Journal of Universal Computer Science, vol. 26, no. 10 (2020), 1343-1363
submitted: 2/4/19, accepted: 24/10/20, appeared: 28/10/20

A Semantic Web-based Representation of Human-logical Inference for Solving Bongard Problems

Jisha Maniamma
(Kyushu Institute of Technology, Kitakyushu, Japan
jishapillai1991@gmail.com)

Hiroaki Wagatsuma
(Kyushu Institute of Technology, Kitakyushu, Japan
RIKEN CBS, Saitama, Japan
waga@brain.kyutech.ac.jp)

Abstract: Bongard Problems (BPs) are a set of 100 visual puzzles introduced by M. M. Bongard in the mid-1960s. BPs have been established as benchmark puzzles for understanding the human context-based learning abilities to solve ill- posed problems. The puzzle requires the logical explanation as the answer to distinct two classes of figures from redundant options, which can be obtained by a thinking process to alternatively change the target frame (hierarchical level of analogy) of thinking from a wide range concept networks as D. R. Hofstadter suggested. Some minor research results to solve a limited set of BPs have reported based a single architecture accompanied with probabilistic approaches; however the central problem on BP's difficulties is the requirement of flexible changes of the target frame, therefore non-hierarchical cluster analyses does not provide the essential solution and hierarchical probabilistic models needs to include unnecessary levels for learning from the beginning to prevent a prompt decision making. We hypothesized that logical reasoning process with limited numbers of meta-data descriptions realizes the sophisticated and prompt decision-making and the performance is validated by using BPs. In this study, a semantic web-based hierarchical model to solve BPs was proposed as the minimum and transparent system to mimic human-logical inference process in solving of BPs by using the Description Logic (DL) with assertions on concepts (TBox) and individuals (ABox). Our results demonstrated that the proposed model not only provided individual solutions as a BP solver, but also proved the correctness of Hofstadter's idea as the flexible frame with concept networks for BPs in our actual implementation, which no one has ever achieved. This fact will open the new horizon for theories for designing of logical reasoning systems especially for critical judgments and serious decision-making as expert humans do in a transparent and descriptive way of why they judged in that manner.

Key Words: Bongard Problem (BPs), Meta-data Ontology, Resource Descriptive Framework (RDF), Semantic Web Rule Language (SWRL), Web Ontology Language (OWL)
Category: H.2, H.3.7, H.5.4

1 Introduction

In recent years, computer vision and machine learning approaches have made significant progress by relying on a massive amount of digital data with annotations, or teacher signals [Johnson et al. 2017, Salameh et al. 2014, Zand et al. 2016], which sometime demonstrates a superior ability against human experts at games [Silver et al. 2016]. Such non-negligible developments in the field of the artificial intelligence even in comparison with human abilities led a science fiction-like prediction known as technological singularity [Kurzweil 2014]. On the other hand, as McCarthy and Hayes (1969) proved in the form of the frame problem [McCarthy and Hayes 1969], the machine has a limited power to classify necessary and unnecessary events with respect to the current context for making a decision at each short moment if the events are described as an infinite number of discrete and countable representations. Therefore, a key question of why the human intuition can avoid the frame problem and prevents the halt of thinking can be replaced by the question of the existence of the ability in the brain to change abstraction and analogy levels flexibly depending on the current context. A good example is the Steiner Tree Problem in graphs, which is a NP-hardness in computational complexity theory [Chopra and Rao 1994, Sun 2019]. For example, the problem asks us to 'find the point to minimize the summation of distances from every node of the rectangle' and the answer can be found easily as the crossing point of two lines connecting two most distant points individually, by introducing the concept of 'connecting lines' instead of finding a point from $R \times R$ infinitely [Arai et al. 2017]. The fact indicates that the human intelligence focuses on vital information that is crucial for current decision making by using an appropriate abduction with logic. Some researchers have tried to introduce the semantics in the form of annotations with classified data obtained by probabilistic approaches [Salameh et al. 2014, Zand et al. 2016]; however the practical combinations with semantics into the data analysis do not provide the right answer to our cognitive ability to set an appropriate level of abstraction [Chen et al. 2016, Pfeifer and Bongard 2007]. In the early stage, such a higher level cognitive function was modeled with long-term memory, planning, and logical decision-making abilities as conceptual models [Forbus et al. 1998, Mitchell 2003], which were far from an actual benchmark validation of how the fine logical reasoning process contributes to the function.

Interestingly, Bongard Problems (abbreviated as BPs) introduced by M. M. Bongard [Bongard 1970] as a set of 100 visual puzzles will be the benchmark test for understanding the flexible change of the analogy level fit to the given problem as D. R. Hofstadter suggested [Hofstadter 1979], which is considered as a NP-hardness similarly to the Steiner Tree Problem. In his book, he summarized that necessary functional components to solve BPs are i) concept network, ii) frames, iii) meta-descriptor, iv) filtering and v) sameness detector. Foundalis (2006) tried to formulate the components and verified with human subjects as solvers. There are recent works tackled with BPs [Saito and Nakano 1995, Linhares 2000, Weitnauer and Ritter 2012, Depeweg et al. 2018] with challenging frameworks individually, while those works are within case studies and the

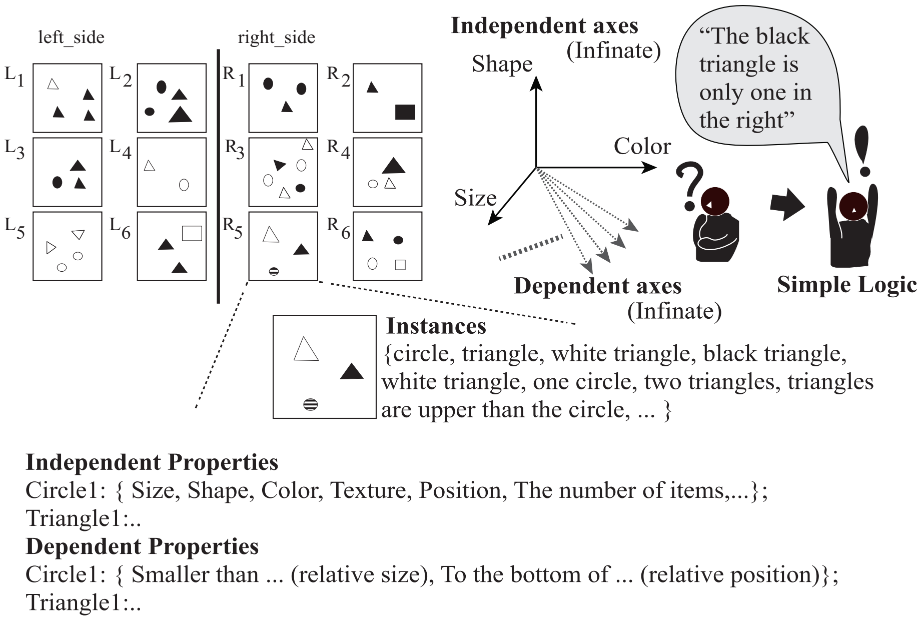

Figure 1: Complexity in solving BPs (each objects in a box has multiple independent and dependent properties, which increase exponentially with increases in the object count).

systematic solution still remain unsolved. The possible and plausible systematic solution is to prove whether Hofstadter's hypothesis truly right or not, by reconstructing the testable framework based on tools with recent advancements in the semantic web approach. More concretely, in the present paper, we clearly hypothesized that an appropriate combination given as the workflow of i) ontology-based description to represent the necessary concept network, ii) description logic (DL) with assertions on concepts (TBox) and individuals (ABox) to provide minimum frames, iii) meta-data template, iv) SPARQL Protocol and RDF Query Language shortly SPARQL for the filtering function and v) Semantic Web Rule Language (SWRL) rules as sameness detector.

