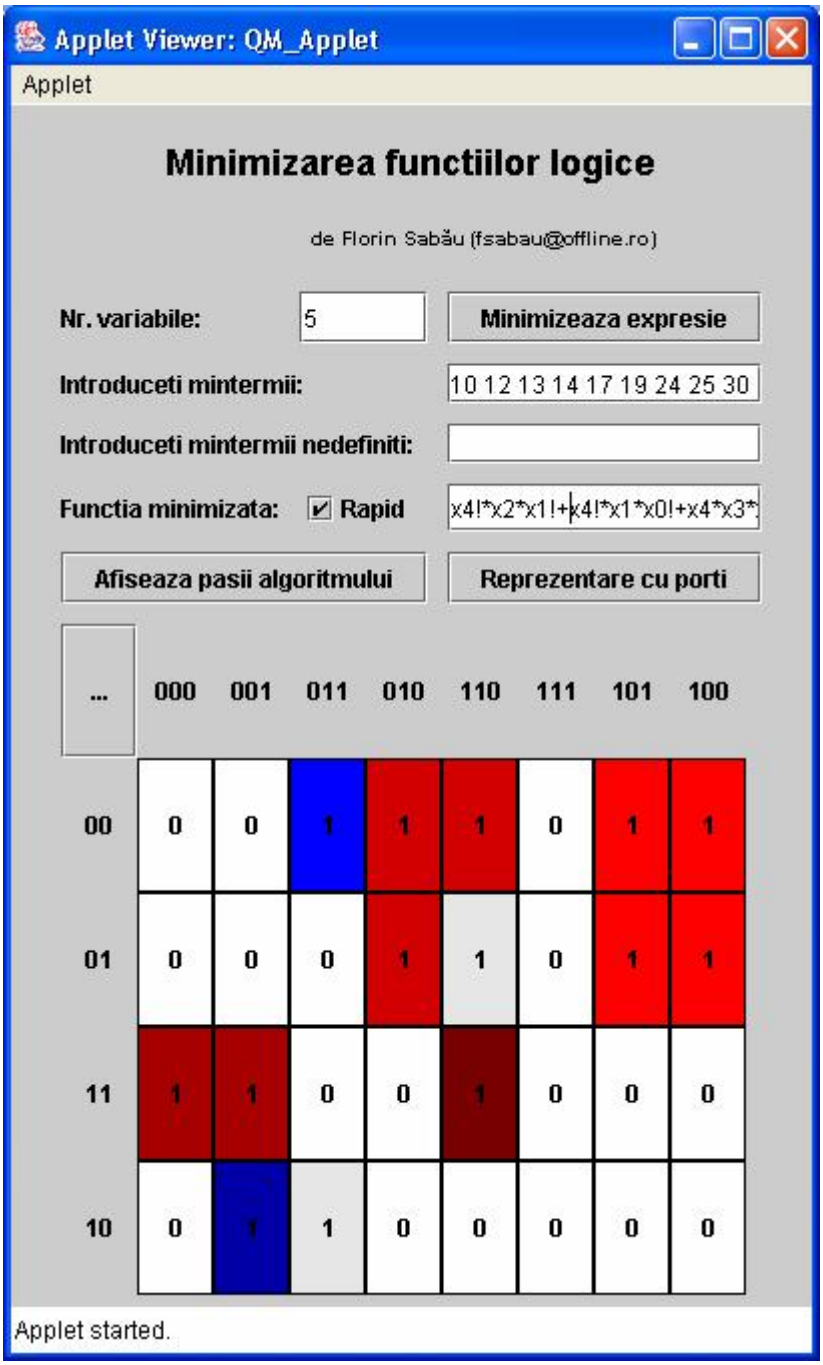
anul II, grupa 3-2

Cuprins

Introducere

Proiectul de faţă realizează minimizarea funcţiilor logice utilizând algoritmul Quinne-McCluskey. Este format dintr-un applet/aplicaţie Java ce poate rula atât din browser inclus într-o pagină web, cât şi ca aplicaţie de sine stătătoare. Pentru detalii privind implementarea algoritmului a fost utilizat articolul „Minimizarea funcţiilor logice" (autor Vasile Airinei) publicat în revista GInfo 04/2002 (articol acesibil şi pe Internet de pe site-ul www.ginfo.ro). Applet-ul poate rula pe configuraţii de Java Runtime de la 1.3 în sus!

