2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Facilitating Investment Strategy Negotiations through Logic

1st Benjamin Callewaert
KU Leuven, De Nayer Campus, Belgium,
Leuven.AI, Dept. of Computer Science,
Flanders Make – DTAI-FET
benjamin.callewaert@kuleuven.be

2nd Nicholas Decleyre
Intelli-Select
Belgium
nicholas.decleyre@intelli-select.com

3rd Simon Vandevelde
KU Leuven, De Nayer Campus, Belgium,
Leuven.AI, Dept. of Computer Science,
Flanders Make – DTAI-FET
s.vandevelde@kuleuven.be

4rd Nuno Comenda
Intelli-Select
Belgium
nuno.comenda@intelli-select.com

5th Bart Coppens
Intelli-Select
Belgium
bart.jan.coppens@intelli-select.com

6th Joost Vennekens
KU Leuven, De Nayer Campus, Belgium,
Leuven.AI, Dept. of Computer Science,
Flanders Make – DTAI-FET
joost.vennekens@kuleuven.be

***Abstract*—In the process of negotiating investment strategies between a fund and investors, establishing trust, transparency, traceability, and correctness among the involved parties is crucial to ensure smooth and successful outcomes. The adoption of logic-based AI, with its reliability, consistency, and explainability, can serve as a crucial catalyst to assist parties during negotiations by providing useful insights and explainable suggestions. This paper showcases how various Knowledge Representation and Reasoning (KRR) techniques can be leveraged to assist financial parties during investment negotiations. It demonstrates the use of logical definitions to represent complex financial investment strategies, allowing parties to gain a comprehensive understanding of the policies under discussion. Furthermore, automated reasoning is used to generate useful insights and actionable information enabling informed decision-making and enhancing the overall negotiation process.**

***Index Terms*—Fintech, Symbolic AI, KRR**

I. Introduction

The models learned by data-based AI are often unintelligible to humans. This poses a barrier to explainability and transparency, which may hinder the adoption and implementation of these models, particularly in highly regulated and risk-averse industries like Finance. Explainable Artificial Intelligence (XAI) methods aim to overcome these hurdles by providing explanations that render the inner workings of AI models interpretable and comprehensible.

Symbolic AI is a sub-field of AI that is explainable by design since it focuses on the high-level symbolic, human-readable representation of problems, logic, and search. The use of symbolic AI and automated reasoning techniques has proven throughout the years to be a great facilitator of transparency and trust in several domains including finance [7], [8], [12].

In this paper, we examine the use of logic for the financial activity of negotiating the investment strategy between a fund and an investor, a complex and time-consuming endeavor. Often, the fund and the investors adhere to different investment strategies, requiring them to engage in a series of negotiations to establish a mutually acceptable approach. These discussions involve careful consideration and analysis of each party's investment policies and objectives. In order to align their interests and reach a consensus, both the fund manager and the investors may need to make certain concessions by relaxing some of their respective policies. Once agreed upon, the fund receives a *discretionary mandate* to invest according to the negotiated policies. This mandate empowers the fund to make investment decisions, leveraging its expertise and insights to optimize returns while adhering to the negotiated strategy. Establishing trust, transparency, traceability, and correctness among the involved parties is crucial to ensure smooth and successful outcomes. To achieve this, symbolic AI, due to its above-mentioned characteristics, can serve as the catalyst to assist parties during negotiations by proving useful insights and actionable suggestions.

In this paper, we showcase how several Knowledge Representation and Reasoning (KRR) techniques can be leveraged to assist financial parties during negotiations about their investment strategies. We start by establishing the KRR systems and techniques used in this paper in Section II. Next, we showcase how financial investment strategies can be represented in an intuitive and understandable way using logical definitions in Section III. In Section IV, we demonstrate how automated reasoning techniques can be used to derive useful insights and suggestions for the parties during negotiations. Finally, we highlight potential areas for future research and conclude in Section V.