The paper was organized with 5 sections. Section 2 provides an overview of the relevance of BPs and the related past research approaches, and Section 3 described a general perspective into knowledge representation using ontology accompanied with discussions on necessary system design. In Section 4, we showed results of computer experiments of our proposed system, which solved 62 BPs using ontology. Finally, in Section 5, we discussed the obtained solutions and provided concluding remark and the future scope of the present approach.

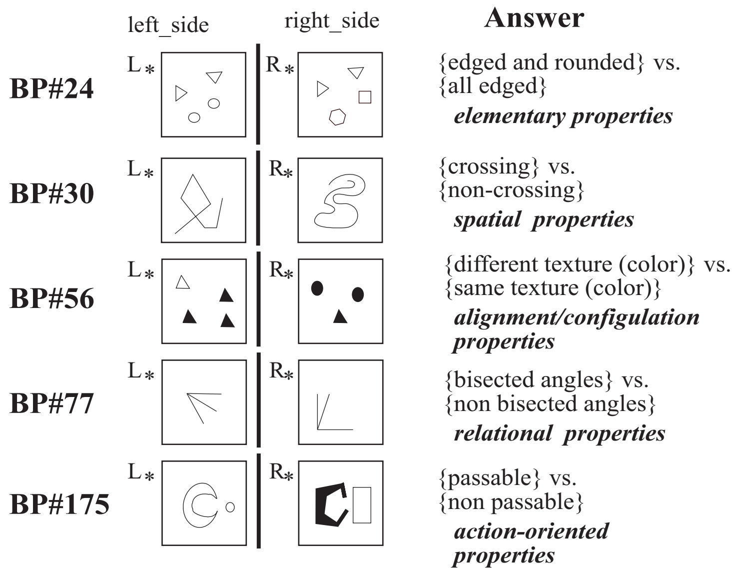

Figure 2: Levels of difficulty is consistent with abstraction levels.

2 Problem Definition

BPs are visual and logical puzzles firstly introduced by Bongard as the original 100 set [Bongard 1970] and then it was developed and added by Hofstadter with 56 pieces [Hofstadter 1979] and Foundalis with 46 pieces [Foundalis 2006]. Currently the number of the BPs is more than 200 pieces. The most important point of the puzzles that differ from other visual puzzles is that the solver has to get rules to discriminate groups of visual patterns and provide a simple logic, as illustrated in Figure 1.

In the theoretical point of view, the problem can be treated conventionally such as the definition of the solution space by the combination of all possible properties, while it apparently faces two difficulties. The first one is an infinite number of combinations of properties and the latter one is the computation time in such a large solution space. Considerable properties are classified into two at least and they are independent and dependent properties. The independent properties include elementary features such as size, shape, color, texture, spatial relationship such as the absolute position in the box, which represent with prepositions and adverbs as "below" and "above" and so on and numeric count. On the other hand, even independent properties have an infinite number of representations. For example, a name of shape "triangle" can be represented by three edge shape and a shape consisted of lines only, which goes to more primitive elements of what “triangle” means. In the same way, the numeric count has a high degree of freedom for the selection of the target property, such as the number of “black circles”, “edged objects”, “squares in the corner” and so on. In addition, dependent properties are apparently infinite. Therefore, the parameter space to find the solution which is defined frequently in conventional data-driven approaches is not given a priori. The hint to solve this problem is coming from the abduction of the initial frame of thinking.

Hofstadter [Hofstadter 1979] has discussed in his book about a concept of a systematic recursive approach to the context-based description to set an appropriate abstraction level in the beginning based on semantics and logic and then proposed a combination of functional components as i) concept network, ii) frames, iii) meta-descriptor, iv) filtering and v) sameness detector. As illustrated in Figure 2, there are different levels of concepts and it might be treated in the design of the concept network for minimizing the frame to find a solution, which is accompanied with an effective filter based on meta-data descriptions. If it is possible, the solver can deal with BPs in varying level of complexity. Piaget [Piaget 1953] was also discussed the developmental process in the cognitive development of children in his theory and it explains that there are stages of analogy from schema to mental operation of actions. In BPs with advanced versions added by Foundalis [Foundalis 2006], BP#175 requires an action-related representation obtained by the combination of the cognitive perception and mental object manipulation as classified in Figure 2 in the bottom.

According to the concepts, mathematical formulation is possible based on the theory of sets. Here, each side in a given BP is indexed as left side and right side as L_1 to L_6 and R_1 to R_6 (Figure 1). Each box on either side of a given BP holds a potentially infinite number of properties per objects, as discussed above, while if there exists an appropriate filter to select instances from a well-organized database based on defined classes (independent properties and dependent properties) with respect to requirements, it is possible to find the solution. The solution of properties P_b and P_a satisfies following conditions.

$$(L_i \notin P_a) \cap (R_i \notin P_b) \cap (L_i \in P_b) \cap (R_i \in P_a) \cap (P_a \cap P_b = \emptyset) \tag{1}$$

Then the problem is how we can build the system to reproduce the above conditions. Recent advancements of semantic knowledge representation and actual implementations can help to build the system with raw concepts and flexible relationships [Maarala et al. 2011, Durbha and King 2005, Protégé]. Some of approaches with the semantic network and some hybrid approaches applied to the limited numbers of BPs have contributed to the reaffirmation of the importance of semantics and logic. The model called RF4 [Saito and Nakano 1995] is an adaptive concept learning algorithm which solved 41 of the 100 BPs using prewritten first-order logical formulas and the model called Phaeaco [Foundalis 2006]

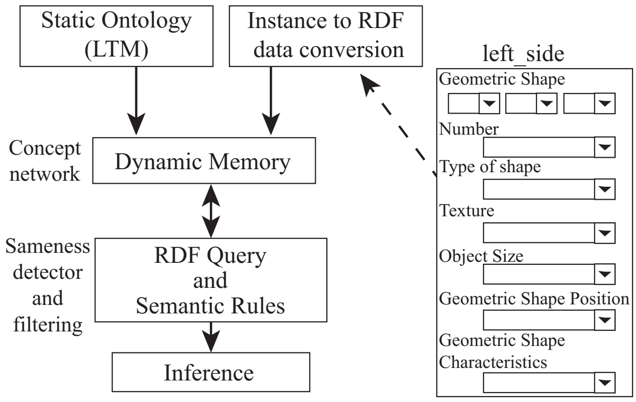

Figure 3: Ontology based framework for solving BPs.

proposed by a two-layer architecture focusing on a retinal and cognitive level computation using visual input solved 15 BPs [Foundalis 2006]. They partially demonstrated the effectiveness of the semantic network and description logic, and then the common principle is expected to be formulated and the principle explains why the semantics and logic works well to solve the BPs to avoid an infinite time for the calculation.

