Application of L-Fuzzy Relation to Social Choice Theory

Clement Frimpong Osei

Supervised by
Michael Winter

Department of Computer Science
Brock University

August 23, 2024

Social Choice Theory

Social Choice Theory deals with combining preferences or choices. That is, if R_i is a preference relation of individual i for a set of individuals, then $F(R_1, \ldots, R_n)$ is the combined preference according to an aggregation rule F.

We consider three (3) ways by which the preferences of an individual are presented:

1. **Preference Structures** (P, I): This is composed of two relations; a strict preference relation and an indifference relation.*Strict Preference Relation* (P) poses the notion of how we prioritize or rank various options based on our personal criteria. Hansson [1] simply put it as a binary relation reflecting "is preferred to" or "is better than" relations. An *Indifference relation* (I) indicates two items are equally

valued. Which means a person doesn't prefer one over the other. Or a person prefers one over the other for a reason and vice versa.

2. **Weak Preference** (R): This relation combines both preference and indifference into one unified relation;i.e., given two options, an individual will prefer one as much as he or she prefers the other.
3. **Choice Relation** (C): It identifies which options a person selects from a set of alternatives, where they might choose none, one, or multiple options.

Binary Representation

The concepts can be precisely modeled using binary relations between sets. If R is a subset of the Cartesian product $A \times B$ of two sets A and B, then R is a relation, denoted by $R : A \to B$. We write xRy to indicate that $(x, y) \in R$. A choice relation $C : P(X) \to X$ relates the power set of X (all subsets of X) to X. The notation MCx means that x is a possible choice from the set M.

A weak preference relation $R : X \to X$ indicates xRy, meaning x is weakly preferred to y. A strict preference relation $P : X \to X$ with xPy shows x is strictly preferred to y, while an indifference relation $I : X \to X$ with xIy means x is indifferent to y. The strict preference relation P is asymmetric, i.e., $\forall x, y : xPy \to$ not yPx, while the indifference relation I is symmetric, i.e., $\forall x, y : xIy \to yIx$.

Concrete Examples I

Consider a voter choosing between three candidates a, b, and c; we can have the following deductions:

1. **Strict Preference**:aPb, aPc, and bPc denotes a strict preference of the voter preferring a over b and c and b over c.

	a	b	c
a	0	1	1
b	0	0	1
c	0	0	0

2. **Weak Preference**: The voter might think a is at least as good as b or c and b is as good as c denoted by aRb, aRc, and bRc.

	a	b	c
a	1	1	1
b	0	1	1
c	0	0	1

Concrete Examples II

3. **Indifference Preference**: On the other hand the voter might see no difference between a and b, or b and c that is represented as aIb and bIc.

	a	b	c
a	1	1	0
b	1	1	1
c	0	1	1

4. **Choice Preference**: If the candidates are in a set M then a choice preference of the voter can be denoted as MCa, i.e., he or she prefers a from the set of candidates.

	a	b	c
a	1	1	1
b	0	1	1
c	0	0	1

Uncertainty

This points out the notion that there is a challenge or difficulty in determining an outcome due to the lack of clarity or inconsistency in people's preferences hence it is barely impossible to make a decision that satisfies everyone.

A clear example is that an individual in a voting scenario might only be able to indicate that he or she prefers a candidate over the others to a certain degree. Any aggregation of these kinds of preferences needs to take the degree of each individual's preference into account.

Also, an individual might have two criteria for preferring one car over the other, speed and costs. if the first car is faster than the second, then he or she may prefer that car to some extent because it is faster. If the second is cheaper, then he or she would prefer that also to some extent because it is cheaper.

Main Contribution

This thesis aims to explore the three previously mentioned relations and to identify the most general framework in which these concepts are interconnected or equivalent. The system is modeled using L-fuzzy sets and relations, focusing on L-fuzzy social choice theory and the mathematical representation of personal preferences under uncertainty.

L-Fuzzy Relation

An L-fuzzy relation is a relation between two sets where the degree of relatedness is measured using elements from a complete lattice L, rather than just values between 0 and 1 as in traditional fuzzy relations Gluck [2].

A fuzzy set assigns to each element a degree of membership. This degree indicates how much the element belongs to the set. A general approach uses a lattice, i.e., a specific partially ordered set, as membership values.

