# T.C. ANKARA ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# BLM2536 Bulanık Mantık – İkinci Proje Raporu

# Alp Ertunga Elgün 19290238

### 2021

# İçindekiler

# [Bölüm 1 Özet](#_Bölüm_1_Özet)

# [Bölüm 2 Larsen Method](#_Bölüm_2_Larsen)

## [2.1 Singleton Input](#_2.1_Singleton_Input)

## [2.2 Fuzzy Input](#_2.2_Fuzzy_Input)

# [Bölüm 3 Program](#_Bölüm_3_Program)

## [3.1 Kütüphaneye Larsen Method’unun Entegrasyonu](#_3.1_Kütüphaneye_Larsen)

## [3.2 Main Program](#_3.2_Main_Program)

# [Bölüm 4 Inputlar](#_Bölüm_4_Inputlar)

# Bölüm 1 Özet

Bu rapor BLM2536 Bulanık Mantık dersinin ikinci projesi için hazırlanmıştır.Programla dili olarak [buradan](http://www.lua.org) ulaşabileceğiniz Lua dili kullanılmıştır. Kendiniz compile edip farklı inputlar için denemek isterseniz [buradan](https://luajit.org) ulaşabileceğiniz LuaJIT compiler’ı kullanmanızı tavsiye ederiz.

Kaynak kodlarına [bu](https://github.com/Sayuris1/fuzzy2) 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](https://github.com/lorensi/luafuzzy) ulaşabilirsiniz.

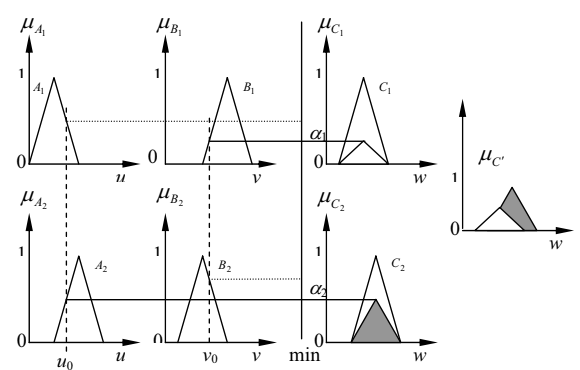
Program geliştirilirken kaynak olarak Roberto Lerusalimschy’nin yazdığı [Programming in Lua](https://www.lua.org/pil/) kitabı ve LuaFuzzy kütüphanesi kullanılmıştır. Fuzzy algoritmaları geliştirirken kaynak olarak Murat Osmanoğlu’nun dersleri ve Kwang Hyung Lee’nin yazdığı [First Course on Fuzzy Theory and Applications](https://www.springer.com/gp/book/9783540229889) 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.

