

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI
FACULTATEA DE INFORMATICĂ

TITLU PROVIZORIU

ANDREI-ALIN CORODESCU

Sesiunea: *iulie, 2018*

Coordonator științific
Conf. Dr. Ciobâcă Ștefan

Cuprins

1	Introducere	2
2	Contribuții	3
3	Fundamente Teoretice	3
3.1	Limbaaj utilizat	3
3.1.1	Model de stocare	3
3.1.2	Sintaxa și semantica limbajului	3
3.2	Logica Hoare	3
3.3	Logica Separatoare	3
3.4	Logica Separatoare Relațională	3
3.5	Logica Rescrierii	3
4	Prover Logică Separatoare Relațională	3
4.1	Prezentare Generală Maude	4
4.2	Semantică executabilă a limbajului folosit	4
4.3	Modelarea Logicilor Separatoare	4
4.4	Interacțiunea prin LOOP-MODE	4
4.5	Schemă Generală de funcționare	4
4.6	Procese automatizate	4
4.6.1	Potrivire automată a axiomelor și a goal-urilor dovedite	4
4.6.2	Demonstrarea automată a implicațiilor	4
4.7	Interfața	4
4.8	Viitor	4
5	Cercetări adiționale	4
5.1	Automatizarea demonstrațiilor	4
5.2	Extensii pentru programe concurente	4
5.3	Java+ITP	4
6	Concluzii	4

1 Introducere

Compararea programelor sau a fragmentelor de cod și studierea echivalenței acestora este parte din activitatea fiecărui programator atunci când testează o implementare alternativă la ceva existent, repară erori din codul sursă, lansează versiuni noi ale unui produs, etc. Natural, pentru orice proces realizat de oameni se încearcă să se găsească soluții care să eficientizeze, să reducă potențiale erori și, în final, să automatizeze complet procesul. Odată ce procesul poate fi automatizat, acesta poate fi inclus în etapele realizării unui program. Cel mai relevant astfel de exemplu ar fi optimizarea codului realizată de compilator: în astfel de situație, codul optimizat trebuie să fie echivalent cu varianta originală.

Prezenta lucrare descrie dezvoltarea unui utilitar de verificare a modului cum sunt relaționate două programe, utilizând fundamente teoretice deja bine studiate și folosite, și tehnologii ce facilitează implementarea respectivelor concepte.

Utilitarul se bazează pe Logica Hoare împreună cu extensiile sale, Logica Separatoare (numită în continuare Logica Separatoare) și Logica Separatoare Relațională (numită în continuare Logica Relațională), ce ușurează procesul de argumentare asupra programelor, implementate folosind o abordare bazată pe Logica Rescrierii în limbajul Maude.

Acesta reprezintă un program de tip CLI în care se pot demonstra în mod interactiv relații între programe și, consecință a dependenței logicii relaționale de cea separatoare, proprietăți ale programelor (fără a fi relaționate cu alt program). Dezvoltarea utilitarului s-a realizat cu ideea de a putea fi extins ulterior, principalele extensii fiind suportarea modelării programelor concurente și automatizarea demonstrațiilor.

Restul lucrării este structurat după cum urmează :

- Capitolul 1 va descrie limbajul utilizat pentru a descrie programele, împreună cu Logicile utilizate pentru a realiza raționamente peste programele descrise de limbaj
- Capitolul 2 va descrie în mod detaliat modul de construire și folosire a utilitarului, cu accent pe funcționalitățile oferite de limbajul Maude.
- Capitolul 3 va prezenta câteva din cercetările conexe cu tema lucrării și cum acestea ar putea fi integrate în aceasta

- Capitolul 4 va prezenta concluziile lucrării, atât pentru aspectele teoretice utilizate (Logicile prin care se realizează raționamente), cât și pentru aspectele ce țin de implementare (limbajul Maude).

2 Contribuții

Contribuțiile aduse de absolvent:

- Modelarea logicilor relaționale și separatoare prin intermediul limbajului Maude
- Modelarea unui limbaj simplu de programare folosind Maude
- Crearea unui utilitar interactiv de demonstrare a relațiilor dintre programe sau a proprietăților unor programe
- Integrarea acestuia cu o soluție deja existentă pentru a rezolva o sub-problemă (dacă reușesc să mai fac integrarea cu Cyclist)

3 Fundamente Teoretice

În acest capitol voi prezenta pe scurt fundamentele teoretice ale prezentei lucrări.

3.1 Limbaj utilizat

3.1.1 Model de stocare

3.1.2 Sintaxa și semantica limbajului

3.2 Logica Hoare

3.3 Logica Separatoare

3.4 Logica Separatoare Relațională

3.5 Logica Rescrierii

4 Prover Logică Separatoare Relațională

Ordinea subsecțiunilor mai poate fi schimbată ulterior. Capitolul cel mai semnificativ din lucrare, va conține toate detaliile de implementare ale soluției curente

- 4.1 Prezentare Generală Maude
- 4.2 Semantică executabilă a limbajului folosit
- 4.3 Modelarea Logicilor Separatoare
- 4.4 Interacțiunea prin LOOP-MODE
- 4.5 Schemă Generală de funcționare
- 4.6 Procese automatizate
 - 4.6.1 Potrivire automată a axiomelor și a goal-urilor dovedite
 - 4.6.2 Demonstrarea automată a implicațiilor
- 4.7 Interfața
- 4.8 Viitor
- 5 Cercetări adiționale
 - 5.1 Automatizarea demonstrațiilor
 - 5.2 Extensii pentru programe concurente
 - 5.3 Java+ITP
- 6 Concluzii