MQP Title

A Major Qualifying Project Report
submitted to the Faculty of

Worcester Polytechnic Institute

in partial fulfillment of
the requirements for the degree of
Bachelor of Science

by

.....
Michael Ficarra

on

Thursday 23rd September, 2010

.....
Daniel Dougherty
professor, project advisor

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam diam diam, elementum eget fringilla non, hendrerit at lacus. Vivamus at purus mauris. Praesent vel nibh ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin faucibus gravida purus, ut volutpat quam cursus in. Praesent nulla est, laoreet a mattis id, rutrum ac augue. Etiam bibendum urna molestie sem lobortis lacinia. Curabitur aliquam nisl pulvinar enim aliquam adipiscing. Praesent non ligula eget odio lacinia sagittis. Quisque blandit metus vel ipsum elementum tincidunt. Nulla viverra dui nec metus viverra ut interdum tellus ullamcorper. Quisque fringilla, felis vitae placerat facilisis, metus nibh sagittis lacus, quis volutpat tortor massa sit amet lectus. Proin eu arcu quam, at accumsan lectus. Proin dignissim consectetur nisl mattis scelerisque. Nulla ac enim ipsum. Pellentesque tellus massa, aliquam nec malesuada eget, porta vitae metus. Mauris congue, enim eget suscipit tristique, mauris magna faucibus nunc, ac iaculis quam orci non nisl. Aliquam pharetra arcu vel sapien commodo dapibus. In commodo risus quis elit lobortis quis ornare tellus ultricies. Donec adipiscing, velit at gravida suscipit, ligula sapien ullamcorper mi, et tempor erat mi vel quam.

Table of Contents

List of Figures

List of Tables

1 Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum aliquam tortor eu tortor aliquam interdum. Nullam hendrerit dapibus dui, blandit lobortis velit feugiat non. Suspendisse fermentum ipsum non odio feugiat sed egestas justo suscipit. Aliquam dui nisi, pellentesque mattis luctus non, porta a erat. Sed sed sagittis odio. Nullam sit amet lacus ipsum. Vivamus tempor neque ac dolor venenatis tristique.

1.1 Section 1

Sed pharetra semper sagittis. Ut molestie elementum mi, eu pellentesque nulla aliquam sit amet. Sed posuere gravida orci sit amet cursus. Nunc mi odio, facilisis nec pulvinar sit amet, consequat non nisi. Pellentesque commodo quam mollis massa faucibus sodales. Sed condimentum egestas mollis. In iaculis elit non nibh ultricies faucibus. Sed ut est sed lacus fermentum tristique. Nullam nunc est, feugiat vitae ornare sed, porta a magna. Nullam nisl felis, porta congue laoreet in, vulputate non elit.

1.1.1 Subsection

In semper, massa nec hendrerit semper, leo sapien porttitor mauris, non consectetur urna lorem in tortor. Aliquam placerat, sapien viverra faucibus porta, felis magna rutrum augue, a commodo arcu mi a mi. Etiam tristique urna quis erat pretium fringilla. Aenean ut nunc erat. Maecenas at turpis non diam vehicula sollicitudin. Duis vel tortor ut diam dignissim mattis ac at felis. Etiam ac interdum nibh. Phasellus ut nibh purus. Donec sem dolor, hendrerit sagittis pellentesque sit amet, pellentesque vel ipsum.

Nunc mollis fringilla metus, sed aliquam mi hendrerit ut. Maecenas suscipit nibh vitae dui aliquet convallis a nec sem. Aliquam ut mi ligula, non mollis enim. Nunc eget libero vitae tellus lacinia tincidunt. Cras sodales pellentesque sagittis. Sed vitae tincidunt augue. Sed elementum blandit consequat. Mauris tincidunt porta ipsum, ac sagittis nibh lacinia ut. Vestibulum aliquam augue eget tellus ultrices ac interdum neque molestie. Integer at dolor magna, posuere commodo sapien.

Proin ullamcorper, magna ac mollis ultrices, risus tellus pharetra magna, eget pellentesque elit sem dignissim metus. Maecenas tincidunt varius congue. Nunc vel luctus elit. In faucibus pellentesque ipsum, sed suscipit ligula volutpat at. Fusce sagittis nisl nec lorem eleifend sit amet pretium purus tempor. Pellentesque commodo, mi dignissim placerat consectetur, est risus sollicitudin dui, ac cursus magna nisi bibendum sem.

Donec pulvinar egestas est vel scelerisque. Donec at libero ante, a vulputate nisi. Fusce at sem diam. Quisque vel enim vel odio tincidunt commodo. In ut nibh nec ligula facilisis sagittis quis sit amet lectus. Nulla mi urna,

aliquam sit amet convallis vitae, commodo sed nisl. Phasellus volutpat blandit elementum. Curabitur volutpat, arcu eu interdum rhoncus, sem libero dapibus nisl, eget viverra tellus ligula hendrerit est. Phasellus ut felis quis nisl vehicula aliquet.