Interfața applet-ului



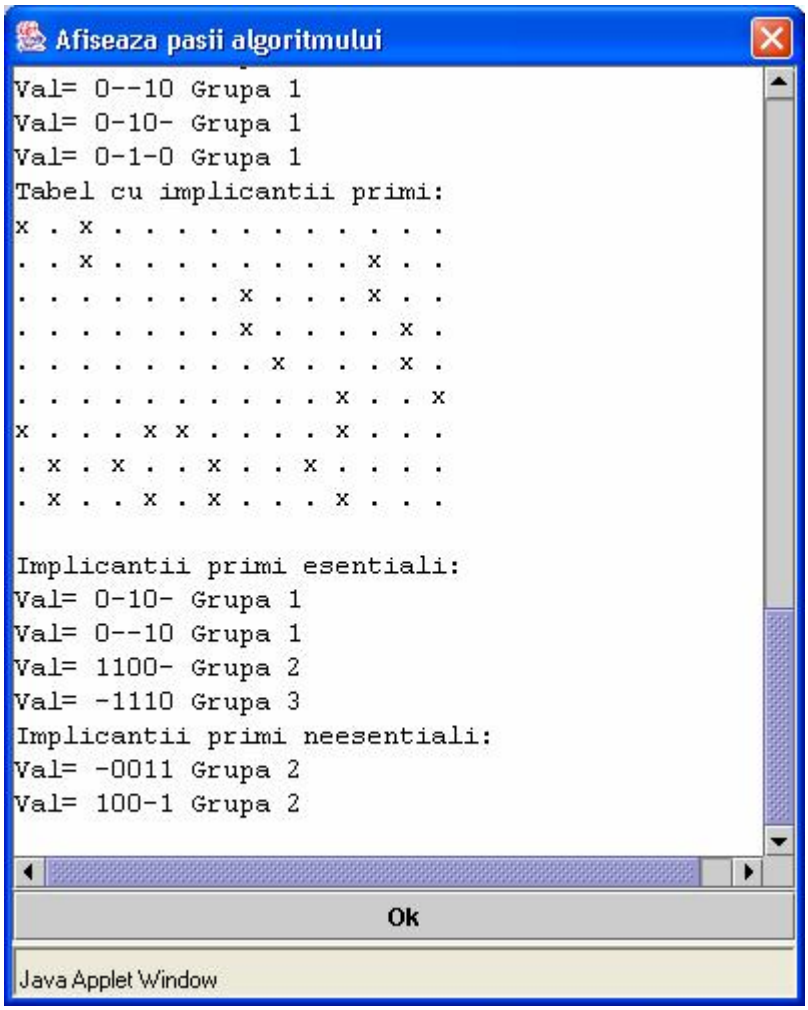
După cum se poate vedea şi din poza alăturată, interfaţa applet-ului se vrea una intuitivă:

- **în partea de sus a ferestrei există o casetă de text în care se poate introduce numărul de variabile al funcţiei logice;**
- **un buton ce declanşează minimizarea propriu-zisă („Minimizează expresie");**
- **2 casete de text în care se pot introduce mintermii funcţiei logice respectiv termenii nedefiniţi – acest**

lucru se poate realiza şi utilizând mouse-ul prin selectarea căsuţelor asociate mintermilor pe grila din parte de jos a ferestrei (diagrama Veitch-Karnaugh);

♦ o casetă de text în care se afişează rezultatul, funcţia minimă dată sub forma normal-disjunctivă (unde x! înseamnă x negat);

♦ un buton care afişează paşii intermediari prin care algoritmul trece (cu rezultatele parţiale) – poate fi folosită în scopuri de debugging (vezi fereastra alăturată);



♦ un buton („Reprezentare cu porţi") care deschide o nouă fereastră în care rezultatul minimizării este afişat prin intermediul unei reprezentări grafice cu porţi logice (ŞI-NU, ŞI-SAU-NU, DEC, MUX); vezi pagina următoare pentru un exemplu de reprezentare cu porţi ŞI-NU;