3 Proposed Framework

According to the concept by Hofstadter [Hofstadter 1979], we rebuilt the components based on the theory of sets and tools in the semantic web and proposed the system to solve BPs with i) ontology-based description to represent the necessary concept network, ii) description logic (DL) with assertions on concepts (TBox) and individuals (ABox) to provide minimum frames, iii) meta-data template, iv) SPARQL Protocol and RDF Query Language shortly SPARQL for the filtering function and v) Semantic Web Rule Language (SWRL) rules as sameness detector. The detail formulation is given in following sections and the validation of the system as the BP solver is shown in the chapter of results.

Knowledge Base as *Concept network*:

In our approach towards solving BPs, we have developed an Ontology-based Knowledge Base (KB) with large-scale interoperability and axioms for our use. An ontology O can be represented as: $O = (O_v, O_a)$, where O_v represents terms (vocabularies) in an ontology and O_a represents a set of ontological assertions made using the set of Vocabulary.

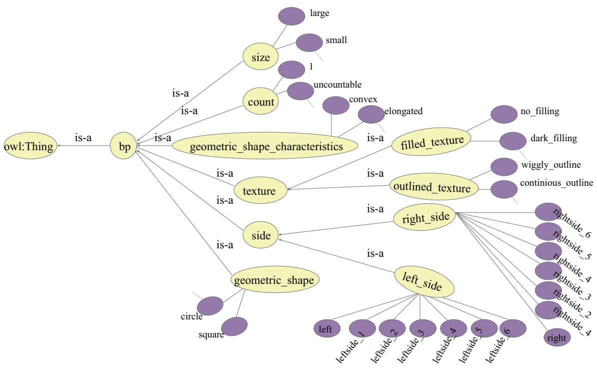

Figure 4: TBox and the class categorization used in solving BPs.

Here the vocabulary $O_v = O_c \sqcup\ O_p \sqcup\ O_e$, where O_c is a set of ontology classes (*i.e. size, texture* etc.), O_p is a set of ontology properties (they define the relationship based predicates for RDF data format (like: *hascount, hasshape* etc.) and fall into three distinct categories, namely- object properties, data properties and annotation properties) and O_e is a set of instances (entities comprising of labels, comments and literals). Assertion $O_a = O_{ca} \sqcup\ O_{pa}$, where O_{ca} is a set of ontological class assertion and O_{pa} is a set of ontological property assertion. These subsumption relations constitutes an assertion set *i.e.*

$\mathit{circle} : \mathit{geometric\_shape}$
$\mathit{left\_side} \sqsubseteq\ \mathit{side}.$

In our ontology the above assertion set can be explained as entity "*circle*" belongs to class *geometric_shape* and *left_side* is a subclass of class *side* (where $\mathit{circle} \in\ O_{\mathrm{e}}, \mathit{left\_side} \sqcup\ \mathit{side} \in\ O_{\mathrm{c}}$).

In RDF (Resource Description Framework) format class assertion can be represented as: <*circle,rdf:type,geometric_shape*>. An example for property assertion $O_{\mathrm{pa}} = \mathit{hassize}(x, \mathit{large})$, where $\mathit{hassize} \in\ O_{\mathrm{p}}$ and $x \sqcup\ \mathit{large} \in\ O_{\mathrm{e}}$, indicating in RDF format as: <*x,hassize,large*>. Class assertion in our approach assumes that every individuals of a class is different from each other (*i.e.* all the geometric shapes are different etc.). This can be represented in DL (Description Logic) syntax as follows:

Axiom: DifferentIndividual($\mathit{ClassIndividual}_1, \cdots, \mathit{ClassIndividual}_n$)
DL syntax: $\cup_{i \neq j}\ \mathit{ClassIndividual}_i \neq\ \mathit{ClassIndividual}_j$
*i.e.*DifferentIndividuall(*circle, square*) ::> $\mathit{circle} \neq\ \mathit{square}$

$$
\begin{array}{|ll|}
\hline
T_{bp} = \{count \sqsubseteq bp, & (T_{bp}, 1) \\
\quad geometric\_shape \sqsubseteq bp, & (T_{bp}, 2) \\
\quad geometric\_shape\_characteristics \sqsubseteq bp, & (T_{bp}, 3) \\
\quad side \sqsubseteq bp, & (T_{bp}, 4) \\
\quad size \sqsubseteq bp, & (T_{bp}, 5) \\
\quad texture \sqsubseteq bp, & (T_{bp}, 6) \\
\quad right\_side \sqsubseteq bp \sqcap side, & (T_{bp}, 7) \\
\quad left\_side \sqsubseteq bp \sqcap side, & (T_{bp}, 8) \\
\quad left\_side \sqsubseteq \neg right\_side, & (T_{bp}, 9) \\
\quad left\_side \sqcup right\_side \sqsubseteq side, & (T_{bp}, 10) \\
\quad filled\_texture \sqsubseteq \neg outlined\_texture, & (T_{bp}, 11) \\
\quad filled\_texture \sqcup outlined\_texture \sqsubseteq texture, & (T_{bp}, 12)\} \\
\hline
\end{array}
$$

Table 1: TBox for solving Bongard Problem.

Annotation is an important feature of an ontology which allows adding "non-logical" comments in the given ontology. Annotation to add a comment "To the Left side of BP" for class *left_side* can be represented as:

ClassAssertion(Annotation(*rdfs:comment*
"To the Left set of BP"):*side*:*left_side*))

Using the first-order logic, the above informations can be rendered as (with x and y representing *leftside_1* and *circle* respectively) -

Vocabulary – $\forall x(left(x) \Leftrightarrow \; side(x) \wedge \; bp(x) \wedge \; \exists \; y(has(x, y)))$
Assertion - geometric_shape(*circle*)

ABox (assertion components- O_a) of our ontology is represented as follows.
(Note: "__" represents the user input from GUI)

$ABox(A_{bp}) = \{1 : count,$ $(A_{bp}, 1)$
....
$infinite : count,$ $(A_{bp}, 21)$
$circle : geometric\_shape,$ $(A_{bp}, 22)$
....
$star : geometric\_shape,$ $(A_{bp}, 47)$
$parallel : geometric\_shape\_characteristics,$ $(A_{bp}, 48)$
....
$elongated\_horizontally : geometric\_shape\_characteristics,$ $(A_{bp}, 88)$
$middle : side,$ $(A_{bp}, 89)$

....

$\mathit{to\_left} : \mathit{side},$ $(A_{bp}, \mathit{94})$

$\mathit{left} : \mathit{side} \sqcap \mathit{left\_side},$ $(A_{bp}, \mathit{95})$

....

$\mathit{left\_6} : \mathit{side} \sqcap \mathit{left\_side},$ $(A_{bp}, \mathit{101})$

$\mathit{right} : \mathit{side} \sqcap \mathit{right\_side},$ $(A_{bp}, \mathit{102})$

....

$\mathit{right\_6} : \mathit{side} \sqcap \mathit{right\_side},$ $(A_{bp}, \mathit{108})$

$\mathit{large} : \mathit{size},$ $(A_{bp}, \mathit{109})$

....

$\mathit{uneven\_shape} : \mathit{size},$ $(A_{bp}, \mathit{117})$

....