Application to Choice and Preference Relations

L-Fuzzy logic can be applied to a choice and preference relation by representing the different options or possibilities in fuzzy sets or values.

It entails assessing the degree of membership or preference of each option based on a specific criterion or condition, taking into account the fuzziness or uncertainty of the values involved. This enables one to make more detailed decisions and take into account personal preferences and priorities.

For example, consider an L-fuzzy relation $C : P(X) \to X$ as a choice relation. If a pair (M, x) is in C with degree a ($C(M, x) = a$), it indicates that one would choose alternative x from the set of alternatives M with a degree a. This allows specifying choices up to a certain degree of uncertainty.

Choice and Weak Relations I

We consider the relation between choice defined as $C : P(A) \to A$ and weak preferences defined as $R : A \to A$ in the following definitions and lemmas of Osei and Winter [3]. These Definitions and lemmas establish how one relation can be converted to the other.

Definition

1. $C_R = \varepsilon \backslash R^{\smile} \sqcap \varepsilon^{\smile}$
2. $R_C = C^{\smile}; (syQ(\pi^{\smile} \sqcup \rho, \varepsilon))^{\smile} \sqcap \pi^{\smile}); \rho^{\smile}$
3. *normal*: $\varepsilon \backslash \varepsilon; C \sqcap \varepsilon^{\smile} \sqsubseteq C$
4. *left quasi-reflexive*: $R \sqsubseteq (I \sqcap R); R$

Choice and Weak Relations II

We represent these preferences using the language of relations. The definition of normality is used to show how inclusions translate into regular first-order properties for binary relations.

$$
\begin{aligned}
& M(\varepsilon\backslash\varepsilon; C \sqcap \varepsilon^{\smile})x \\
& \quad \iff M(\varepsilon\backslash\varepsilon; C \sqcap \varepsilon^{\smile})x \sqcap M\varepsilon^{\smile}x \\
& \quad \iff M(\varepsilon\backslash\varepsilon; C \sqcap \varepsilon^{\smile})x \sqcap x\varepsilon M \\
& \quad \iff x\varepsilon M \sqcap \forall y : y\varepsilon M \to y(\varepsilon; C)x \\
& \quad \iff x\varepsilon M \sqcap \forall y : y\varepsilon M \to \exists N : y\varepsilon N \sqcap NCx
\end{aligned}
$$

A relation is **normal** if for each x in a set M, x is in M, and for every y in M, there is a set containing both x and y where x is chosen.

A **Left quasi-reflexive** relation means that every element is related to itself. If x is weakly preferred to y, then x must also be weakly preferred to itself.

Choice and Weak Relations III

Lemma

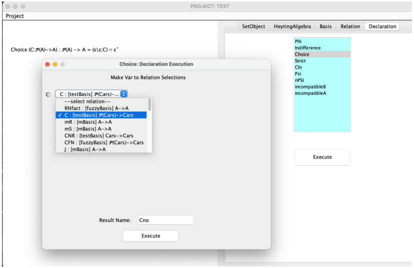
1. *Given* $C : P(A) \to A$*,* C *is normal iff* $\varepsilon \backslash \varepsilon; C \sqcap \varepsilon^{\smile} = C$
2. *Let* $R : A \to A$*, then* $\varphi(R)$ *is normal provided that* R *is left quasi-reflexive*
3. *Given* $C : P(A) \to A$*,* $\chi(C)$ *is left quasi-reflexive*
4. C *is normal iff* $\varphi(\chi(C)) = C$
5. R *is left quasi-reflexive iff* $\chi(\varphi(R)) = R$

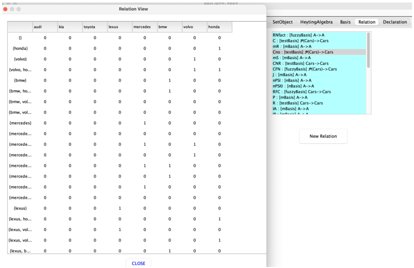
We take a look at various examples in the context of boolean and fuzzy lattices to cover the aforementioned lemmas.