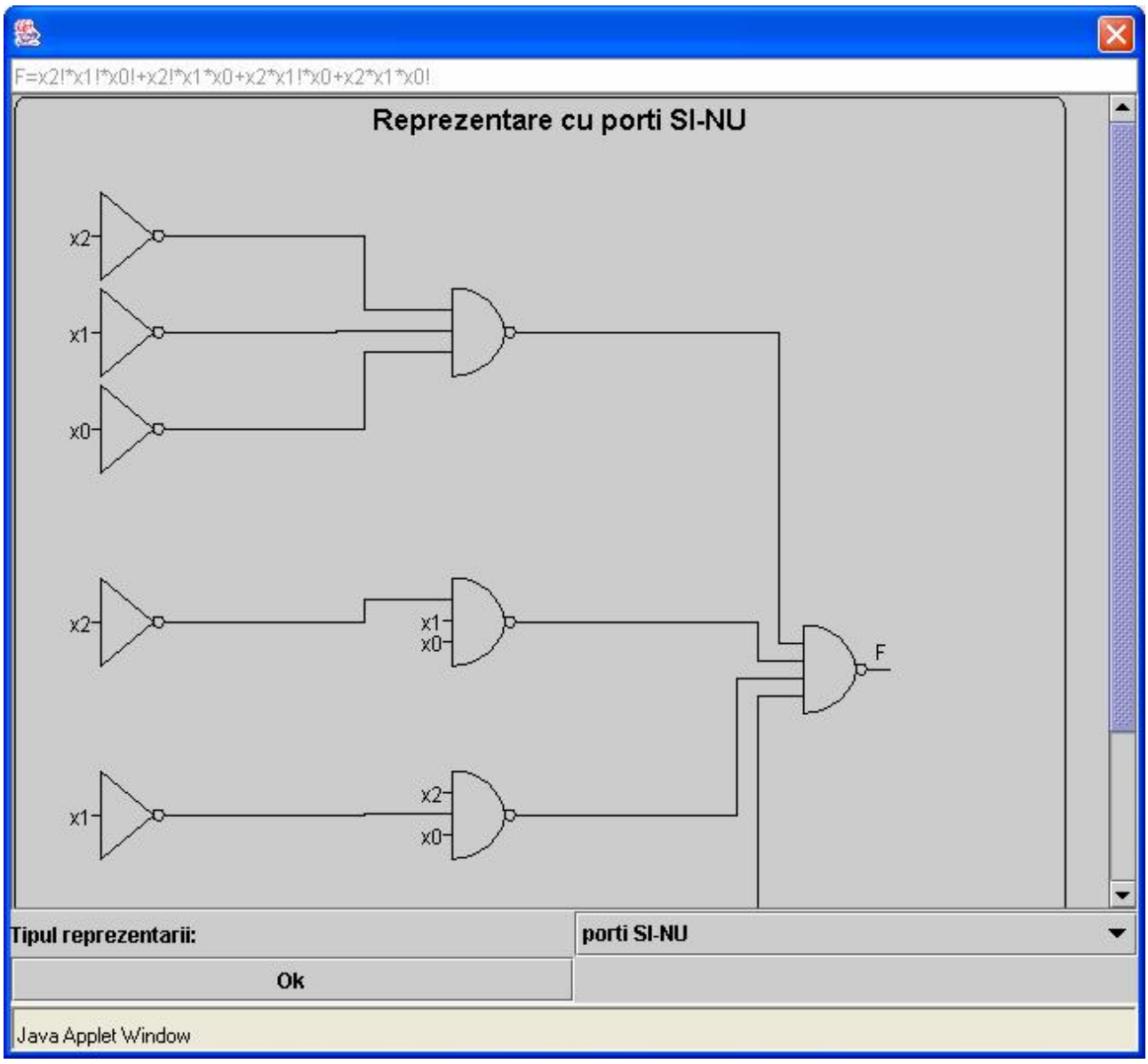
♦ butonul „Reset" iniţializează grila la „0" (prin click dreapta pe acest buton se realizează inversarea grilei – adică 1 → 0 şi 0 → 1);

♦ checkbox-ul „Rapid" selectează algoritmul de lucru – în unele cazuri (în care trebuie aleşi un număr mare de implicanţi neesenţiali), diferenţa de viteză între cei doi algoritmi este notabilă;

