CSI5101 - Knowledge Representation
Winter 2024

Assignment 2 - Ontologies

Submitted by

Romario Vaz : 300308477
Saranya Krishnasami : 300321456

Index

Ontologies

Q1 – Ontology Design

(a) The topic is to create university ontology. Some **in-scope competency questions** are ,

- What are the courses offered in the department of Engineering ?
- What are the additional services available for international students ?

Out-of-scope competency questions

- How many parking spots are available on the University of Ottawa campus?
- How many alumni from the Engineering department have started their own businesses?

(b) Exploring Schema.org

Does it contain sufficient properties for your needs?

For the most part, the CollegeOrUniversity class in Schema.org contains a wide range of properties that are relevant for describing a college or university. It includes properties such as alumni, opening hours, address, aggregate rating, contact point, and more. However, certain specific classes like **Programs and Courses, Accreditation and Rankings, Admission Information** are significant aspects that are missing

Multiple inheritance in Schema.org

In Schema.org, multiple inheritance is a supported feature, as evidenced by entities like *CollegeOrUniversity* inheriting traits from *EducationalOrganization, CivicStructure*, and *Organization*. This design rationale is fitting because educational institutions, including colleges and universities, embody characteristics that span organizational structures, physical facilities, and educational missions.

Strange properties:

foundingLocation: While it might be interesting to know the founding location of an educational organization, including this property for universities might seem less relevant in comparison to other types of organizations. For universities with long histories, the founding location may not necessarily reflect their current identity or operations.

duns: The Dun & Bradstreet DUNS number for identifying an organization might seem unusual for universities, as they are primarily academic institutions rather than commercial entities. While universities may have DUNS numbers for administrative purposes, it may not be a commonly used identifier for describing them in the context of Schema.org.

(c) Exploration of the University Ontology 1.0:

The University Ontology provides a structured framework for describing universities and their components, such as University, Department, Course, Faculty, and ResearchGroup. It outlines relationships between entities, offers definitions for various roles and categories. The ontology is designed to facilitate the organization and understanding of university-related data and activities.

This ontology has some features that are more relevant such as Education Organization and sub classes such as department, institute, university. Publication details , course work e.t.c

(d) Building an Ontology

The Newly designed Ontology (Stanford Publications. (2000)) has the classes that are very crucial for a university setup. The following tables illustrate these classes, along with their relationships and the knowledge base they represent.

Classes	**Relations**
Course	offers(Department, Course)
Department	taughtBy(Course, Professor)
Professor	enrolledIn(Student, Course)
Student	specializesIn(Professor, ResearchArea)
ResearchProject	affiliatedWith(Professor, Department)
ResearchGroup	conductsResearch(ResearchGroup, ResearchArea)
University	locatedAt(ResearchGroup, Department)
Person	

	Course
Properties	CourseCode (Domain: Courses, Range: Text)
	CourseTitle (Domain: Courses, Range: Text)
	Instructor (Domain: Courses, Range: Professor)
	EnrolledStudents (Domain: Courses, Range: Student)

	Professor
Superclass:	Person
Properties:	OfficeNumber (Domain: Professor, Range: Text)
	Department (Domain: Professor, Range: Department)
	TeachesCourses (Domain: Professor, Range: Courses)
	SupervisesProjects (Domain: Professor, Range: ResearchProject)

	Department
Superclass:	Organization Units
Properties:	DepartmentCode (Domain: Department, Range: Text)
	DepartmentName (Domain: Department, Range: Text)
	HeadOfDepartment (Domain: Department, Range: Professor)
	CoursesOffered (Domain: Department, Range: Courses)

	Student
Superclass:	Person
Properties:	StudentID (Domain: Student, Range: Text)
	Department (Domain: Student, Range: Department)
	EnrolledCourses (Domain: Student, Range: Courses)
	ResearchProjects (Domain: Student, Range: ResearchProject)