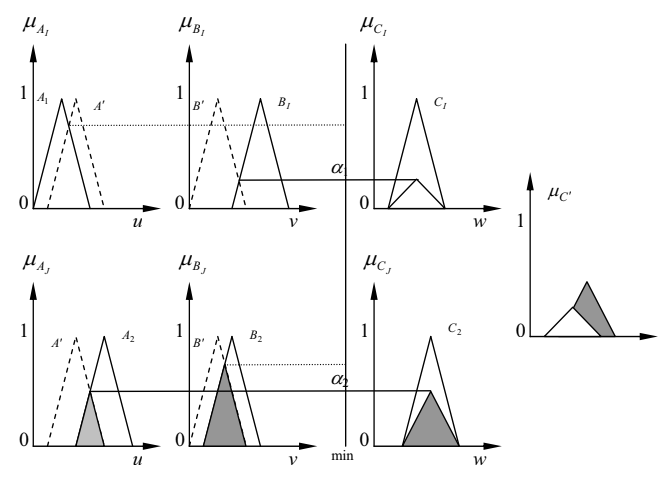
# Bölüm 2 Larsen Method

## 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:



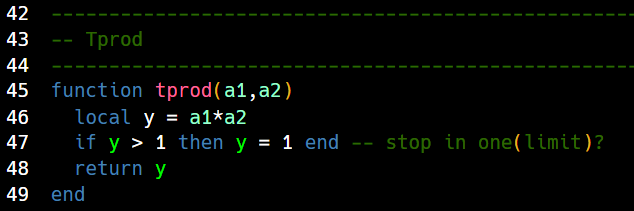
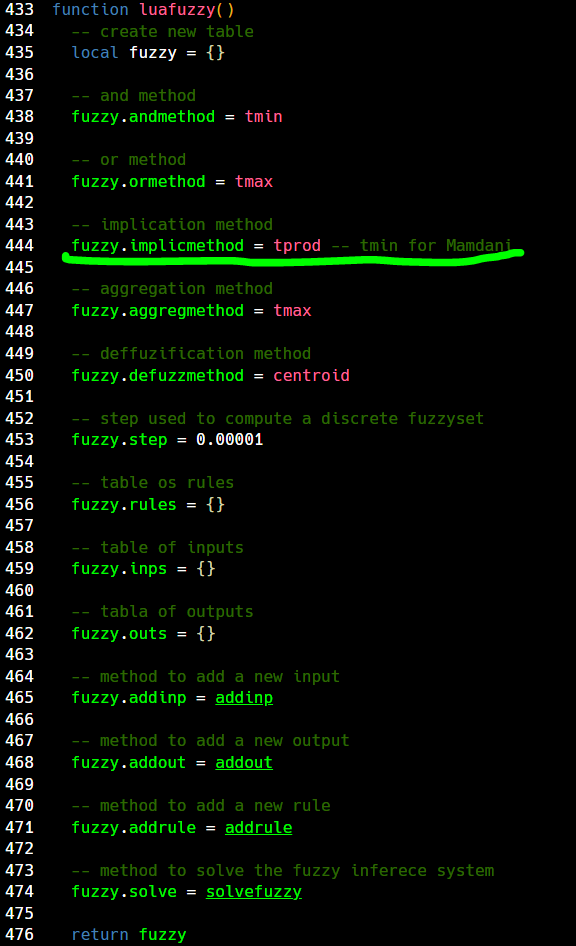
## 2.2 Fuzzy Input

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

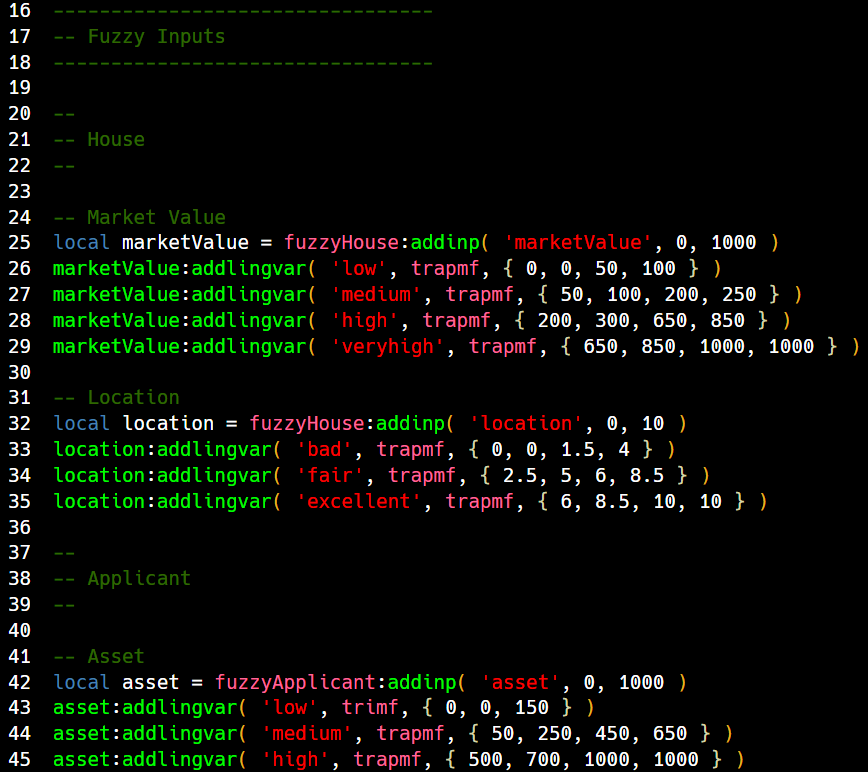
Larsen Method’unu kullanmak için yapmamız gereken işlem implication fonksiyonunu çarpma işlemine değiştirmektir. Objeyi oluşturan luafuzzy() fonksiyonunda implication işlemi fuzzy.implicmethod olarak verilmiştir. Çarpma işlemi ise tprod(a1,a2) fonksiyonu olarak tanımlanmıştır. Yapmamız gereken tek şey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