♦ grila (diagrama Veitch-Karnaugh) care într-o formă grafică afişează rezultatele minimizării astfel:

Ø celulele colorate în nuanţe de roşu reprezintă un implicant prim esenţial;

- **celulele colorate în nuanţe de albastru reprezintă un implicant prim neesenţial;**
- **celulele ce se află la intersecţia a mai multor implicanţi sunt colorate în gri.**
- **<u>Observaţie:</u> Prin trecerea cursorului mouse-ului deasupra celulelor grilei, se realizează selectarea grupării (implicant) din care face partea celula curentă, deasupra căreia se află cursorul (se colorează cu un azuriu deschis). De asemenea se selectează în cadrul casetei de text ce conţine rezultatul, a grupării (implicantului) curent. Dacă sub cursor se află o celulă care face parte din mai multe grupări (deci este colorată într-un gri deschis), atunci prin apăsarea butonului dreapta al mouse-ului, se pot selecta, pe rând, toate grupările din care face parte acea celulă.**



Observaţie: **Momentan singurul mecanism de realizare a persistenţei datelor de intrare (mintermi) se poate realiza prin folosirea clipboard-ului, şi salvarea acestor mintermi pentru o utilizare ulterioară în alte programe care rulează în sistem. Totuşi, atunci când este utilizat sub formă de applet, datorită restricţiilor de securitate care interzic accesul la clipboard-ul din sistem, applet-ul trebuie să fie semnat (signed), iar la încărcarea sa într-un browser, utilizatorul trebuie să fie de acord cu instalarea unui certificat de autenticitate (pentru a putea beneficia de clipboard-ul sistem şi nu de unul local applet-ului). Dacă nu este instalat acest certificat, applet-ul va rula normal, doar că va folosi un clipboard local pentru operaţiile de Copy/Cut/Paste şi nu va putea fi folosită această metodă pentru a copia date în alte programe.**

Atenţie: Dacă alegeţi să introduceţi mintermii direct în casetele de text, veţi observa că grila nu se sincronizează cu ceea ce introduceţi. Acest lucru (sincronizarea) va avea loc când apăsaţi pe butonul „Minimizează expresie”.

Detalii „tehnice”

Pentru scrierea acestui applet am folosit Borland JBuilder 6 versiunea Personal şi EditPlus 2.11. Applet-ul a fost testat sub Internet Explorer 6 şi Opera 7, fiind instalat suportul Java Runtime Environment 1.4.1.

Sursa applet-ului este formată din 5 fişiere care implementează în fapt 5 clase publice:

- **QM.java – conţine algoritmul Quinne-McCluskey propriu-zis;**
- **QM_Applet.java – conţine codul sursă al applet-ului;**
- **MyButtons.java – conţine clasa MyButtons ce implementează butoanele din cadrul grilei (folosită de QM_Applet);**
- **MyDialog.java – implementează caseta de dialog ce afişează paşii algoritmului (cu rezultatele parţiale);**
- **GraphicDialog.java – implementează fereastra ce afişează reprezentarea grafică cu porţi logice a rezultatului.**

Urmează o descriere a celor mai importante funcţii din fişierele sursă:

QM.java

- **QM (int variabile, String intrare, String nedefinit) – constructor, are ca parametrii numărul de variabile (variabile), mintermii (intrare) şi termenii nedefiniţi (nedefinit);**
- **void findIP() – calculează implicanţii primi;**
- **void improve() – calculează implicanţii primi esenţiali şi cei neesenţiali;**
- **Lista getIP(); Lista getIPE(); Lista getIPN() – returnează un obiect de tip Lista (definit în QM.java) care conţine impl. primi, impl. primi esenţiali respectiv impl. primi neesenţiali.**

QM_Applet.java

- **void init() – iniţializează applet-ul cu elementele de interfaţă (JButton, JTextField etc.);**
- **void populateGrid() – în funcţie de nr. de variabile selectat şi de mintermii introduşi generează grila;**
- **void resetButtons() – iniţializează pe „0” celulele grilei;**
- **void drawResults(Lista IPE, Lista IPN) – colorează grila în funcţie de rezultatele minimizării.**