II. FO(.) & IDP

The **IDP system** [4] is a reasoning engine for FO(·), a rich extension of First-Order Logic (FOL). It implements the philosophy of the Knowledge Base Paradigm [5]: knowledge is represented in a purely declarative manner, independent from how it is used. This is done by storing the knowledge in a

Knowledge Base (KB), to which then various inference tasks can be applied to put it to practical use. This approach has two main advantages. Firstly, declaratively representing the knowledge is often easier than developing algorithmic solutions to specific tasks. Secondly, the split between knowledge and its application facilitates the re-use of the knowledge for multiple purposes. In this way, different problems in the same domain can typically be solved using the same KB with different inference tasks.

FO($\cdot$) extends FOL with types, aggregates, (inductive) definitions, arithmetic, partial functions, and intensional objects. In this way, FO($\cdot$) is an expressive and versatile representation language, well-suited for modeling problems from many domains.

Definitions are a very useful form of knowledge: they specify a unique interpretation of a defined symbol, given an interpretation of its parameters. Definitions are often formulated in natural language as a set of "rules" specifying necessary and sufficient conditions for the defined concept to hold.

The KB itself consists of three types of blocks: *vocabularies*, *structures* and *theories*, each representing the corresponding concept from classical logic.

A *vocabulary* specifies a set of *type*, *predicate*, or *function* symbols. A type is a domain of values, such as a list of strings or the domain of real numbers $\mathbb{R}$. A predicate symbol expresses a relation on zero or more types. A proposition is a 0-ary predicate. Lastly, a function symbol expresses a function from the Cartesian product of a number of types $T_1 \times \cdots \times T_n$ to a type T_{n+1}. A function is also called a *constant* if it is 0-ary, i.e., has no input arguments.

A *structure* provides an interpretation for the symbols in its vocabulary. If it provides an interpretation for each symbol in the vocabulary, it is called a *full* interpretation. Otherwise, it is called a partial interpretation.

A *theory* contains a set of logical formulas, written in FO($\cdot$).

By itself, the KB is not executable: it merely represents the knowledge of a domain. To put this knowledge to use, the IDP system offers multiple inference tasks, like propagation and model expansion.

We will briefly go over those tasks relevant to this work. *Model expansion* will, given a partial interpretation $\mathcal{I}$ for the vocabulary of a theory $\mathcal{T}$, expand this interpretation to a full interpretation I that satisfies the theory ($I \models T$). In order to find a model expansion with the lowest/highest value for a specific term, the *optimization* inference is used. *Propagation* derives the consequences of a partial interpretation $\mathcal{I}$ given a theory and its vocabulary. This results in a set of facts that hold in all model expansions of $\mathcal{I}$. Given a theory T and partial interpretation $\mathcal{I}$, the *abstract model generation* (AMG) inference searches for a set C of simple constraints that imply the theory, i.e., such that for all I that extend $\mathcal{I}$, $I \models C \rightarrow T$. Lastly, the *explanation* inference is used to find a minimal explanation for why a structure $\mathcal{I}$ does not satisfy the theory T ($\mathcal{I} \not\models T$). Here, this explanation is a minimal subset of symbol assignments that cause unsatisfiability.

In the past, the IDP system has already proven itself as a suitable tool for several applications [1], [2], [9], [10], [11]. Its approach works well to tackle complex configuration problems in an interactive way. Such applications can be found in many domains, such as manufacturing, finance, and logistics. The latest version of the IDP system, and the one used in this work, is IDP-Z3 [3].

III. Investment strategy description

Usually, during investment negotiations, involved parties do not consider specific financial assets. Instead, they talk about groups of assets that adhere to several complex criteria. These criteria are captured in an investment strategy, which can be seen as a list of rules the assets need to comply with. Traditionally, financial experts are responsible for formalizing these strategies by translating various requirements into lengthy, hard-to-understand programs that contain a lot of enumerations, repetitions, complex nesting of conditional clauses, and exceptions that need to be followed in the right order. This approach lacks transparency, which is troublesome because a full understanding of the discussed strategies is crucial to achieving favorable results during negotiations.

Therefore, we propose to represent an investment strategy through logical definitions in FO($\cdot$). By establishing a common strategy description method, we allow parties to get a full understanding of each other's policies and allow for automated analysis of the strategies.