## C:\Users\Sayuri\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.png3.2 Main Program

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

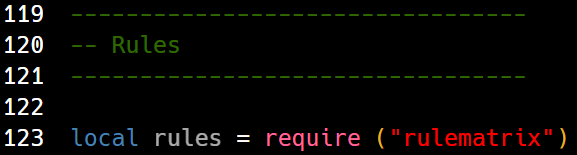
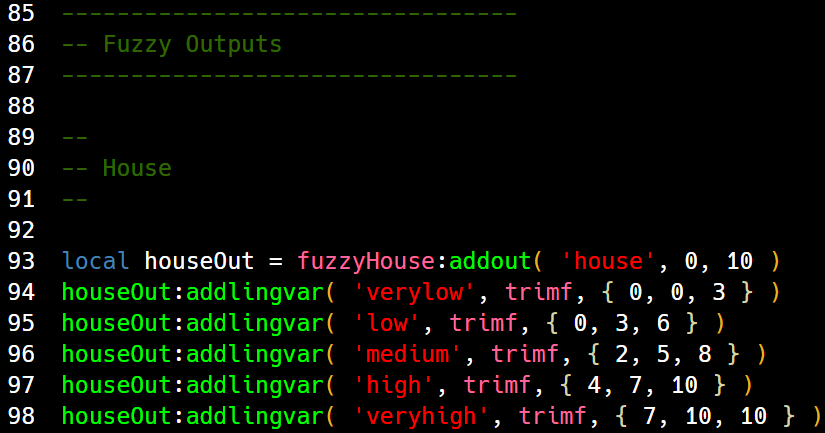
İlk başta LuaFuzzy sistemini yüklüyoruz. Ondan sonra da 3 adet fuzzy sistemi oluşturup bunları global değişkenler olan fuzzyHouse, fuzzyApplicant ve fuzzyCredit değişkenlerine atıyoruz. Geliştirdiğimiz control sisteminde 3 farklı fuzzy çıkarım yaptığımız için, 3 adet fuzzy system oluşturduk.



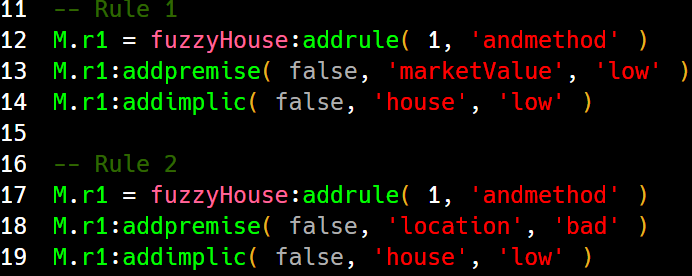
Sonra input değişkenlerini tanımlıyacağız. Bu değişkenler fuzzy sistemin bir parçası olduğu için tanımlarken “self” argümanına fuzzy sistemin kendisini atamalıyız. 1. Argüman değişkenin ismi, 2. argüman min value, 3. argüman ise max value olacak şekilde inputlarımızı tanımlıyoruz.