Lemma	Operation	Properties	Lattice
1	Choice	normal	Boolean
2	$\varphi(R) = C$	R:reflexivity, C:normal	Boolean, Fuzzy
3	$\chi(C) = R$	R:left quasi-reflexive	Fuzzy
4	$\varphi(\chi(C)) = C$	C:normal	Fuzzy
5	$\chi(\varphi(R)) = R$	R:left quasi-reflexive	Fuzzy

Table: Relation between choice and weak relations

Example 1: Lemma 1 - $Choice(C) = C$





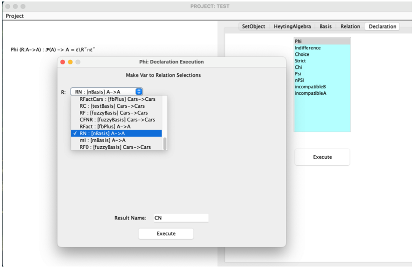
Relation C is a choice relation, i.e., the preferences of a user given by his choices of cars from a given set of cars in the primitive set object *Cars*.

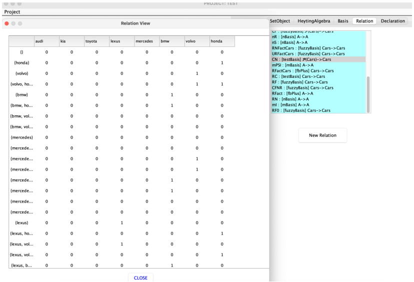
```xml
<HeytingAlgebra size="2">
    <Elements>
        0,1
    </Elements>
    <Meet>
        0,0,
        0,1
    </Meet>
    <Join>
        0,1,
        1,1
    </Join>
</HeytingAlgebra>
```

Listing 4.2: The *BooleanHA* xml

```xml
<Basis>
    <HeytingAlgebra name="BooleanHA"/>
    <Star>
        0,0,
        0,1
    </Star>
    <Plus>
        0,0,
        0,1
    </Plus>
</Basis>
```

Listing 4.1: The *testBasis* xml





Conversely, when we execute $\varphi(R)$ where R is left quasi-reflexive, the result is the original choice relation C. Hence $\varphi(R)$ is a normal choice relation confirming Lemma 2.

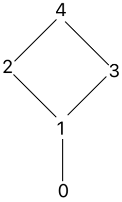
```xml
<Relation>
    <Basis name="testBasis"/>
    <SetObject Source="Cars" Target="Cars"/>
    <Matrix>
        1,0,1,0,0,1,0,1,
        0,1,1,1,0,0,1,1,
        1,0,1,0,1,1,0,1,
        0,1,0,1,1,0,1,0,
        1,0,1,0,0,1,1,0,
        0,1,0,1,1,1,0,1,
        1,0,1,0,1,0,1,1,
        0,1,1,1,0,0,1,1
    </Matrix>
</Relation>
```

Listing 4.3: The *RN* Relation xml

```xml
<Basis>
    <HeytingAlgebra name="BooleanHA"/>
    <Star>
        0,0,
        0,1
    </Star>
    <Plus>
        0,0,
        0,1
    </Plus>
</Basis>
```

Listing 4.1: The *testBasis* xml

EXAMPLE 2B: LEMMA 2 - $\varphi(R) = C$ - FUZZY




```xml
<Basis>
    <HeytingAlgebra name="fuzzyHA"/>
    <Star>
        0,0,0,0,0,
        0,1,1,1,1,
        0,1,2,2,2,
        0,1,2,3,3,
        0,1,2,3,4
    </Star>
    <Plus>
        0,1,2,3,4,
        1,1,2,3,4,
        2,2,2,3,4,
        3,3,3,3,4,
        4,4,4,4,4
    </Plus>
</Basis>
```