To represent investment strategies in IDP, we need a way to describe their base elements: financial assets. We do this by creating an IDP vocabulary with a general type *Asset*, which enumerates the identifiers of the possible assets, and the types (or domains) of each of the possible asset attributes, such as security type, country, ... Additionally, a function is added for each attribute, which maps each identifier to its corresponding attribute value. To logically represent a financial asset, it now suffices to assign a value to each of the functions. A simplified example of such a vocabulary is given in Listing 1.

```
vocabulary V {
  type Asset := {asset_0001, asset_0002, asset_0003}
  type Security_Type := {bond, equity, fund}
  type Country := {belgium, france, luxembourg}
  type Rating := {aaa, aa, a, bbb, bb, b}
  type Industry := {automotive, banking, chemicals}
  type Currency := {eur, usd, gbp, jpy}
  asset_security_type: Asset → Security_Type
  asset_country: Asset → Country
  asset_rating: Asset → Rating
  asset_industry: Asset → Industry
  asset_currency: Asset → Currency
}
```

Listing 1: IDP vocabulary - financial assets

This representation of individual assets does not yet allow us to describe investment strategies. Indeed, a strategy could be seen as a specific subset of the asset space, more specifically those assets that a party might want to invest in. In logical terms, each strategy corresponds to the definition of a class of acceptable assets. If we introduce a predicate $Class(Asset)$ to denote that $Asset$ is a member of the $Class$, we can include a definition of this predicate into our theory and use this to check whether an asset is part of the investment strategy or not (binary classification). Since different parties have different strategies, we can also include multiple classes $Class_1, Class_2, \ldots$, each with its own definition, as shown in Listing 2.

$$\left\{\begin{array}{l} \forall a \text{ in } Asset : \text{Class}_1(a) \leftarrow \ \phi_1. \\ \forall a \text{ in } Asset : \text{Class}_2(a) \leftarrow \ \phi_2 \land \neg\text{Class}_1(a). \\ \forall a \text{ in } Asset : \text{Class}_3(a) \leftarrow \ \phi_3 \lor \text{Class}_1(a). \end{array}\right\}$$

Listing 2: Generalized example definition rules

In the definitions of Listing 2, ϕ_1, ϕ_2 and ϕ_3 represent combinations of the asset attributes, using the logical operators $\land$ (and), $\lor$ (or) and $\neg$ (not). In addition to attributes, it is possible to use class predicates in the bodies of other definition rules, as can be seen in the second and third rules. An important scenario to take into account is when an asset does not belong to any of the classes. To capture those assets, we can add a "default"-definition that applies when none of the other definitions do or collect them with a dedicated post-processing step.

In the following sections, the "strategy membership" classification will be used to describe investment strategies. In this setting, we define whether a financial asset is a member of a strategy. To define this, we introduce a predicate *Member* and a predicate *NotMember*. In this way, the investor or fund gains the ability to explicitly determine whether an asset definitely falls within its strategy or definitively does not. To keep these subsets disjoint, not being a member of the strategy takes priority over being a member, as can be seen in the definition of Listing 3. Note that this setting implies that some assets are not classified as either part or not part of the strategy. By employing this approach, clients can express their strategy in a more intuitive manner by specifying the assets they would definitely invest in and those they would definitely avoid. The assets that are not classified can then serve as a means to suggest potential relaxations to the strategies during negotiations.

$$\left\{\begin{array}{l} \forall a \text{ in } Asset : \text{Member}(a) \leftarrow \ \phi_1 \land \neg\text{NotMember}(a). \\ \forall a \text{ in } Asset : \text{Member}(a) \leftarrow \ \phi_2 \land \neg\text{NotMember}(a). \\ \forall a \text{ in } Asset : \text{NotMember}(a) \leftarrow \ \phi_3. \end{array}\right\}$$

Listing 3: Generalized example strategy membership use case

To assist parties in creating these strategy definitions and to overcome the syntactic hurdle of FO($\cdot$), a Natural Language interface was implemented in earlier work [6]. This interface allows parties to define their investment strategy by means of controlled natural language (CNL), employing a step-by-step selection of building blocks for sentence construction. The resulting structured sentence is automatically translated into FO($\cdot$). The interface also integrates a deep learning NLP module that handles free-form English, suggesting three likely CNL statements for the user to choose and adjust if necessary.