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.

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.

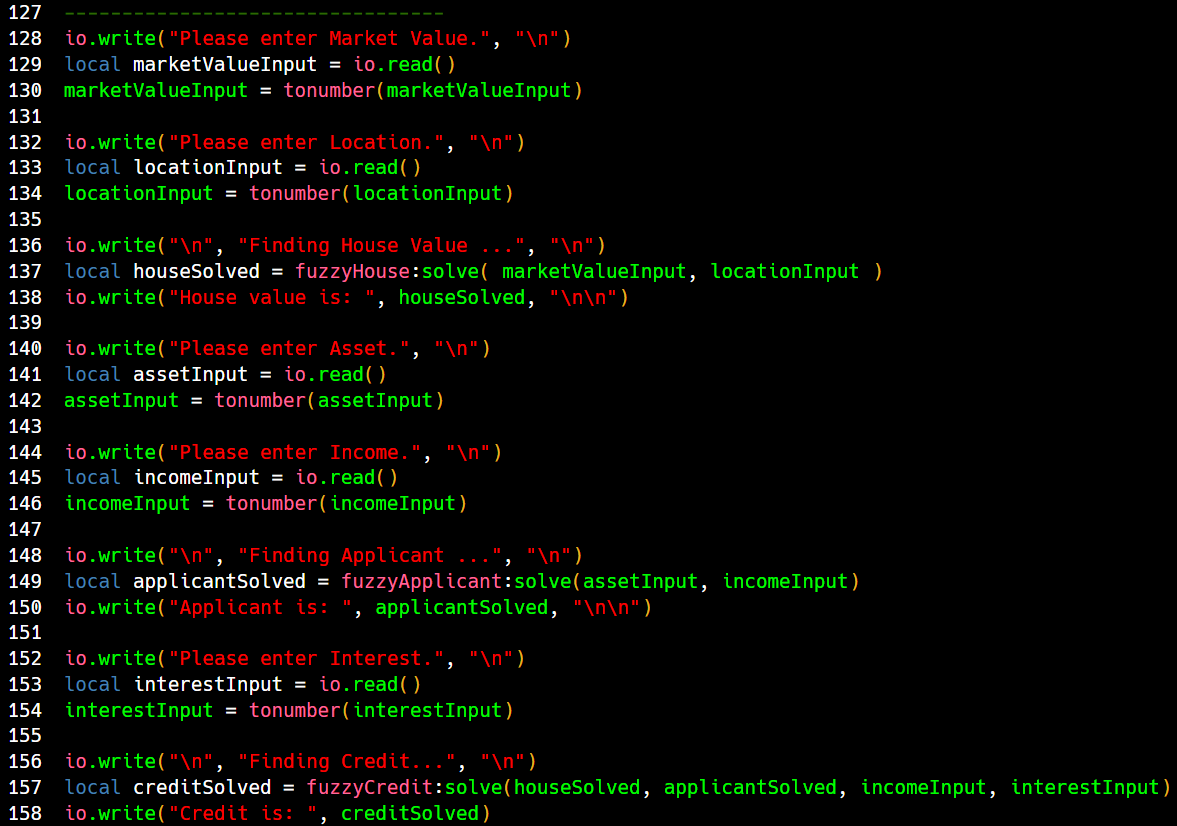


Kuralları tanımlarken, kurallar fuzzy sistemin bir parçası olduğu için “self” argümanı yerine fuzzy sistemin kendisini atamalıyız. Fuzzy değişkenini global tanımladığımız için farklı bir file’da kullanabiliyoruz. 1. Argümana kuralın ağırlığını, 2. Argümana ise kuralda kullandığımız connection methodu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



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 dönuştürdükten sonra solve fonksiyonunu kullanarak sistemi çözüyoruz. Bulduğumuz değerleri 3. sistemin çözümünde kullanacağımız için bir değişkende depolamayı unutmamalıyız.



# 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.