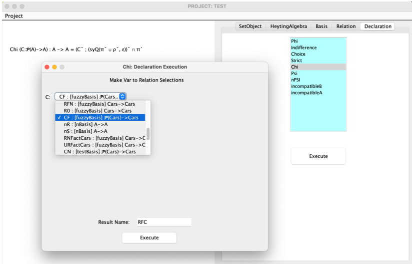
Listing 4.5: The *fuzzyBasis* xml

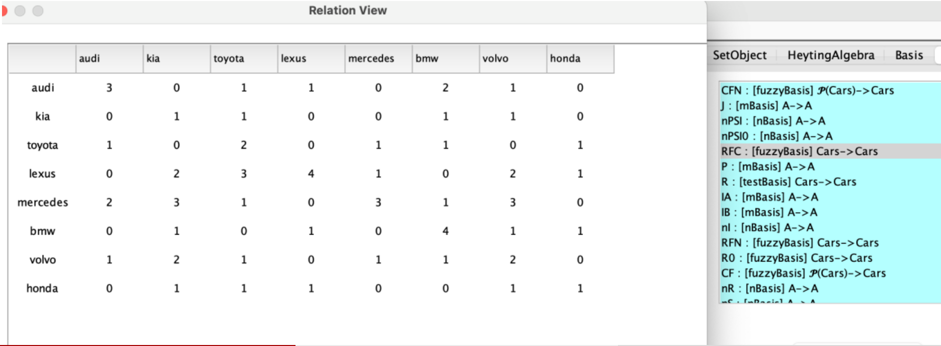
The result of $\varphi(RF)$ is CF and when *Choice* is executed with CF, we get back CF which also validates Lemma 2 in the fuzzy context.

```xml
<Relation>
    <Basis name="fuzzyBasis"/>
    <SetObject Source="Cars" Target="Cars"/>
    <Matrix>
        3,0,1,1,0,2,1,0,
        0,1,1,0,0,1,1,0,
        1,0,2,0,1,1,0,1,
        0,2,3,4,1,0,2,1,
        2,3,1,0,3,1,3,0,
        0,1,0,1,0,4,1,1,
        1,2,1,0,1,1,2,0,
        0,1,1,1,0,0,1,1
    </Matrix>
</Relation>
```

Listing 4.7: The *RF* relation

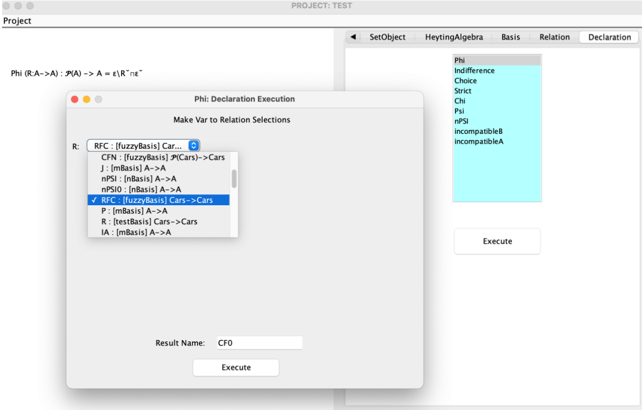
Example 3: Lemma 3 - $\chi(C) = R$ - Fuzzy





	audi	kia	toyota	lexus	mercedes	bmw	volvo	honda
audi	3	0	1	1	0	2	1	0
kia	0	1	1	0	0	1	1	0
toyota	1	0	2	0	1	1	0	1
lexus	0	2	3	4	1	0	2	1
mercedes	2	3	1	0	3	1	3	0
bmw	0	1	0	1	0	4	1	1
volvo	1	2	1	0	1	1	2	0
honda	0	1	1	1	0	0	1	1

Example 4: Lemma 4 -$\varphi(\chi(C)) = C$ - Fuzzy



Lemma 5 - $\chi(\varphi(R)) = R$

We prove Lemma 5 by applying χ to the relation $CF0$, which is obtained by $\varphi(RFC)$. The resulting relation, $RF1$, is left quasi-reflexive and identical to RFC.

Summary: Relation between Choice and Weak Relations

The examples show that establishing a weak preference from a choice, or vice versa, is achievable through the χ and φ operations, provided that conditions like left quasi-reflexivity and normality are met. This highlights the interdependence of these relations.

A pair (P, I) of two relations P and I on a set A is called a preference structure if P is $*$-asymmetric and I is symmetric. We consider the following definitions and lemmas [3] to establish the relation between a weak relation $R : A \to A$ and preference structures (P, I).

Definition

5. (P, I) is compatible iff $(P + I) - I = P$, and $(P \sqcap P^{\smile}) + I = I$
6. $R : A \to A$ is factorisable (with respect to $+$) iff $(R - R^{\smile}) + (R \sqcap R^{\smile}) = R$
7. $\psi(P, I) = P + I$
8. $\omega(R) = (R - R^{\smile}, R \sqcap R^{\smile})$.