$\mathit{dark\_filling} : \mathit{filled\_texture},$ $(A_{bp}, \mathit{108})$

....

$\mathit{no\_filling\_shape} : \mathit{filled\_texture},$ $(A_{bp}, \mathit{123})$

$\mathit{continuous\_outlined} : \mathit{outlined\_texture},$ $(A_{bp}, \mathit{124})$

....

$\mathit{wiggly\_outlined} : \mathit{outlined\_texture},$ $(A_{bp}, \mathit{134})$

$(\mathit{leftside\_1}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{135})$

$(\mathit{leftside\_1}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{136})$

$(\mathit{leftside\_1}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{137})$

$(\mathit{leftside\_1}, \_\_) : \mathit{hascount},$ $(A_{bp}, \mathit{138})$

$(\mathit{leftside\_1}, \_\_) : \mathit{hastexture},$ $(A_{bp}, \mathit{139})$

$(\mathit{leftside\_1}, \_\_) : \mathit{alsohastexture},$ $(A_{bp}, \mathit{140})$

$(\mathit{leftside\_1}, \_\_) : \mathit{hassize},$ $(A_{bp}, \mathit{141})$

$(\mathit{leftside\_1}, \_\_) : \mathit{isonside},$ $(A_{bp}, \mathit{142})$

$(\mathit{leftside\_1}, \_\_) : \mathit{hasshapefeature},$ $(A_{bp}, \mathit{143})$

....

$(\mathit{leftside\_6}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{180})$

$(\mathit{leftside\_6}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{181})$

$(\mathit{leftside\_6}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{182})$

$(\mathit{leftside\_6}, \_\_) : \mathit{hascount},$ $(A_{bp}, \mathit{183})$

$(\mathit{leftside\_6}, \_\_) : \mathit{hastexture},$ $(A_{bp}, \mathit{184})$

$(\mathit{leftside\_6}, \_\_) : \mathit{alsohastexture},$ $(A_{bp}, \mathit{185})$

$(\mathit{leftside\_6}, \_\_) : \mathit{hassize},$ $(A_{bp}, \mathit{186})$

$(\mathit{leftside\_6}, \_\_) : \mathit{isonside},$ $(A_{bp}, \mathit{187})$

$(\mathit{leftside\_6}, \_\_) : \mathit{hasshapefeature},$ $(A_{bp}, \mathit{188})$

$(\mathit{rightside\_1}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{189})$

$(\mathit{rightside\_1}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{190})$

$(\mathit{rightside\_1}, \_\_) : \mathit{has},$ $(A_{bp}, \mathit{191})$

$(\mathit{rightside\_1}, \_\_) : \mathit{hascount},$ $(A_{bp}, \mathit{192})$

$(rightside\_1, \_\_)$: $hastexture$, $(A_{bp}, 193)$
$(rightside\_1, \_\_)$: $alsohastexture$, $(A_{bp}, 194)$
$(rightside\_1, \_\_)$: $hassize$, $(A_{bp}, 195)$
$(rightside\_1, \_\_)$: $isonside$, $(A_{bp}, 196)$
$(rightside\_1, \_\_)$: $hasshapefeature$, $(A_{bp}, 197)$
....
$(rightside\_6, \_\_)$: has, $(A_{bp}, 234)$
$(rightside\_6, \_\_)$: has, $(A_{bp}, 235)$
$(rightside\_6, \_\_)$: has, $(A_{bp}, 236)$
$(rightside\_6, \_\_)$: $hascount$, $(A_{bp}, 237)$
$(rightside\_6, \_\_)$: $hastexture$, $(A_{bp}, 238)$
$(rightside\_6, \_\_)$: $alsohastexture$, $(A_{bp}, 239)$
$(rightside\_6, \_\_)$: $hassize$, $(A_{bp}, 240)$
$(rightside\_6, \_\_)$: $isonside$, $(A_{bp}, 241)$
$(rightside\_6, \_\_)$: $hasshapefeature$, $(A_{bp}, 242)$
$DifferentIndividuals(circle, square, rectangle.......)$, $(A_{bp}, 243)$
$DifferentIndividuals(large, small, ...)$, $(A_{bp}, 244)$
$DifferentIndividuals(dark\_filling, no\_filling, ...)$, $(A_{bp}, 245)$
$DifferentIndividuals(closed\_shape, open\_shape, ...)$, $(A_{bp}, 246)$
$DifferentIndividuals(to\_left, to\_right, ...)$, $(A_{bp}, 247)$
....

The TBox (terminological components- O_v) (represented in Figure 3) is the meta-data that defines the terms of an ontology vocabulary. The TBox representation of our ontology is as shown in Table1. ABox (assertion components- O_a) of our ontology is as represented in Figure 4 (for *left* side). The combination of both TBox and ABox is called as a Knowledge Base (domain knowledge) in this paper. In this paper, this knowledge base depicts the concept network, which according to Hofstadter [Hofstadter 1979] in his book: "*is a kind of semantic net in which all the known nouns, adjectives, etc., are linked in ways which indicate their interrelations*".

SPARQL queries as *Filters*:

In this research we employ SPARQL queries (SPARQL Protocol and RDF Query Language) as replica of the concept "Filters" by Hofstadter [Hofstadter 1979]. According to the concept [Hofstadter 1979], the concept of "Filtering" is "*making a description which concentrates on some particular way of viewing the contents of the box, and deliberately ignoring all other aspects*".

String queryString= "PREFIX
relationshipUri2:http://bongardproblem.org/bp/relationship/includes/ "+
"SELECT ?side ?feature1 ?feature2 ?feature3"

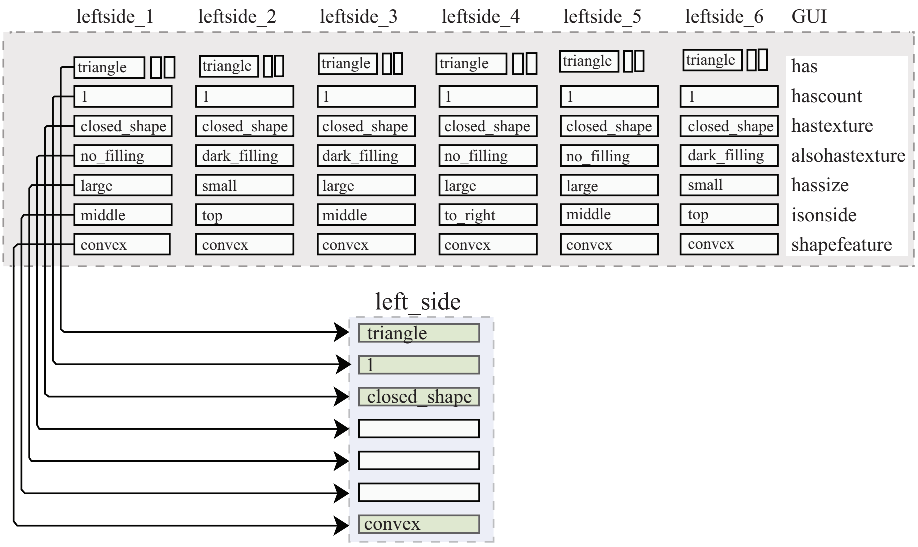

Figure 5: ABox framework for left set of boxes- for solving BP#6.

```
+ "?feature4 ?feature5 ?feature6 ?feature7" +
"WHERE { " +
" ?side relationshipUri2:has ?feature1 . "+
" ?side relationshipUri2:hascount ?feature2 . "+
" ?side relationshipUri2:hastexture ?feature3 . "+
" ?side relationshipUri2:alsohastexture ?feature4 . "+
" ?side relationshipUri2:hassize ?feature5 . "+
" ?side relationshipUri2:isonside ?feature6 . "+
" ?side relationshipUri2:hasshapefeature ?feature7 . "+
" } ";
```

Using the SPARQL queries, shown above, we try to filter out the important assertions in the ontology O, using the template (as shown below) for each box in the BP.