1.2 Section 2

Morbi ac augue ac nisi euismod venenatis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vulputate dolor vitae purus dignissim ullamcorper. Aenean ante orci, posuere eu convallis at, feugiat et elit. Aenean rhoncus eros ut dui pellentesque convallis. Curabitur consectetur pretium varius. Vestibulum interdum convallis eros id pretium. Maecenas aliquam erat varius odio molestie consectetur.

Aliquam vulputate mi non metus lacinia rutrum. In hac habitasse platea dictumst. Quisque magna nisi, lacinia quis molestie in, varius eu diam. Nullam tristique porta ante, malesuada egestas purus viverra nec. Nulla vestibulum pretium massa id mattis. Donec ut velit urna. Suspendisse potenti. Vivamus vitae consectetur quam. Mauris non ante mauris. Nulla id lectus ut velit mollis convallis vel non leo. Integer ac pulvinar nisl. Maecenas posuere fringilla consectetur.

References

[1] A Cottrell, *Word Processors: Stupid and Inefficient*, www.ecn.wfu.edu/~cottrell/wp.html

A Appendix Name

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum rutrum, libero id aliquam elementum, neque augue aliquam lorem, a malesuada augue mauris a felis. Curabitur erat ipsum, dictum vitae cursus in, dignissim quis nunc. Phasellus nec ante felis. Cras et lectus purus, at feugiat ligula. Morbi varius sapien et felis ultrices ac semper risus pretium. Duis aliquam, dui a sollicitudin semper, justo magna semper massa, pulvinar suscipit massa tellus hendrerit libero. Nullam pretium dolor sit amet lacus fermentum malesuada. Suspendisse at imperdiet diam. Cras ut lectus sem, quis rutrum ligula. Donec magna purus, bibendum in lacinia nec, luctus quis tellus. Suspendisse ullamcorper, nunc sit amet posuere placerat, sem nisi varius enim, in mattis neque risus a ipsum. Donec pulvinar congue feugiat. Vivamus dapibus, nunc et vehicula tincidunt, erat arcu dictum mauris, sit amet congue augue mauris ac dui. Duis suscipit blandit eros, sit amet facilisis lectus mollis ac.

Ut sit amet tellus metus, sed porta ipsum. Donec porttitor imperdiet laoreet. Curabitur congue convallis justo vel porta. Sed vel pharetra magna. Integer et odio eu lacus commodo feugiat et et erat. Etiam rutrum, arcu nec porta semper, mauris nibh iaculis quam, nec hendrerit quam tortor feugiat eros. Ut non posuere est. Integer hendrerit ultricies leo sit amet dapibus. Fusce eget porttitor turpis. Duis urna tellus, condimentum in consectetur et, ornare et nisl. Pellentesque dignissim, nisl ac pulvinar luctus, ipsum tortor tincidunt arcu, consequat sagittis massa orci et augue. Aliquam erat volutpat. Vestibulum in diam risus. Maecenas viverra, arcu fringilla dignissim tincidunt, nibh est luctus purus, ac bibendum metus purus quis leo. In varius mi vitae quam adipiscing consequat. Mauris consequat, lectus at dictum gravida, felis leo commodo libero, vitae ornare lacus ante sit amet lorem. Pellentesque congue iaculis ligula et bibendum.

A.1 Appendix Subsection

Vestibulum et egestas est. Integer sed purus ipsum. Vivamus pharetra feugiat urna, in suscipit turpis fermentum sed. Curabitur ut ligula leo, quis molestie libero. Nulla sollicitudin consequat sodales. Nulla nulla ipsum, faucibus vel luctus id, eleifend a erat. Etiam pharetra, lectus in dapibus laoreet, elit turpis facilisis libero, nec gravida lacus lacus in augue. Sed euismod ante nec dui hendrerit adipiscing. Donec non libero ut ligula malesuada fringilla. Ut justo nunc, aliquet consequat pharetra sed, placerat et justo. Mauris venenatis erat eget leo ornare tempus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eu lectus urna, vitae pretium sem. Aenean vulputate turpis eu tortor pulvinar mattis. Vivamus quam lectus, bibendum quis rutrum quis, ultrices sed est. Cras neque.