To showcase how our approach can facilitate negotiations, we will employ a running example involving an investor and a fund with different investment strategies. Suppose the fund adopts the following investment strategy:

> "Our fund focuses on investments in bonds with a top-tier AAA rating. Our fund also invests in equity in the banking sector. Notably, our fund does not invest in assets denominated in either the euro or the British pound."

And the investor pursues the following strategy:

> "We invest in bonds with a rating of AAA or AA. We also invest in equity denominated in dollars. Importantly, our investment approach excludes assets originating from Belgium or France."

These strategies can then be translated using the NL interface into the FO($\cdot$), as shown in Listing 4 and Listing 5, respectively.

$$\left\{\begin{array}{ll} \forall a \text{ in } Asset : & \text{Member}(a) \leftarrow \text{Type}(a) = \text{bond} \land \\ & \text{Rating}(a) = \text{aaa} \ \land \text{NotMember}(a). \\ \forall a \text{ in } Asset : & \text{Member}(a) \leftarrow \text{Type}(a) = \text{equity} \land \\ & \text{Industry}(a) = \text{banking} \land \\ & \neg\text{NotMember}(a). \\ \forall a \text{ in } Asset : & \text{NotMember}(a) \leftarrow \\ & \text{Currency}(a) \text{ in } \{\text{eur, gbp}\}. \end{array}\right\}$$

Listing 4: Strategy of fund in FO($\cdot$) (Strategy_1)

$$\left\{\begin{array}{ll} \forall a \text{ in } Asset : & \text{Member}(a) \leftarrow \text{Type}(a) = \text{bond} \land \\ & \text{Rating}(a) \text{ in } \{\text{aaa, aa}\} \land \\ & \neg\text{NotMember}(a). \\ \forall a \text{ in } Asset : & \text{Member}(a) \leftarrow \text{Type}(a) = \text{equity} \land \\ & \text{Currency}(a) = \text{usd} \ \land \neg\text{NotMember}(a). \\ \forall a \text{ in } Asset : & \text{Member}(a) \leftarrow \text{Type}(a) = \text{fund} \land \\ & \neg\text{NotMember}(a). \\ \forall a \text{ in } Asset : & \text{NotMember}(a) \leftarrow \text{Country}(a) \\ & \text{ in } \{\text{belgium, france}\}. \end{array}\right\}$$

Listing 5: Strategy of investor in FO($\cdot$) (Strategy_2)

IV. Generating insights

With logical representations of both financial assets and strategies in place, these can now be used to generate practical insights. In this work, we consider three types of insights: strategy membership validation, strategy gap analysis, and strategy alignment suggestions. For the first insight, the system checks if a financial asset falls under a given strategy. The second insight focuses on comparing two strategies and identifying logical gaps between them. Finally, in the third insight type, the system proposes which policies each party could relax in order to align the strategies more closely.

While it is possible to use more complex logical combinations in the FO($\cdot$) representation of a strategy, for the current use case, it is assumed that the bodies of the definition rules are conjunctions of (negated) predicates.

Note that, it is possible to convert any formula in first-order logic to *Disjunctive Normal Form* (DNF). Since every rule with a disjunction as a body can be written as individual rules, you can thus convert every rule to a set of rules with only conjunctions in the body. Therefore it is always possible to achieve the required disjunct definition form.

$$\{\forall a \text{ in } Asset : \mathrm{Class}(a) \leftarrow\ \phi_1 \vee\ \phi_2 \vee\ \phi_3.\}$$
$$\updownarrow$$
$$\left\{\begin{array}{l} \forall a \text{ in } Asset : \mathrm{Class}(a) \leftarrow\ \phi_1. \\ \forall a \text{ in } Asset : \mathrm{Class}(a) \leftarrow\ \phi_2. \\ \forall a \text{ in } Asset : \mathrm{Class}(a) \leftarrow\ \phi_3. \end{array}\right\}$$

Listing 6: DNF to separate rules

A. Strategy membership validation

If the fund or investor already possesses financial assets, one may want to identify which of these assets align with the strategy of the other. This can be done by constructing a structure for the vocabulary shown in Listing 1, that represents the given set of assets. If we then run the *model propagation* inference task, the system derives all those facts that are certainly true or certainly false for these assets. The result shows whether the asset is definitely a *member* of the defined strategy or not.