As a *template* (*i.e.* a description schema) to describe each box in a problem as illustrated in Figure 1, we use

Geometricshapespresent :*like : circle ∪ square...*
Countofnumberofobjectspresent :*like :* 1 ∪ 2...
Textureoftheobjects(filling) :*like : no_filling ∪ dark_filling...*
Outertexture(surface)ofobjects :*like : wiggly_outline ∪ dotted_outline...*
Objecthassize :*like : small ∪ large...*
Objectliesonside :*like : top ∪ to_left...*

Characteristicsoftheshapeoftheobjects :*like* : *parallel* $\cup$ *convex_shape..*

This template (with inputs obtained from GUI as shown in Figure 5) for each box in a given BP provides a uniform format for the description. These descriptions are then further expandable into sub-descriptions (TBox and ABox) for the SWRL rules to evaluate the knowledge base.

SWRL rules as *Sam* (*i.e.* "sameness detector"):

The semantic web rule language (SWRL) is a standard language for expressing rules over the ontology O. The syntax of the SWRL rule (for $R \geq 1$) is in the form

$$\textbf{RuleR}: antecedent(body_1 ...., body_n) \rightarrow \; consequence(head_1 ...., head_m)$$

Here antecedent (rule body: $body_x$ *for* $1 \leq x \leq n$) must be satisfied for the consequence (rule head: $head_y$ *for* $1 \leq y \leq m$) to be asserted. Here $body_x$ and $head_y$ are axioms in the form $C(V)$ or $P(O_v, O'_v)$ with $C \in O_c, V \in O_v, P \in O_p$ *and* $(O_v, O'_v) \in O_v$.

SWRL Example 1: To check for the presence of "polygon" in ontology O,

Vocabulary $V_{polygon}$= {*has, isa, hastexture, consists_of_shape*} $\cup$ { *left, closed_shaped, polygon, setoflines, leftside_1, leftside_2, leftside_3, leftside_4, leftside_5, leftside_6, a, b, c, d, e, f* }

Assertion $A_{polygon}$={*has(leftside_1,a), has(leftside_2,b), has(leftside_3,c), has(leftside_4,d), has(leftside_5,e), has(leftside_6,f), isa(a,setoflines), isa(b,setoflines), isa(c,setoflines), isa(d,setoflines), isa(e,setoflines), isa(f,setoflines), hastexture(leftside_1,closed_shaped), hastexture(leftside_2,closed_shaped), hastexture(leftside_3,closed_shaped), hastexture(leftside_4,closed_shaped), hastexture(leftside_5,closed_shaped), hastexture(leftside_6,closed_shaped), consists_of_shape(left,polygon)*}

For checking the presence of *"Polygon"* on *left_side* of a BP, we can use the Vocabulary $V_{polygon}$ with variable symbols- {*p, q, r, s, t, u*}. To check for polygon, the condition that must be satisfied can be written in natural language as: *"any closed shape that is formed by straight lines is a polygon"*. This form of knowledge in natural language can be written as a SWRL rule to check for the presence of polygon on the left and right set of boxes in a BP. Hence, rule $R_{polygon}$ can be written as:

$R_{polygon}$= **Rule 1**: has(leftside_1, *?p*) $\wedge$ has(leftside_2, *?q*) $\wedge$ has(leftside_3, *?r*) $\wedge$ has(leftside_4, *?s*) $\wedge$ has(leftside_5, *?t*) $\wedge$ has(leftside_6, *?u*) $\wedge$

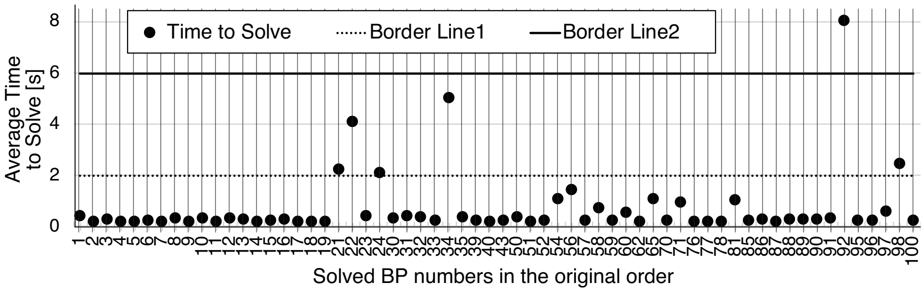

Figure 6: Results of our proposed model. Border lines are given as tentative borders to determine "Moderate BP" and "Difficult BP' in our model.

isa(*?p*,setoflines) $\wedge$ isa(*?q*,setoflines) $\wedge$ isa(*?r*,setoflines) $\wedge$ isa(*?s*,setoflines) $\wedge$ isa(*?t*,setoflines) $\wedge$ isa(*?u*,setoflines) $\wedge$ hastexture(leftside_1,closed_shaped) $\wedge$ hastexture(leftside_2,closed_shaped) $\wedge$ hastexture(leftside_3,closed_shaped) $\wedge$ hastexture(leftside_4,closed_shaped) $\wedge$ hastexture(leftside_5,closed_shaped) $\wedge$ hastexture(leftside_6,closed_shaped) $\rightarrow$ consists_of_shape(left,polygon)

This rule $R_{polygon}$ can be written in first-order logic as:

$\forall x\ \exists y_1\ \exists y_2\ \exists y_3\ \exists y_4\ \exists y_5\ \exists y_6\ \exists x_1\ \exists x_2\ \exists x_3\ \exists x_4\ \exists x_5\ \exists x_6$(consists_of_shape($x$, polygon) $\Leftrightarrow x(x_1) \wedge x(x_2) \wedge x(x_3) \wedge x(x_4) \wedge x(x_5) \wedge x(x_6) \wedge$ (has(x_1, y_1)) $\wedge$ (has(x_2, y_2)) $\wedge$ (has(x_3, y_3)) $\wedge$ (has(x_4, y_4)) $\wedge$ (has(x_5, y_5)) $\wedge$ (has(x_6, y_6)) $\wedge$ (isa(y_1,setoflines)) $\wedge$ (isa(y_2,setoflines)) $\wedge$ (isa(y_3,setoflines)) $\wedge$ (isa(y_4,setoflines)) $\wedge$ (isa(y_5,setoflines)) $\wedge$ (isa(y_6,setoflines)) $\wedge$ (hastexture(x_1,closed_shaped)) $\wedge$ (hastexture(x_2,closed_shaped)) $\wedge$ (hastexture(x_3,closed_shaped)) $\wedge$ (hastexture(x_4,closed_shaped)) $\wedge$ (hastexture(x_5,closed_shaped)) $\wedge$ (hastexture(x_6,closed_shaped)))

In order to evaluate any rule the real entities must be mapped to the variables (*i.e.* replacing the variables {*p, q, r, s, t, u*} by the entities {*a, b, c, d, e, f*}). If the mapping of rule body *(b)* is true then the mapping of rule head *(h)* must be true. Such an SWRL rule for checking for polygons could be employed for solving BP#5.

