T.C.
ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM2536 Bulanık Mantık – İkinci Proje Raporu

Alp Ertunga Elgün 19290238

2021

İçindekiler

Bölüm 1
Özet

Bu rapor BLM2536 Bulanık Mantık dersinin ikinci projesi için hazırlanmıştır.Programla dili olarak buradan ulaşabileceğiniz Lua dili kullanılmıştır. Kendiniz compile edip farklı inputlar için denemek isterseniz buradan ulaşabileceğiniz LuaJIT compiler'ı kullanmanızı tavsiye ederiz.

Kaynak kodlarına bu adresteki Github sayfasından ulaşabilirsiniz. Comment'lerle birlikte gayet anlaşılabilir bir program olsa da bu raporda daha detaylı bir şekilde nasıl çalıştığı anlatılmıştır. Kullanılan LuaFuzzy kütühanesinin Github sayfasına ise buradan ulaşabilirsiniz.

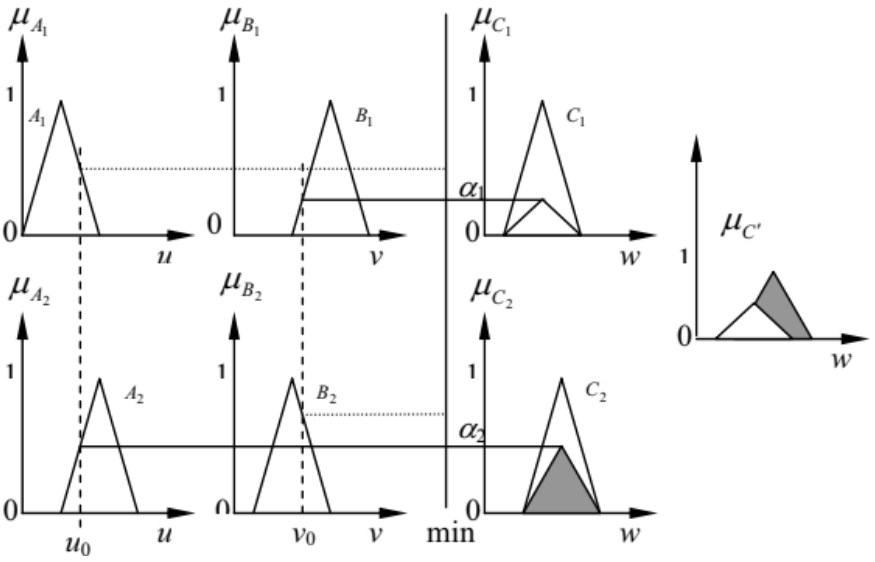
Program geliştirilirken kaynak olarak Roberto Lerusalimschy'nin yazdığı Programming in Lua kitabı ve LuaFuzzy kütüphanesi kullanılmıştır. Fuzzy algoritmaları geliştirirken kaynak olarak Murat Osmanoğlu'nun dersleri ve Kwang Hyung Lee'nin yazdığı First Course on Fuzzy Theory and Applications kitabı kullanılmıştır.

İlk projenin raporundan farklı olarak, bu raporun 2. bölümünde tüm system açıklanmamış, sadece Larsen çıkarım yöntemi açıklanmıştır. 3. Bölümde, ilk raporda açıkladığımız için, LuaFuzzy kütüphanesi detaylı olarak açıklanmamış, sadece Larsen çıkarım yönteminin entegrasyonu ve kütüphanenin projemizdeki bulanık mantık sistemini oluşturmak için nasıl kullanıldığı anlatılmıştır. 4. Bölümde ise bazı inputlar ve oluşturdukları sonuçlar verilmiştir.