Weak Relation and Preference Structures II

Lemma

6. (P, I) *is compatible if* $(P + I) - I = P$
7. *if Lemma 6 holds, then* $\psi(P, I)$ *is factorisable*
8. $\omega(R) = (R - R^{\smile}, R \sqcap R^{\smile})$*, if Lemma 7 holds then* $\omega(R)$ *is compatible*
9. R *is factorisable iff* $\psi(\omega(R)) = R$
10. (P, I) *is compatible iff* $\omega(\psi(P, I)) = (P, I)$

Example 1: Lemma 6 and 7 I

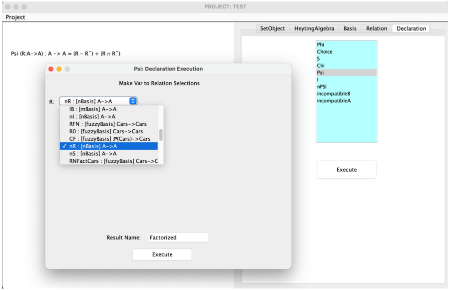
For clarity, P and S are interchangeably used, while I remains unchanged. We define $P = R - R^{\smile}$ and $I = R \sqcap R^{\smile}$, aiming to prove $(R - R^{\smile}) + (R \sqcap R^{\smile}) = R$. We introduce the relation nR, defined using

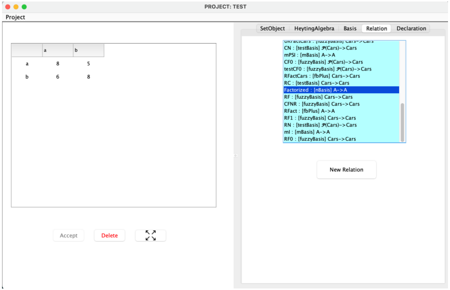
nBasis and *nAlgebra*, within a lattice structure $0, \dots, 8$ representing all subsets of $\{a, b, c\}$ where 0 is least element, $1 = \{\}$, $2 = \{a\}$, $3 = \{b\}$, $4 = \{c\}$, $5 = \{a, b\}$, $6 = \{a, c\}$, $7 = \{b, c\}$, and $8 = \{a, b, c\}$. Through matrix computation, Lemma 6 and 7 are validated.

$$(S + I) - I = \begin{bmatrix} 0 & \{b\} \\ \{c\} & 0 \end{bmatrix} + \begin{bmatrix} \{a,b,c\} & \{a\} \\ \{a\} & \{a,b,c\} \end{bmatrix} - \begin{bmatrix} \{a,b,c\} & \{a\} \\ \{a\} & \{a,b,c\} \end{bmatrix} = \begin{bmatrix} 0 & \{b\} \\ \{c\} & 0 \end{bmatrix} = P$$

$$S + I = \begin{bmatrix} 0 & \{b\} \\ \{c\} & 0 \end{bmatrix} + \begin{bmatrix} \{a, b, c\} & \{a\} \\ \{a\} & \{a, b, c\} \end{bmatrix} = \begin{bmatrix} \{a, b, c\} & \{a, b\} \\ \{a, c\} & \{a, b, c\} \end{bmatrix} = nR$$

Example 1: Lemma 6 and 7 II

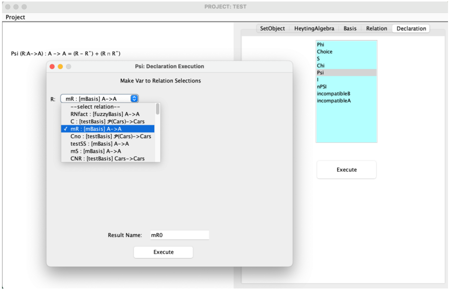


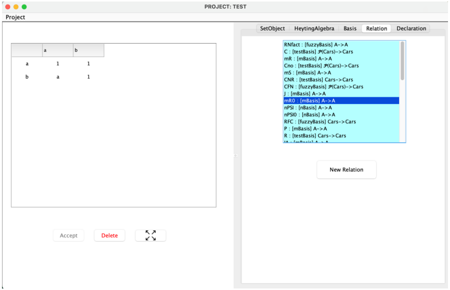


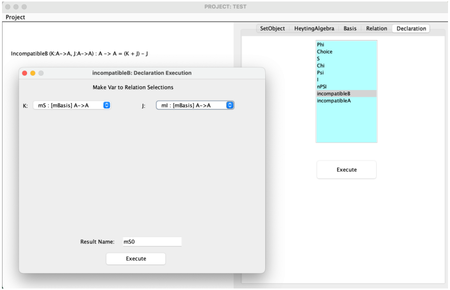

```
<Relation>
    <Basis name="nBasis"/>
    <SetObject Source="A" Target="A"/>
    <Matrix>
        8,5,
        6,8
    </Matrix>
</Relation>
```