SWRL Example 2 : To check if a common shape is present on left, which is not common for the right side in BP ("shape common for left set of boxes is different from shape common for right set of boxes"- the solution of BP#97 is: <left,has,triangle> and <right,has,circle>) can be written as:

Rule1 :has(leftside_1, ?a) $\wedge$ has(leftside_2, ?a) $\wedge$ has(leftside_3, ?a) $\wedge$

has(leftside_4, $?a$) $\wedge$ has(leftside_5, $?a$) $\wedge$ has(leftside_6, $?a$) $\wedge$
DifferentIndividuals($?a$,null) $\rightarrow$ consists_of_shape(left, $?a$)
Rule2 :has(leftside_1, $?a$) $\wedge$ has(leftside_2, $?b$) $\wedge$ has(leftside_3, $?c$) $\wedge$
has(leftside_4, $?d$) $\wedge$ has(leftside_5, $?e$) $\wedge$ has(leftside_6, $?f$) $\wedge$
DifferentIndividuals($?a, ?b$) $\wedge$ DifferentIndividuals($?a, ?f$) $\wedge$
DifferentIndividuals($?e, ?d$) $\rightarrow$ consists_of_shape(left,notempty)
Rule3 :has(rightside_1, $?aa$) $\wedge$ has(rightside_2, $?aa$) $\wedge$ has(rightside_3, $?aa$) $\wedge$
has(rightside_4, $?aa$) $\wedge$ has(rightside_5, $?aa$) $\wedge$ has(rightside_6, $?aa$) $\wedge$
DifferentIndividuals($?aa, null$) $\rightarrow$ consists_of_shape(right, $?aa$)
Rule4 :has(rightside_1, $?a$) $\wedge$ has(rightside_2, $?b$) $\wedge$ has(rightside_3, $?c$) $\wedge$
has(rightside_4, $?d$) $\wedge$ has(rightside_5, $?e$) $\wedge$ has(rightside_6, $?f$) $\wedge$
DifferentIndividuals($?a, ?b$) $\wedge$ DifferentIndividuals($?a, ?f$) $\wedge$
DifferentIndividuals($?e, ?d$) $\rightarrow$ consists_of_shape(right,notempty)
Rule5 :consists_of_shape(right, $?aa$) $\wedge$ consists_of_shape(left, $?a$) $\wedge$
DifferentIndividuals($?a, ?aa$) $\rightarrow$ has_inferred_shape(left, $?a$) $\wedge$
has_inferred_shape(right, $?aa$)
Rule6 :consists_of_shape(right, $?aa$) $\wedge$ consists_of_shape(left, $?a$) $\wedge$
consists_of_shape(right,notempty) $\wedge$ consists_of_shape(left,notempty) $\wedge$
DifferentIndividuals($?a, ?aa$) $\rightarrow$ has_inferred_shape(left, $?a$) $\wedge$
has_inferred_shape(right, $?aa$)

Here **Rule 1**, **Rule 2**, **Rule 3** and **Rule 4** provides first level of inference (for similarity check), while **Rule 5** and **Rule 6** provides second level of inference (for dissimilarity check) to find solution to a give BP. The solution from the first level of inference is provided as an input for the second level of inference.

In BP#51, the first and second level of inference was derived in the output:

First-level of Inference-
[from **Rule 1** and **Rule 2**:]
Input-
<left,relationshipUri2:has,circles>,
Output-
<left,relationshipUri2:consists_of_shape,circles>,
<left,relationshipUri2:consists_of_shape,notempty>
<left,relationshipUri2:consists_of_shape,curvilinear>
[from **Rule 3** and **Rule 4**:]
Input-
<right,relationshipUri2:has,circles>,
Output-
<right,relationshipUri2:consists_of_shape,circles>,
<right,relationshipUri2:consists_of_shape,notempty>,
<right,relationshipUri2:consists_of_shape,curvilinear>

Table 2: Algorithm 1

Cross checking dissimilarity to detect possible solution for a BP
Step−1: *Select* $Rule_A$
if *(*$Rule_A$ *satisfies (*L_1,L_2,L_3,L_4,L_5,L_6*))*
if *(*$Rule_A$ *consistent in Left of BP)*
(Inference − > Left, $Predicate_{Left}$*,* $Object_1$*)*
else *(**GOTO** Step−1)*
Step−2: *Select* $Rule_B$
$Rule_B$ *satisfies (*R_1,R_2,R_3,R_4,R_5,R_6*)*
if *(*$Rule_B$ *consistent in Right of BP)*
(Inference − > Right, $Predicate_{Right}$*,* $Object_2$*)*
else *(**GOTO** Step−2)*
Step−3:*(Left,* $Predicate_{Left}$*,* $Object_1$*), (Right,* $Predicate_{Right}$*,* $Object_2$*)*
if *(*$Object_1$ *isSameAs* $Object_2$*)*
$Rule_A$ *and* $Rule_B$*, not consistent for Left and Right respectively*
else if *(*$Object_1$ *DifferentFrom* $Object_2$*)*
$Rule_A$ *and* $Rule_B$*, consistent for Left and Right respectively*
else *(**GOTO** Step−2)*
else *(**GOTO** Step−1)*

Second-level of Inference-
(after mapping real entities to the variables as-{?aa : *circle*,?a : *circle*},{?aa : *curvilinear*,?a : *curvilinear*},{?aa : *circle*,?a : *curvilinear*},{?aa : *curvilinear*,?a : *circle*})
[from **Rule 5:** and **Rule 6:**]
Input-
<*left,relationshipUri2:consists_of_shape,circles*>,
<*left,relationshipUri2:consists_of_shape,notempty*>,
<*left,relationshipUri2:consists_of_shape,curvilinear*>,
<*right,relationshipUri2:consists_of_shape,circles*>,
<*right,relationshipUri2:consists_of_shape,notempty*>,
<*right,relationshipUri2:consists_of_shape,curvilinear*>
Output-
No Output (based on geometric shapes; because of DifferentIndividuals(*?a*,*?aa*))

Hence, the above mentioned SWRL rule can be modified, for any properties and entities ($p \cup O_v$), and can be used to detect the sameness and distinct features among the two sides in a given BP. This logical reasoning was computed by embedding in the algorithm as shown in Table 2 [Maniamma and Wagatsuma 2018a, Maniamma and Wagatsuma 2018b].

In this paper, SWRL rules are used to depicts the idea of "*sam*" (sameness detector). In considering of sameness detector [Hofstadter 1979], it describes as: "*Sam is a special agent...runs around within individual descriptions and within different descriptions, looking for descriptors or other things which are repeated* (ontology in our case) *or other things which are repeated....Any structure they have in common will make comparing them that much easy*".