Bölüm 2
Larsen Method

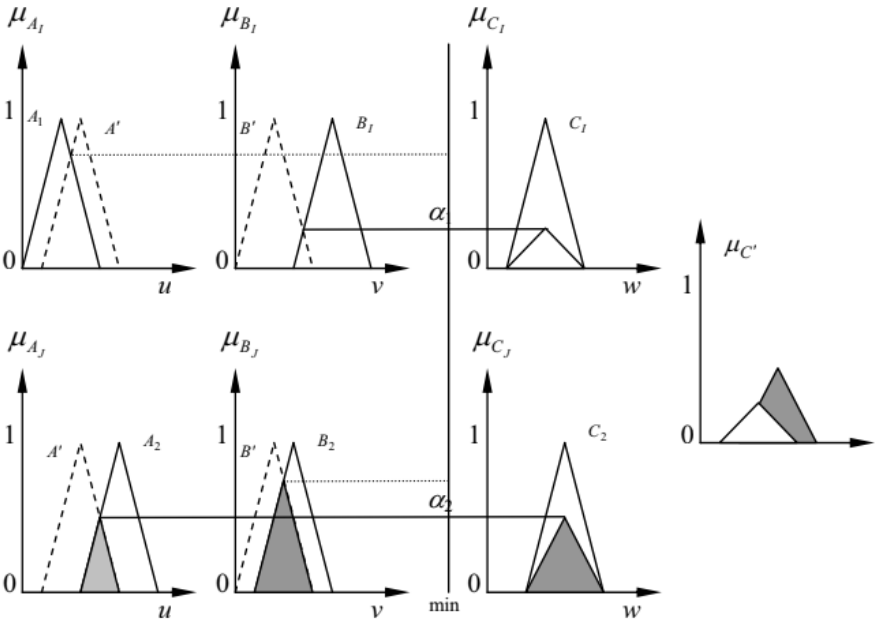
2.1 Singleton Input

Larsen Method'unda implication kısmında çarpma işlemi kullanılır. İlk önce bize input olarak verilen verinin üyelik değerlerini buluruz. Sonra bu değerlerden küçük olanını sonuç fonksiyonuyla çarparız. Şekil ile gösterirsek:



2.2 Fuzzy Input

Eğer Input bir fuzzy set ise, bu input ile membership fonksiyonun kesişim noktasının üyelik değerini buluruz. . Sonra bu değerlerden küçük olanını sonuç fonksiyonuyla çarparız. Şekil ile gösterirsek:



Bölüm 3
Program

3.1 Kütüphaneye Larsen Method'unun Entegrasyonu

Larsen Method'unu kullanmak için yapmamız gereken işlem implication fonksiyonunu çarpma işlemine değiştirmektir. Objeyi oluşturan luafuzzy() fonksiyonunda implication işlemi fuzzy.implicmethod olarak verilmiştir. Çarpma işlemi ise tprod(a1,a2) fonksiyonu olarak tanımlanmıştır. Yapmamız gereken tek şey fuzzy. Implicmethod değişkenine tprod fonksiyonunu atamaktır.

```
433 function luafuzzy()
434   -- create new table
435   local fuzzy = {}
436
437   -- and method
438   fuzzy.andmethod = tmin
439
440   -- or method
441   fuzzy.ormethod = tmax
442
443   -- implication method
444   fuzzy.implicmethod = tprod -- tmin for Mamdani
445
446   -- aggregation method
447   fuzzy.aggregmethod = tmax
448
449   -- deffuzification method
450   fuzzy.defuzzmethod = centroid
451
452   -- step used to compute a discrete fuzzyset
453   fuzzy.step = 0.00001
454
455   -- table os rules
456   fuzzy.rules = {}
457
458   -- table of inputs
459   fuzzy.inps = {}
460
461   -- tabla of outputs
462   fuzzy.outs = {}
463
464   -- method to add a new input
465   fuzzy.addinp = addinp
466
467   -- method to add a new output
468   fuzzy.addout = addout
469
470   -- method to add a new rule
471   fuzzy.addrule = addrule
472
473   -- method to solve the fuzzy inferece system
474   fuzzy.solve = solvefuzzy
475
476   return fuzzy
```

```
42  ----------------------------------------------------
43  -- Tprod
44  ----------------------------------------------------
45  function tprod(a1,a2)
46    local y = a1*a2
47    if y > 1 then y = 1 end -- stop in one(limit)?
48    return y
49  end
```

3.2 Main Program

Burada LuaFuzzy'de tanımlı olan fonksiyonları cağırarak sistemimizi oluşturuyor ve sonra da çözüp ekrana basıyoruz.

```
--
-- Load Fuzzy Lua Framework
--
require 'luafuzzy'

-----------------------------
-- Fuzzy Systems
-----------------------------

-- Systems are global

fuzzyHouse = luafuzzy()
fuzzyApplicant = luafuzzy()
fuzzyCredit = luafuzzy()
```