B Chase code

```
1  module Chase where
2  import Parser
3  import Helpers
4  import Debug.Trace
5  import Data.List
6
7  chaseVerify :: [Formula] -> [Formula]
8  -- verifies that each formula is in positive existential form and performs some
9  -- normilization on implied/constant implications
10 chaseVerify formulae =
11     let isNotPEF = not.isPEF in
12     map (\f -> case f of
13         Implication a b ->
14             if isNotPEF a || isNotPEF b then error ("implication must be in positive existential form: " ++ s
15             else f
16         _ ->
17             if isNotPEF f then error ("formula must be in positive existential form: " ++ showFormula f)
18             else (Implication Tautology f)
19     ) formulae
20
21 chase :: [Formula] -> [Model]
22 -- runs the chase algorithm on a given theory and returns a list of models that
23 -- satisfy it
24 chase formulae = chase' (chaseVerify formulae) ([],[(mkModel [] [])])
25
26 chase' :: [Formula] -> ([Model],[Model]) -> [Model]
27 -- used by the chase function to hide the model identity argument
28 chase' formulae (done,[]) = done
29 chase' formulae (done,pending) =
30     let self = chase' formulae in
31     let (p:ending) = pending in
32     trace ("running chase on " ++ show (done,pending)) $
33     if all (\f -> holds p (UniversalQuantifier (freeVariables f) f)) formulae then
34         trace ("  all formulae in theory hold for model " ++ showModel p) $
35         trace ("  moving model into done list") $
36         self (union done [p],ending)
37     else
38         let possiblySatisfiedModels = attemptToSatisfyFirstFailure p formulae in
39         trace ("  at least one formula does not hold for model " ++ showModel p) $
40         trace ("  unioning " ++ show ending ++ " with [" ++ intercalate ", " (map showModel possiblySatisfiedM
41         self (done, union ending possiblySatisfiedModels)
42
43 attemptToSatisfyFirstFailure :: Model -> [Formula] -> [Model]
44 -- checks if each formula holds, sequentially, until one does not, then tries
45 -- to satisfy that formula
46 attemptToSatisfyFirstFailure model (f:ormulae) =
47     let self = attemptToSatisfyFirstFailure model in
48     if holds model (UniversalQuantifier (freeVariables f) f) then self ormulae
49     else attemptToSatisfy model f
50
51 attemptToSatisfy :: Model -> Formula -> [Model]
52 -- returns a model that is altered so that the given formula will hold
53 attemptToSatisfy model formula =
54     let f' = UniversalQuantifier (freeVariables formula) formula in
55     trace ("  attempting to satisfy (" ++ showFormula formula ++ ")") $
56     attemptToSatisfy' model [] f'
57
58 attemptToSatisfy' :: Model -> Environment -> Formula -> [Model]
59 -- hides the environment identity in the `attemptToSatisfy` function arguments
60 attemptToSatisfy' model env formula =
61     let (domain,relations) = model in
62     let domainSize = length domain in
63     let self = attemptToSatisfy' model in
64     -- trace ("  attempting to satisfy (" ++ showFormula formula ++ ") with env " ++ show env) $
65     case formula of
66         Tautology -> [model]
67         Contradiction -> []
68         Or a b -> union (self env a) (self env b)
69         And a b -> concatMap (\m -> attemptToSatisfy' m env b) (self env a)
70         Implication a b -> if holds' model env a then self env b else []
71         Atomic predicate vars ->
72             let newRelation = mkRelation predicate (length vars) [genNewRelationArgs env vars (fromIntegral (
73             let newModel = mkModel (mkDomain domainSize) (mergeRelation newRelation relations) in
74             trace ("    adding new relation: " ++ show newRelation) $
75             [newModel]
76         ExistentialQuantifier [] f -> self env f
77         ExistentialQuantifier (v:vs) f ->
78             let f' = ExistentialQuantifier vs f in
79             let nextDomainElement = fromIntegral $ (length domain) + 1 in
80             if any (\v' -> holds' model (hashSet env v v') f') domain then
81                 trace ("    " ++ showFormula formula ++ " already holds") $
```

```
82                  [model]
83              else
84                  trace ("    adding new domain element " ++ show nextDomainElement ++ " for variable " ++ (sho
85                  attemptToSatisfy' (mkDomain nextDomainElement,relations) (hashSet env v nextDomainElement) f'
86          UniversalQuantifier [] f -> self env f
87          UniversalQuantifier (v:vs) f ->
88              let f' = UniversalQuantifier vs f in
89              concatMap (\v' -> self (hashSet env v v') f') domain
90          _ -> error ("formula not in positive existential form: " ++ showFormula formula)
91
92  genNewRelationArgs :: Environment -> [Variable] -> DomainElement -> [DomainElement]
93  -- for each Variable in the given list of Variables, retrieves the value
94  -- assigned to it in the given environment, or the next domain element if it
95  -- does not exist
96  genNewRelationArgs env [] domainSize = []
97  genNewRelationArgs env (v:ars) domainSize =
98      let self = genNewRelationArgs env in
99      case lookup v env of
100         Just v' -> v' : (self ars domainSize)
101         _ -> (domainSize+1) : (self ars (domainSize+1))
```