ABox and TBox as *Frames :*

In considering of Frames [Hofstadter 1979], it describes as: "*..mental representation of situations involve frames nested within each other. Each of the various ingredients of a situation has its own frame...nested structure of a frame gives you a way of "zooming in" and looking at small details from as close up as you wish: you just zoom in on the proper sub-frame...*" .

According to Fritz Lehmann [Lehmann 1992] "*A frame is a named data object with a flexible collection of named slots (attributes or fields) which can have values. The value are often pointers to other frames, which permits you to have a network of frames pointing to one another*". In our proposed framework, we consider classes (TBox) as "Frames" (O_c).

4 Results of Computer Experiments

Our proposed framework was implemented using Jena API to interact with the ontology, and computer experiments were verified using a PC with the Intel Core i7-3770K running at 3.40 GHz. The logical reasoning was demonstrated with a set of 55 SWRL rules to generate 12 new RDF inferred data. Among these 55 SWRL rules, 32 rules were used as first level of inference for similarity check (step 1 and 2 in Table 2) and the rest 23 rules were used as second level of inference for dissimilarity check (step 3 in Table 2) to find solution to a given BP. The system demonstrated to be a solver of 62 BPs as shown in Figure 6.

Foundalis [Foundalis 2006] carried out a survey with 31 students as whether or no they can solve 100 BPs and analyzed the difficulty levels according to ratio of solved subject as partially shown in the first column of Table 3 and Figure 6(a). In our computer experiments shown in second and third columns in Table 3 is not simply correspond to the difficulty levels given by human subjects, while our system provide a hint of reasons why the BP takes time to solve in the number fo inferences in Stage I as shown in Table 3.

Interestingly, there is an inverse correlation between the ratio of solved subjects and average time to solve in them (Figure 6(a)). It suggests the importance of the re-order of BPs according to the difficulty levels based on the human

BP	**Categorization** [Foundalis 2006] ($\hat{N}/N$)	**Number of Inferences** Stage I, Stage II, Stage III (SPARQL, SWRL, SWRL)	**Average Time to Solve** [s]
BP#7	Easy (90%)	143, 8, 2	0.22
BP#9	Easy (100%)	120, 14, 4	0.22
BP#10	Easy (87.1%)	120, 10, 2	0.34
BP#13	Easy (82.6%)	149, 12, 4	0.28
BP #16	Moderate (37.5%)	156, 14, 2	0.31
BP#22	Moderate (36.7%)	196, 14, 2	4.08

Table 3: Inference analyses with respect to difficulty levels (7 out of 62 BPs), where $\hat{N}$ and N respectively denote numbers of solved subjects and total subjects. Computation time is extracted only in the inference process from 99 trials.

performance. Indeed, this types of analyses was difficult by Phaeaco's performance due to a limited number of solved BPs (15) [Foundalis 2006] and in RF4 [Saito and Nakano 1995] which is formulated with the stochastic model in part and they report that the model solved 41 BPs without descriptions of which BPs were solved. Figure 6(b) showed the advantage of our proposed models in the sense of how many BPs were solved. According to the increase of the difficulty level (Figure 6(c), our proposed model provided solutions of 62 BPs with a similar level of computation time without an infinite loop of calculation by the assignment of multiple meta-data information, which was highly organized as the framework design with as i) ontology-based description for concept network, ii) DL with TBox and ABox for frames, iii) meta-data template, iv) SPARQL for filtering and v) SWRL rules as sameness detector.

In the logical inference of BP#9, which is categorized in "Easy BP," our model provided the first and second level of inference as output as follows.

Output of First-level of Inference (Stage II)-

<*left,relationshipUri2:alsoconsists_of_texture,continious_outlined*>,
<*left,relationshipUri2:consists_of_character,null*>,
<*left,relationshipUri2:consists_of_count,1*>,
<*left,relationshipUri2:consists_of_position,middle*>,
<*left,relationshipUri2:consists_of_shape,notempty*>,
<*left,relationshipUri2:consists_of_size,large_figure*>,
<*left,relationshipUri2:consists_of_texture,closed_shaped*>
<*right,relationshipUri2:alsoconsists_of_texture,wiggly_outlined*>,
<*right,relationshipUri2:consists_of_character,null*>,
<*right,relationshipUri2:consists_of_count,1*>,
<*right,relationshipUri2:consists_of_position,middle*>,

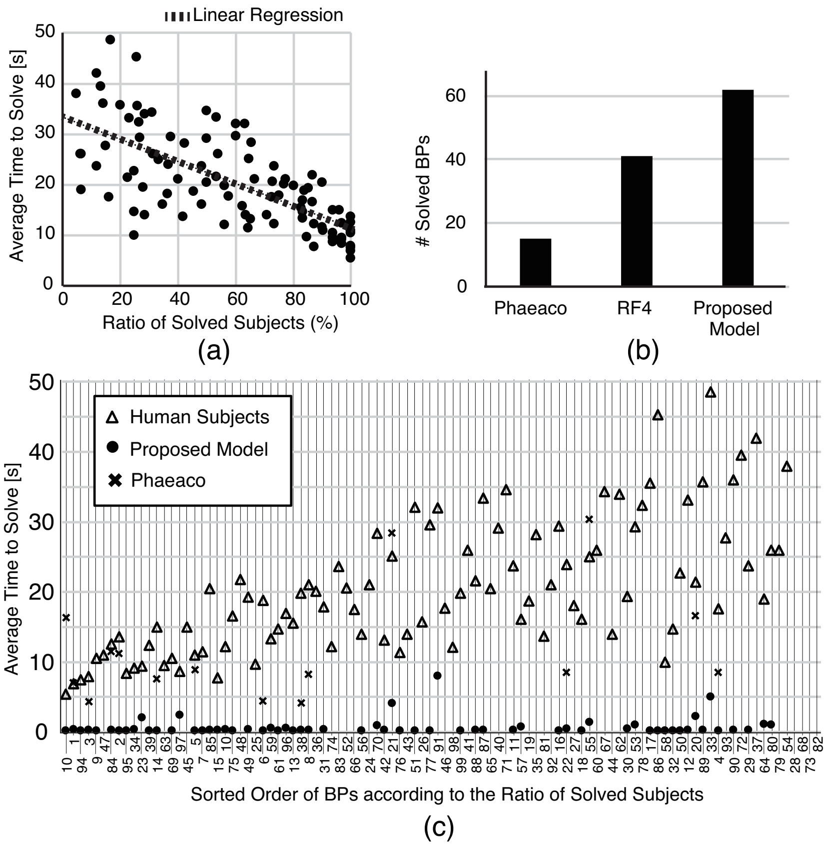

Figure 7: Comparison with other proposed models and human subjects. Phaeaco model and human subject data is replotted from data from Foundalis (2006) [Foundalis 2006]. There is no description that which BPs can be solved by RF4 [Saito and Nakano 1995].

<right,relationshipUri2:consists_of_shape,notempty>,
<right,relationshipUri2:consists_of_size,large_figure>,
<right,relationshipUri2:consists_of_texture,closed_shaped>

Output of Second-level of Inference (Stage III)-

<left,relationshipUri2:has_infered_texture,continious_outlined>,
<left,relationshipUri2:has_infered_doesnothastexture,wiggly_outlined>
<right,relationshipUri2:has_infered_texture,wiggly_outlined>,
<right,relationshipUri2:has_infered_doesnothastexture,continious_outlined>

Our results revealed that the logical inference process and therefore, even in unsolved BPs, this provides a tool of the reversal engineering of the human intelligence to analyze what kind logical components are necessary to solve.