İlk başta LuaFuzzy sistemini yüklüyoruz. Ondan sonra da 3 adet fuzzy sistemi oluşturup bunları global değişkenler olan fuzzyHouse, fuzzyApplicant ve fuzzyCredit değişkenlerine atıyoruz. Geliştirdiğimiz control sisteminde 3 farklı fuzzy çıkarım yaptığımız için, 3 adet fuzzy system oluşturduk.

```
---------------------------------
-- Fuzzy Inputs
---------------------------------

--
-- House
--

-- Market Value
local marketValue = fuzzyHouse:addinp( 'marketValue', 0, 1000
marketValue:addlingvar( 'low', trapmf, { 0, 0, 50, 100 } )
marketValue:addlingvar( 'medium', trapmf, { 50, 100, 200, 250
marketValue:addlingvar( 'high', trapmf, { 200, 300, 650, 850 }
marketValue:addlingvar( 'veryhigh', trapmf, { 650, 850, 1000,

-- Location
local location = fuzzyHouse:addinp( 'location', 0, 10 )
location:addlingvar( 'bad', trapmf, { 0, 0, 1.5, 4 } )
location:addlingvar( 'fair', trapmf, { 2.5, 5, 6, 8.5 } )
location:addlingvar( 'excellent', trapmf, { 6, 8.5, 10, 10 } )

--
-- Applicant
--

-- Asset
local asset = fuzzyApplicant:addinp( 'asset', 0, 1000 )
asset:addlingvar( 'low', trimf, { 0, 0, 150 } )
asset:addlingvar( 'medium', trapmf, { 50, 250, 450, 650 } )
asset:addlingvar( 'high', trapmf, { 500, 700, 1000, 1000 } )
```

Sonra input değişkenlerini tanımlıyacağız. Bu değişkenler fuzzy sistemin bir parçası olduğu için tanımlarken "self" argümanına fuzzy sistemin kendisini atamalıyız. 1. Argüman değişkenin ismi, 2. argüman min value, 3. argüman ise max value olacak şekilde inputlarımızı tanımlıyoruz.

Sonra bu değişkenlerin linguistic kısımlarını tanımlamalıyız. Bu değişkenler input'un bir parçası olduğu için tanımlarken "self" argümanına input'un kendisini atamalıyız. 1. Argümana linguistic ismi, 2. argümana membership fonksiyonu, 3. argümana ise table içinde membership fonksiyonunun argumanlarını atıyoruz.

```
---------------------------------
-- Fuzzy Outputs
---------------------------------

--
-- House
--

local houseOut = fuzzyHouse:addout( 'house', 0, 10 )
houseOut:addlingvar( 'verylow', trimf, { 0, 0, 3 } )
houseOut:addlingvar( 'low', trimf, { 0, 3, 6 } )
houseOut:addlingvar( 'medium', trimf, { 2, 5, 8 } )
houseOut:addlingvar( 'high', trimf, { 4, 7, 10 } )
houseOut:addlingvar( 'veryhigh', trimf, { 7, 10, 10 } )
```

Aynı işlemi output değeri için yaptıktan sonra sıra kuralları tanımlamaya geliyor. 42 tane kural olduğu ve hepsini ana programda tanımlarsak karmaşa yaratacağı için kuralları başka bir file'da tanımlayıp ana file'a yüklüyoruz.

```
-----------------------------------
-- Rules
-----------------------------------

local rules = require ("rulematrix")
```

Kuralları tanımlarken, kurallar fuzzy sistemin bir parçası olduğu için "self" argümanı yerine fuzzy sistemin kendisini atamalıyız. Fuzzy değişkenini global tanımladığımız için farklı bir file'da kullanabiliyoruz. 1. Argümana kuralın ağırlığını, 2. Argümana ise kuralda kullandığımız connection methodu atıyoruz. Geliştirdiğimiz kontrol sistemiminde kuralların ağırlığı eşit olduğu için hepsinin ağırlığına 1, hepsinin connection method'u AND olduğu için 2. argümanlarına "andmethod" değerini atıyoruz.

```
11  -- Rule 1
12  M.r1 = fuzzyHouse:addrule( 1, 'andmethod' )
13  M.r1:addpremise( false, 'marketValue', 'low' )
14  M.r1:addimplic( false, 'house', 'low' )
15
16  -- Rule 2
17  M.r1 = fuzzyHouse:addrule( 1, 'andmethod' )
18  M.r1:addpremise( false, 'location', 'bad' )
19  M.r1:addimplic( false, 'house', 'low' )
```

Sonra bu kurallara öncülleri ve bu öncüller doğru ise çıkacak sonucu ekliyoruz. Öncüller ve gerektirmeler kuralın bir parcası olduğu için "self" argümanına kuralın kendisini atıyoruz. 1. argümana not kullanıp kullanmadığımızı, 2. argümana değişkenin ismini, 3. argümana da değişkenin alması gereken linguistic değişkeni atıyoruz.