In the strategy of the fund described in section III, AAA bonds and equities active in banking are part of the strategy, as long as their currency is not euro or pound sterling. If the asset is not identified as either a member or not a member, then its status is set to “unknown” in a post-processing step.

After classifying the asset as (not) being a member of the strategy, the next step is to find out why this is the case, i.e. what attribute combinations make the *Member*- or *NotMember*-predicate true. This can be done in a straightforward manner by replacing the “general” class predicates in the definition with rule-specific ones and introducing an auxiliary rule combining the rule-specific predicates to define the general *Member* and *NotMember* class. Listing 7 shows this for the fund’s strategy description in Listing 4.

When the *model propagation* inference task is executed using this augmented theory, the system will still specify if the asset is a member of the defined strategy. In addition, it will show which rule-specific predicates are also certainly true and thus which rule and combination(s) of attributes made the asset (not) a part of the overall strategy

Fig. 1 shows an example of how the system gives feedback about the asset membership to the user. Membership itself is indicated by a green tick or a red cross and an explanation is provided for why the asset is (not) a member of the strategy.

$$\left\{\begin{array}{ll} \forall a \text{ in } Asset : & \mathrm{Member}_{rule1}(a) \leftarrow \mathrm{Type}(a) = \text{bond} \wedge \\ & \mathrm{Rating}(a) = \text{aaa} \ \wedge \neg\mathrm{NotMember}(a). \\ \forall a \text{ in } Asset : & \mathrm{Member}_{rule2}(a) \leftarrow \mathrm{Type}(a) = \text{equity} \wedge \\ & \mathrm{Industry}(a) = \text{banking} \wedge \\ & \neg\mathrm{NotMember}(a). \\ \forall a \text{ in } Asset : & \mathrm{NotMember}_{rule3}(a) \leftarrow \\ & \mathrm{Currency}(a) \text{in } \{\text{eur, gbp}\}. \\ \forall a \text{ in } Asset : & \mathrm{Member}(a) \leftarrow \mathrm{Member}_{rule1}(a) \vee \\ & \mathrm{Member}_{rule2}(a). \\ \forall a \text{ in } Asset : & \mathrm{NotMember}(a) \leftarrow \mathrm{NotMember}_{rule3}(a). \end{array}\right\}$$

Listing 7: Strategy membership validation - Explanations

	Identifier	Security Type	Currency	Rating	Industry	Country
✗	SEC001	Fund	AED	A-1+	Industrial	
✗	SEC002	Equity	BGN	B+	Food & Beverage	Barbados
✓	SEC003	Bond	BHD	AAA	Automotive	Argentina

? All bonds with rating AAA are a member of the investment strategy

Fig. 1: Feedback to the user

B. Strategy comparison

To gain a deeper insight into the other party’s strategy, it can be helpful to compare it with your own strategy. By highlighting where the strategies differ or overlap, parties can improve their decision-making during the negotiation process. In section III, we stated that the body of a definition rule is a logical combination of asset attributes and class predicates. When the body of the definition is limited exclusively to class predicates, it is possible to define classes of assets that represent the outcomes of logical set operations. Some simple examples are shown below in Listing 8 and visually represented in Fig. 2.

$$\left\{\begin{array}{ll} \forall a \text{ in } Asset : & \mathrm{Intersection}_{12}(a) \leftarrow \mathrm{Class}_1(a) \wedge \\ & \mathrm{Class}_2(a). \\ \forall a \text{ in } Asset : & \mathrm{Union}_{12}(a) \leftarrow \mathrm{Class}_1(a) \vee \mathrm{Class}_2(a). \\ \forall a \text{ in } Asset : & \mathrm{Difference}_{1-2}(a) \leftarrow \mathrm{Class}_1(a) \wedge \\ & \neg\mathrm{Class}_2(a). \\ \forall a \text{ in } Asset : & \mathrm{Difference}_{2-1}(a) \leftarrow \mathrm{Class}_2(a) \wedge \\ & \neg\mathrm{Class}_1(a). \end{array}\right\}$$

Listing 8: Example of combined definitions

By applying these different set operations to the strategy definitions of the investors and the fund, we can logically compare their strategies and provide useful insights. In this work, the focus lies on finding the differences and commonalities between strategies. Therefore only the difference and intersection operations are considered. The definition rules defining the resulting classes are respectively called *difference rules* and *overlap rules*.