5 Discussion and Conclusion

Our proposed framework could solve 62 BPs out of the 100 BPs (Figure 6). The inferred knowledge of each BP undergoes three-level of regressive funneling and pruning (SPARQL query, SWRL based first level of inference and SWRL based second level of inference). Each stage notices a reduction in the predicted outcome of the selected BP, which was the significant extension of preliminary reports [Maniamma and Wagatsuma 2018a, Maniamma and Wagatsuma 2018b]. As the novelty, the solver of BPs can be described in Equation 1 theoretically and simply; however no model was presented to realize the correctness. This work proved the equation works well in the realistic implementation with semantics and logic.

We have proved that our model with RDF based knowledge base is efficient in solving BPs. The current approach focuses more on independent properties of objects in a box, which demonstrated with 6 independent properties as a minimum set and then it will extend to BPs with dependent properties in the same scheme with a fine hierarchy including upper classes. In the future work, this framework can be embedded in the hybrid system as an automatic BP solver changing analogies associated with vision-based analyzers for spatial representation. It can open the new horizon of the logical reasoning system to incorporate data-driven models for decision making process in the dynamic environment.

Acknowledgments

This work was supported in part by JSPS KAKENHI (16H01616, 17H06383) and the New Energy and Industrial Technology Development Organization (NEDO).

References

[Arai et al. 2017] Arai, H. N., Todo, N., Arai, T., Bunji, K., Sugawara, S., Inuzuka, M., Matsuzaki, T., Ozaki, K.: "Reading Skill Test to Diagnose Basic Language Skills in Comparison to Machines"; Proceedings of the 39th Annual Cognitive Science Society Meeting (CogSci 2017), 1556-1561, 2017.

[Bongard 1970] Bongard, M. M.: "Pattern Recognition"; Rochelle Park, N.J.: Hayden Book Co., Spartan Books, 1970.

[Chen et al. 2016] Chen, S., M.úller H. J., and Conci M.: "Amodal completion in visual working memory"; Journal of Experimental Psychology: Human Perception and Performance, 42 (9): 1344–1353, 2016.

[Chopra and Rao 1994] Chopra, S., Rao, M. R.: "The Steiner tree problem I: Formulations, compositions and extension of facets"; Mathematical Programming, 64 (2):209-229, 1994. 10.1007/BF01582573.

[Depeweg et al. 2018] Depeweg, S., Rothkopf, C., and J´akel, F.: "Solving Bongard Problems with a Visual Language and Pragmatic Reasoning"; arXiv:1804.04452, 2018.

[Durbha and King 2005] Durbha, S. S., King R. L.: "Semantics-Enabled Framework for Knowledge Discovery from Earth Observation Data Archives"; IEEE Transactions on Geoscience and Remote Sensing, 43 (11): 2563–2572, 2005.

[Forbus et al. 1998] Forbus, K. D., Gentner, D., Markman, A. B., Ferguson, R. W.: "Analogy just looks like high level perception: Why a domain- general approach to analogical mapping is right"; Journal of Experimental and Theoretical Artificial Intelligence, 10 :231–257, 1998.

[Foundalis 2006] Foundalis, H.: "Phaeaco: A Cognitive Architecture Inspired by Bongard's Problems"; Doctoral dissertation, Indiana University, Center for Research on Concepts and Cognition (CRCC), Bloomington, Indiana, 2006.

[Hofstadter 1979] Hofstadter, D. R.: "Gödel, Escher, Bach: an Eternal Golden Braid"; New York: Basic Books, 1979.

[Johnson et al. 2017] Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Zitnick, C. L., Girshick, R.: "Inferring and Executing Programs for Visual Reasoning"; IEEE International Conference on Computer Vision, 3008–3017, 2017.

[Kurzweil 2014] Kurzweil, R.: "The Singularity is Near"; In Ethics and Emerging Technologies, (London: Palgrave Macmillan UK) 393–406, 2014.

[Lehmann 1992] Lehmann, F.: "Semantic networks"; Computers & Mathematics with Applications, vol. 23, no. 2-5, pp. 1-50, 1992. 10.1016/0898-1221(92)90135-5.

[Linhares 2000] Linhares, A.: "A glimpse at the metaphysics of Bongard problems"; Artificial Intelligence Journal, 121 (1): 251–270, 2000.

[Maarala et al. 2011] Maarala, A. I., Su, X., Riekkit, J.: "Semantic Reasoning for Context-Aware Internet of Things Applications"; IEEE Internet of Things Journal, 4 (2): 461–473, 2011.

[Maniamma and Wagatsuma 2018a] Maniamma, J., Wagatsuma, H.: "Human Abduction for Solving Puzzles to Find Logically Explicable Rules in Bongard Problems: An Ontology-based Model"; Proceedings of the Joint Workshop on Architectures and Evaluation for Generality, Autonomy and Progress in AI (AEGAP 2018), Stockholm, Sweeden, July 15th, 2018.

[Maniamma and Wagatsuma 2018b] Maniamma, J., Wagatsuma, H.: "A Semantic Web Technique as Logical Inference Puzzle-Solver for Bongard Problems"; Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th-to-12th, 2018.

[McCarthy and Hayes 1969] McCarthy, J., Hayes, P. J.: "Some philosophical problems from the standpoint of artificial intelligence"; Machine Intelligence Journal, 4 : 463–502, 1969.

[Mitchell 2003] Mitchell, M.: "Analogy-Making as Perception: A Computer Model (Neural Network Modeling and Connectionism)"; A Bradford Book, 2003.

[Pfeifer and Bongard 2007] Pfeifer, R., Bongard, J.: "How the body shapes the way we think"; A New View of Intelligence, 394, 2007.

[Piaget 1953] Piaget, J.: "The origin of intelligence in the child"; New Fetter Lane (New York- Routledge & Kegan Paul), 1953.

[Protégé] Protégé - Stanford University, https://protege.stanford.edu.

[Saito and Nakano 1995] Saito, K., Nakano, R.: "Adaptive concept learning algorithm: RF 4"; Transactions of Information Processing Society, 36 (4): 832–839, 1995.

[Salameh et al. 2014] Salameh, K., Tekli, J., Chbeir, R.: "SVG-to-RDF Image Semantization"; Similarity Search and Applications, 214–228, 2014.

[Silver et al. 2016] Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: "Mastering the game of Go with deep neural networks and tree search"; nature Journal, 529 (7587): 484–489, 2016.

[Sun 2019] Sun, Y.: "Solving the Steiner Tree Problem in Graphs using Physarum-inspired Algorithms"; ArXiv, 2019. https://yahuisun.com

[Weitnauer and Ritter 2012] Weitnauer, E., Ritter, H.: "Physical Bongard Problems"; IFIP Advances in Information and Communication Technology, 381: 157–163, 2012.

[Zand et al. 2016] Zand, M., Doraisamy, S., Halin, A. A., Mustaffa, M.: "Ontology-Based Semantic Image Segmentation using Mixture Models and Multiple CRFS"; IEEE Transactions on Image Processing, 25 (7): 3233–3248, 2016.