En son olarak consol'dan Inputları okuyoruz. Okuduğumuz Inputlar string formatında olduğu için tonumber fonksiyonu ile bu değerleri number'a dönüştürdükten sonra solve fonksiyonunu kullanarak sistemi çözüyoruz. Bulduğumuz değerleri 3. sistemin çözümünde kullanacağımız için bir değişkende depolamayı unutmamalıyız.

```
127 --------------------------------
128 io.write("Please enter Market Value.", "\n")
129 local marketValueInput = io.read()
130 marketValueInput = tonumber(marketValueInput)
131
132 io.write("Please enter Location.", "\n")
133 local locationInput = io.read()
134 locationInput = tonumber(locationInput)
135
136 io.write("\n", "Finding House Value ...", "\n")
137 local houseSolved = fuzzyHouse:solve( marketValueInput, locationInput )
138 io.write("House value is: ", houseSolved, "\n\n")
139
140 io.write("Please enter Asset.", "\n")
141 local assetInput = io.read()
142 assetInput = tonumber(assetInput)
143
144 io.write("Please enter Income.", "\n")
145 local incomeInput = io.read()
146 incomeInput = tonumber(incomeInput)
147
148 io.write("\n", "Finding Applicant ...", "\n")
149 local applicantSolved = fuzzyApplicant:solve(assetInput, incomeInput)
150 io.write("Applicant is: ", applicantSolved, "\n\n")
151
152 io.write("Please enter Interest.", "\n")
153 local interestInput = io.read()
154 interestInput = tonumber(interestInput)
155
156 io.write("\n", "Finding Credit...", "\n")
157 local creditSolved = fuzzyCredit:solve(houseSolved, applicantSolved, incomeInput, interestInput)
158 io.write("Credit is: ", creditSolved)
```

Bölüm 4
Inputlar

İlk olarak tam ortadan değerler alarak bir deneme yapalım. Market Value 500, Location 5 olursa House value yaklaşık 7 çıkar. Asset 500, Income 50 olursa Applicant 7.3235608996708 çıkar. En son Interesti 5 girip Credit'I hesaplarsak sonuç 230.16926806817 çıkar.

Denediğim diğer değerleri Aşağıda terminalden aldığım ekran görüntüsünde bulabilirsiniz.

```
PS D:\Alp\Code\LuaJIT> .\luajit.exe fuzzy.lua
Please enter Market Value.
500
Please enter Location.
5

Finding House Value ...
House value is: 6.9999999999999

Please enter Asset.
500
Please enter Income.
50

Finding Applicant ...
Applicant is: 7.3235608996708

Please enter Interest.
5

Finding Credit...
Credit is: 230.16926806817
```

```
PS D:\Alp\Code\LuaJIT> .\luajit.exe fuzzy.lua
Please enter Market Value.
10
Please enter Location.
1

Finding House Value ...
House value is: 2.499966666667

Please enter Asset.
10
Please enter Income.
1

Finding Applicant ...
Applicant is: 1.5555296295056

Please enter Interest.
1

Finding Credit...
Credit is: 41.665000000008
PS D:\Alp\Code\LuaJIT>
```

```
PS D:\Alp\Code\LuaJIT> .\luajit.exe fuzzy.lua
Please enter Market Value.
990
Please enter Location.
9

Finding House Value ...
House value is: 9.0000333333266

Please enter Asset.
990
Please enter Income.
90

Finding Applicant ...
Applicant is: 8.4444703704887

Please enter Interest.
9

Finding Credit...
Credit is: 200.43369981019
```

```
PS D:\Alp\Code\LuaJIT> .\luajit.exe fuzzy.lua
Please enter Market Value.
456.12345
Please enter Location.
5.4321

Finding House Value ...
House value is: 6.9999999999999

Please enter Asset.
567.98321
Please enter Income.
98.76543

Finding Applicant ...
Applicant is: 8.4444703704863

Please enter Interest.
2.34567

Finding Credit...
Credit is: 263.88537039513
```