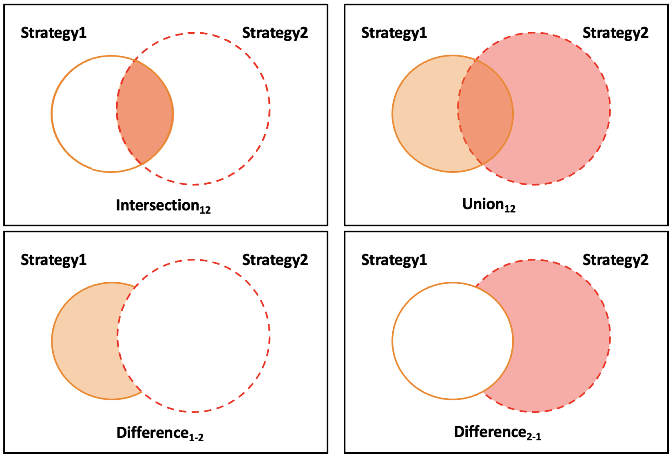

Fig. 2: Visualization of set operations on two strategies

To describe these newly defined classes in terms of asset attributes instead of terms of other classes, an algorithm that makes use of IDP's *abstract model generation* inference task was developed. The algorithm produces a collection of disjoint definition rules, which serve as an abstract representation encompassing all assets that are a member of the newly defined class. This approach is especially interesting when we compare corresponding classes from different strategies. In practice, this means that we will look at what assets are a member of both strategies (overlap) or a member of only one strategy, but not of the other (difference). To illustrate this, Listing 10 shows the result of finding the difference between the fund's Strategy_1 and the investor's Strategy_2. Note that the *Member*-predicates get an index corresponding to the strategy they originate from.

$$\left\{ \begin{array}{l} \forall a \text{ in } Asset : \text{Difference}_{2-1}(a) \leftarrow \text{Member}_2(a) \land \\ \qquad\qquad\qquad \neg\text{Member}_1(a) \end{array} \right\}$$

$$\updownarrow$$

$$\left\{ \begin{array}{ll} \forall a \text{ in } Asset : & \text{Difference}_{2-1}(a) \leftarrow \text{Type}(a) = \text{bond} \land \\ & \text{Rating}(a) = \text{aa} \land \\ & \text{Country}(a) \text{ not in } \{\text{belgium, france}\} \land \\ & \text{Currency}(a) \text{ not in } \{\text{eur, gbp}\}. \\ \forall a \text{ in } Asset : & \text{Difference}_{2-1}(a) \leftarrow \text{Type}(a) = \text{equity} \land \\ & \text{Industry}(a) \neq \text{banking} \land \\ & \text{Currency}(a) = \text{usd} \land \\ & \text{Country}(a) \text{ not in } \{\text{belgium, france}\}. \\ \forall a \text{ in } Asset : & \text{Difference}_{2-1}(a) \leftarrow \text{Type}(a) = \text{fund} \land \\ & \text{Country}(a) \text{ not in } \{\text{belgium, france}\} \land \\ & \text{Currency}(a) \text{ not in } \{\text{eur, gbp}\}. \end{array} \right.$$

Listing 10: Difference between 2 strategies

More concretely, finding the differences or overlaps between two classes comes down to comparing their definition rules in a pair-wise manner. Sometimes, although there exists an overlap between two rules, it might be preferred not to take it into account. This is usually driven by contextual knowledge

$$\left\{ \begin{array}{ll} \forall a \text{ in } Asset : & \text{Member}_1(a) \leftarrow \text{Type}(a) = \text{fund} \land \\ & \text{RuleGroup}(a) = \text{group1} \land \\ & \neg\text{NotMember}_1(a). \\ \forall a \text{ in } Asset : & \text{Member}_2(a) \leftarrow \text{Industry}(a) = \\ & \text{banking} \land \text{RuleGroup}(a) = \\ & \text{group2} \land \neg\text{NotMember}_2(a). \end{array} \right\}$$