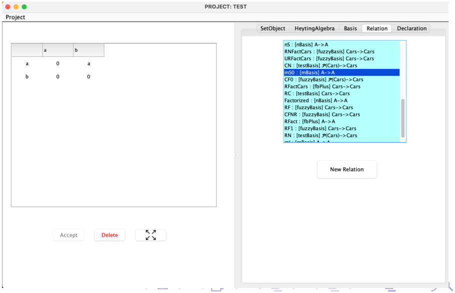
```
<HeytingAlgebra size="9">
    <Elements>
        0,1,2,3,4,5,6,7,8
    </Elements>
    <Meet>
        0,0,0,0,0,0,0,0,0
        0,1,1,1,1,1,1,1,1
        0,1,2,1,1,2,2,1,2
```

Counter Example 1: Lemma 6 and 7









Inferring Lemma 8, 9, and 10

Lemma 8, $\omega(R) = (R - R^{\smile}, R \sqcap R^{\smile})$, asserts that $\omega(R)$ forms a preference structure and is compatible if R is factorizable. This is confirmed in the previous examples, where $\psi(P, I)$ is proven to be compatible and R is shown to be factorisable.

Consequently, Lemma 9 is validated, stating that R is factorizable if $\psi(\omega(R)) = R$. Additionally, it follows that $\omega(\psi(P, I)) = (P, I)$, confirming Lemma 10, which asserts that (P, I) is compatible if $\omega(\psi(P, I)) = (P, I)$.

Summary: Relation Between Weak and Preference Relations

In summary, we show that an incompatible preference structure results in a relation that is not factorisable under the operation ψ. Conversely, if ω of a relation is not factorisable within a preference structure, then the structure is incompatible.

Unifying the Inter-relatedness of the 3 Relations

Lemma

If (P, I) is a compatible preference structure, then the relation $P + I$ is left quasi-reflexive iff (P, I) is a left quasi-reflexive preference structure.

This proof is demonstrated in examples of Lemma 6 and 7, where it is shown that $P + I = R$ and R exhibits left quasi-reflexivity.

Conclusions

The functions ψ and ω establish a bijection between the set of compatible, left quasi-reflexive preference structures and the set of factorisable, left quasi-reflexive relations.

The functions φ and χ establish a bijection between the set of factorisable, normal choice relations and the set of factorisable, left quasi-reflexive relations.

Visualization Environment: Tools

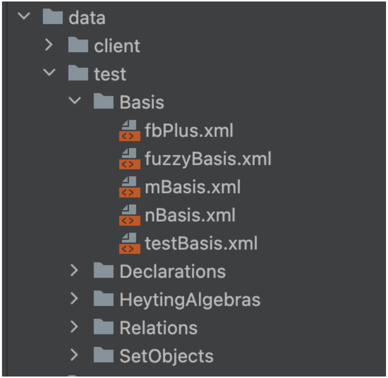
The **Java** core language is used to implement functions representing various fuzzy relations and operations. The **Java Swing** library provides an interactive user interface for visualizing the system, allowing users to interact, select sets, and perform set operations. Additionally, the **JParsec** and **Java.XML** libraries are employed for parsing expressions and processing XML files.

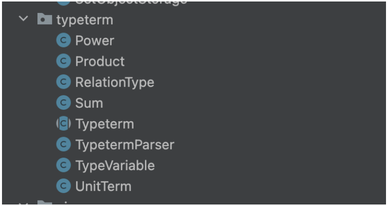
Goal

The bottom line of the visualization is that the designed system is able to handle finite L-fuzzy relations.

Visualization Environment: Implementation I

1. **Packages**: A package organizes source code by grouping related classes or files into a directory, with each package having a unique name to prevent conflicts. The *data* package is what houses all user-operated xml files.