Listing 11: Artificial removal overlap

of the use case or user preferences. For example, since funds do not have a designated industry, rules talking about funds should not be compared to those specifying an industry. To accomplish this, different rules are grouped in such a way that they are only compared to rules within their own group. This can be done by adding an artificial attribute to the rule, specifying the group it belongs to. If the value of this attribute is different for two rules, any existing overlap is nullified. Listing 11 shows such a case, where there would be overlap between both strategies if the *RuleGroup*-attribute would not be present. Using these newly defined classes and their definitions, one can reason about assets being part of the fund strategy, the investor strategy, or any logical combination of them. We could for example extend our first insight type to check whether a given asset is part of the difference between the two strategies.

C. Strategy alignment suggestions

Aligning the different investment strategies of the investor and the fund can be a complex and time-consuming process. In order to reach a common investment strategy, both the fund manager and the investors may need to make certain concessions by relaxing some of their respective policies. Once the differences between the two strategies have been identified, we can propose several logical relaxations that can be made to their respective strategy to align them more closely. Five different relaxations are considered to assist the parties during negotiations:

1) Adding a value to a specified attribute in a *Member*-rule
2) Removing an attribute from a *Member*-rule
3) Adding a difference rule as a *Member*-rule
4) Removing a value from a specified attribute in a *Not-Member*-rule
5) Removing a *NotMember*-rule that (partly) overlaps with a difference rule

In the following explanation, we propose how the fund's Strategy_1 (Listing 4) can be relaxed to reduce the assets that are a member of the investor's Strategy_2 (Listing 5) but not of Strategy_1. These relaxations are based on the result of the corresponding set operation: $Difference_{2-1}$ (Listing 10).

The first three relaxations try to extend the $Member$ definition of Strategy_1. The first relaxation is the result of an additional comparison between the $Difference_{2-1}$ and Strategy_1. By identifying the differing values for the same attribute, one could add those values to the specified attribute to relax Strategy_2. In this case, we suggest that the fund relaxes the first definition rule of their Strategy$_1$ by also allowing bonds with rating aa.

Next, the second relaxation is a generalization of the first one. Indeed, removing an attribute altogether logically comes down to accepting all values of the domain of that attribute. In this scenario, our proposal suggests removing the $Rating$ attribute from the first rule of the definition.

As a third option, one can consider adding one or multiple difference rules directly to the definition of Strategy$_2$. This is the most straightforward way of covering more assets by adding those that are not covered. We could thus propose to add the first difference rule to Strategy$_1$.

In addition to extending the definition of $Member$, we can also narrow the definition of Strategy$_1$'s $NotMember$ to relax policies. To accomplish this, we propose to remove certain values from the attribute assignments in the rules of the $NotMember$ definition. In our example, we propose to remove eur or gbp from the assignment of attribute $Currency$ in the $NotMember$ definition.

Finally, we could also do an "overlap"-comparison between each of the *NotMember*-rules of Strategy$_1$ and $Difference_{2-1}$ and find out which rules overlap with the difference between Strategy$_2$ and Strategy$_1$. We can then propose to revise or remove these rules altogether. Our proposal in this situation is to eliminate the $NotMember$ definition rule from the strategy. Note that we do not propose to introduce new attributes in the body of any $NotMember$ definition rule to narrow the definition since this does not make any sense.

Using these useful insights and logical relaxations, parties can greatly reduce the time needed to achieve a consensus on the investment strategy.

V. Conclusion & Future Work

In conclusion, this paper demonstrates the potential of leveraging logical AI techniques to facilitate investment strategy negotiations in the financial domain. By using logical definitions in FO(·), the paper showcases how investment strategies can be represented in an intuitive and understandable manner. By establishing a common strategy description method, parties get a better understanding of each other investment policies. The application of automated reasoning techniques enables the derivation of valuable insights and actionable suggestions for the parties involved in negotiations. The three types of insights presented, including strategy membership validation, strategy comparison, and strategy alignment suggestions, contribute to establishing trust, transparency, and correctness among the parties.