Visualization Environment: Implementation II

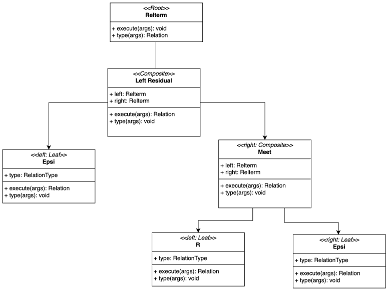
2. **Classes**:In Java, a class is a blueprint for creating objects. It defines an object's properties (fields) and actions (methods), encapsulating data and behaviors to model real-world entities in a program. Some important classes are the Relation, Declaration & DeclarationParser, Relterm & ReltermParser, Typeterm & TypetermParser, SetObject & Storage classes, Exceptions, and UserInterface classes.

```java
public class Relation extends Relationals {
private final SetObject source;
private final SetObject target;
private final Basis truth;
private final int[][] matrix;
private final Typeterm sourceTerm;
private final Typeterm targetTerm;
private final Map<String,SetObject> params;

private Relation(Typeterm sourceTerm, Typeterm targetTerm,
SetObject source, SetObject target, Map<String, SetObject> params
, Basis truth, int[][] matrix) {
    this.sourceTerm = sourceTerm;
    this.targetTerm = targetTerm;
    this.params = params;
    this.truth = truth;
    this.matrix = matrix;
    this.source = source;
    this.target = target;
}

public Relation(Typeterm sourceTerm, Typeterm targetTerm, Map<
String, SetObject> params, Basis truth, int[][] matrix) {
    this(sourceTerm,targetTerm,sourceTerm.execute(params, truth)
,targetTerm.execute(params, truth),params,truth,matrix);
}
```

The Relterm and Typeterm classes are created using the composite design pattern. The relterm expression $\varepsilon \backslash R \sqcap \varepsilon$, is represented in the image below.

Composite Design Pattern

It is a structural design pattern that enables treating tree-structured objects as if they were individual objects.



Visualization Environment: Implementation IV

3. **XML and XSD Files**: *.xml* is the file format in which we store data we work with in the system. *.xsd* are schema files that define the rules for how the xml files are structured. XMLReader classes are used to load the xmls files into their respective storages.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xs:element name="Basis">
     <xs:complexType>
         <xs:sequence>
             <xs:element name="HeytingAlgebra" minOccurs="1" maxOccurs="1">
                 <xs:complexType>
                     <xs:attribute name="name" type="xs:string"/>
                 </xs:complexType>
             </xs:element>
             <xs:element name="Star" minOccurs="1" maxOccurs="1"/>
         </xs:sequence>
     </xs:complexType>
 </xs:element>
</xs:schema>
```

```xml
<Basis>
    <HeytingAlgebra name="BooleanHA"/>
    <Star>
        0,0,
        0,1
    </Star>
</Basis>
```

```xml
<PrimitiveSetObject>
    audi,kia,toyota,lexus,mercedes,bmw,volvo,honda
</PrimitiveSetObject>
```

Our work establishes a framework for applying L-fuzzy relations in social choice theory and offers approaches for incorporating uncertainty. We hope future research will explore Arrow's Impossibility Theorem.

Arrow's Impossibility Theorem

Kenneth Arrow's theorem emphasizes that there exists no perfect way of aggregating individual preferences into a group decision, i.e. no system can be designed to satisfy a set of desirable preferences at the same time.

REFERENCES

1. Bengt Hansson. Choice structures and preference relations. Synthese, 18(4):443–458, 1968
2. Roland Gluck, Luigi Santocanale, and Michael Winter. Relational and Algebraic Methods in Computer Science: 20th International Conference, RAMiCS 2023, Augsburg, Germany, April 3–6, 2023, Proceedings. LNCS 13896. Springer Nature, 2023.
3. Frimpong Osei and Michael Winter. L-fuzzy weak preference, preference, and choice relations. In 21st International Conference on Relational and Algebraic Methods in Computer Science (RAMiCS 21), LNCS 14787. Springer, 2024
4. Bill Joy, Guy Steele, James Gosling, and Gilad Bracha. The java language specification, 2000
5. Project Repository: `https://github.com/frimps-astro/FuzzyProject`

THANK YOU

Link to Repository

Figure: Link to the designed model's repository