For future research, we will explore different ways to translate other investment strategy description methods into our proposed form. Furthermore, by incorporating user preferences into the approach, it becomes possible to prioritize specific investments within the strategy, thus introducing a mechanism for assigning greater importance to certain assets over others. Furthermore, scoring the alignment of two strategies would enable investors to search for funds with the most compatible investment strategies. These potential areas of future investigation aim to further enhance the overall success of negotiating investment strategies.

Ultimately the utilization of automated reasoning and knowledge representation and reasoning (KRR) techniques in facilitating investment strategy negotiations presents a promising avenue for further exploration. This approach has the potential to be extended to various other applications within the financial domain, including financial planning for private and wealth management, debt restructuring, valuations, and, to a certain extent, quantitative trading strategies.

VI. Acknowledgment

This research received funding from the Flemish Government, through Flanders Innovation & Entrepreneurship (VLAIO, project HBC.2022.0477).

References

[1] B. Aerts, M. Deryck, and J. Vennekens, "Knowledge-based decision support for machine component design: A case study," *Expert Systems with Applications*, vol. 187, p. 115869, Jan. 2022.

[2] P. Carbonnelle, B. Bogaerts, J. Vennekens, and M. Denecker, "Interactive Configuration Problems in Observable Environments," p. 8, 2022.

[3] P. Carbonnelle, S. Vandevelde, J. Vennekens, and M. Denecker, "Interactive configurator with fo(.) and idp-z3," 2023.

[4] B. De Cat, B. Bogaerts, M. Bruynooghe, G. Janssens, and M. Denecker, "Predicate logic as a modeling language: the IDP system," in *Declarative Logic Programming: Theory, Systems, and Applications*, M. Kifer and Y. A. Liu, Eds. ACM, Sep. 2018, pp. 279–323. [Online]. Available: https://dl.acm.org/citation.cfm?id=3191321

[5] M. Denecker and J. Vennekens, "Building a Knowledge Base System for an Integration of Logic Programming and Classical Logic," in *Logic Programming*, M. Garcia de la Banda and E. Pontelli, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 5366, pp. 71–76, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-540-89982-2-12

[6] M. Deryck, N. Comenda, B. Coppens, and J. Vennekens, "Combining logic and natural language processing to support investment management," in *Proceedings of the international conference on principles of knowledge representation and reasoning*, vol. 18, 2021, pp. 666–670.

[7] P. E. Hart, A. Barzilay, and R. O. Duda, "Qualitative reasoning for financial assessments: A prospectus," *AI magazine*, vol. 7, no. 1, pp. 62–62, 1986.

[8] L. R. Talluru and V. Akgiray, "Knowledge representation for investment strategy selection," in *[1988] Proceedings of the Twenty-First Annual Hawaii International Conference on System Sciences. Volume III: Decision Support and Knowledge Based Systems Track*, vol. 3. IEEE, 1988, pp. 189–196.

[9] P. Van Hertum, I. Dasseville, G. Janssens, and M. Denecker, "The KB paradigm and its application to interactive configuration," *Theory and Practice of Logic Programming*, vol. 17, no. 1, pp. 91–117, 2017.

[10] S. Vandevelde, B. Callewaert, and J. Vennekens, "Interactive feature modeling with background knowledge for validation and configuration," in *Proceedings of the 26th ACM International Systems and Software Product Line Conference-Volume B*, 2022, pp. 209–216.

[11] H. Vlaeminck, J. Vennekens, and M. Denecker, "A logical framework for configuration software," in *Proceedings of the 11th ACM SIGPLAN Conference on Principles and Practice of Declarative Programming*, ser. PPDP '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 141–148. [Online]. Available: https://doi.org/10.1145/1599410.1599428

[12] P. Weber, K. V. Carl, and O. Hinz, "Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature," *Management Review Quarterly*, pp. 1–41, 2023.