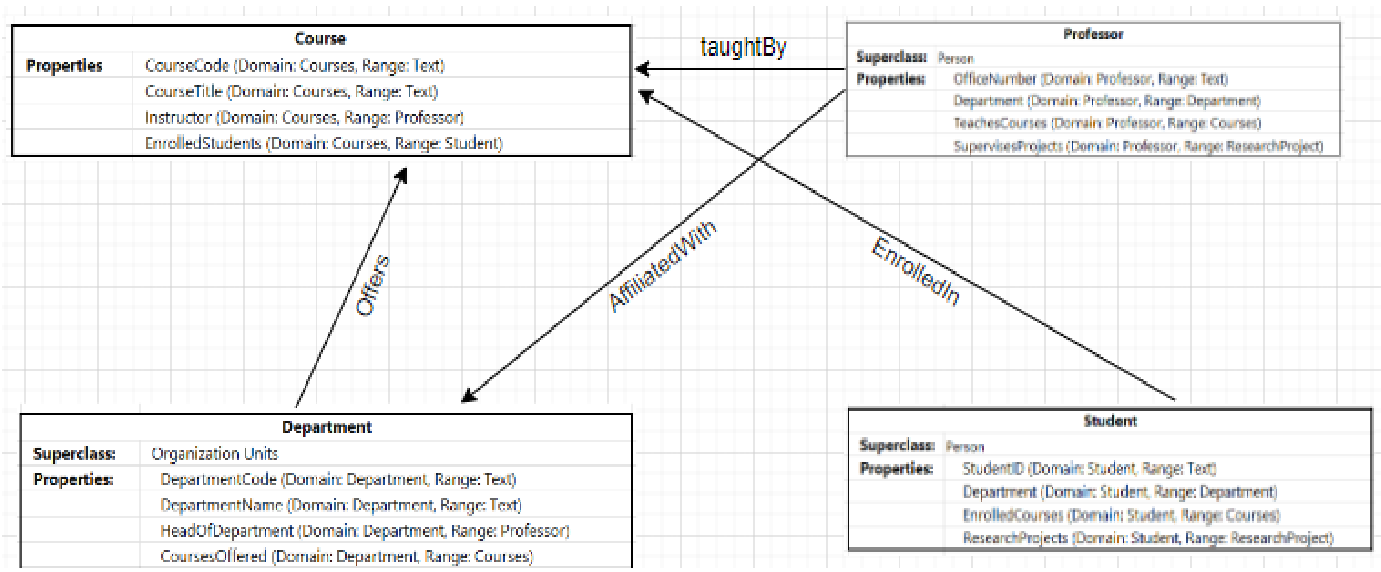
	ResearchProject
Superclass:	Project
Properties:	ProjectID (Domain: ResearchProject, Range: Text)
	ProjectTitle (Domain: ResearchProject, Range: Text)
	ProjectDescription (Domain: ResearchProject, Range: Text)
	PrincipalInvestigator (Domain: ResearchProject, Range: Professor)
	Participants (Domain: ResearchProject, Range: Person)

	ResearchGroup
Superclass:	None
Properties:	GroupID (Domain: ResearchGroup, Range: Text)
	GroupName (Domain: ResearchGroup, Range: Text)
	GroupLeader (Domain: ResearchGroup, Range: Professor)
	GroupMembers (Domain: ResearchGroup, Range: Person)

	University
Superclass:	None
Properties:	UniversityName (Domain: University, Range: Text)
	Location (Domain: University, Range: Text)
	Departments (Domain: University, Range: Department)
	Faculties (Domain: University, Range: Professor)

	Person
Superclass:	None
Properties:	FirstName (Domain: Person, Range: Text)
	LastName (Domain: Person, Range: Text)
	Email (Domain: Person, Range: Text)
	Address (Domain: Person, Range: Text)

Below figure shows a sample ontology diagram representing a few classes from the ontology and their relationship.



Knowledge base:

Course:
Introduction to Artificial Intelligence
Machine Learning
Data Structures and Algorithms
Department:
Computer Science
Professor:
Dr. John Doe
Dr. Jane Smith
Student:
Alice Johnson
Bob Thompson
ResearchProject:
Natural Language Processing
Computer Vision
ResearchGroup:
Machine Learning Lab
University:
My University
Person:
Alice Johnson
Bob Thompson
Dr. John Doe
Dr. Jane Smith

Relations:
offers(Computer Science, Introduction to Artificial Intelligence)
offers(Computer Science, Machine Learning)
offers(Computer Science, Data Structures and Algorithms)
teacherOf(Dr. John Doe, Introduction to Artificial Intelligence)
teacherOf(Dr. Jane Smith, Machine Learning)
member(Computer Science, Dr. John Doe)
member(Computer Science, Dr. Jane Smith)
takesCourse(Alice Johnson, Introduction to Artificial Intelligence)
takesCourse(Bob Thompson, Machine Learning)
researchProject(Machine Learning Lab, Natural Language Processing)
researchProject(Machine Learning Lab, Computer Vision)
alumnus(My University, Alice Johnson)
alumnus(My University, Bob Thompson)

Competency Questions

- What courses does the Computer Science department offer?
    - Classes: Course, Department
    - Relations: offers(Department, Course)

- Who are the professors teaching in the Computer Science department?
  - Classes: Professor, Department
  - Relations: teacherOf(Professor, Course), member(Department, Professor)
- Which students are enrolled in the Artificial Intelligence course?
  - Classes: Student, Course
  - Relations: takesCourse(Student, Course)
- What research projects are affiliated with the Machine Learning research group?
  - Classes: ResearchProject, ResearchGroup
  - Relations: researchProject(ResearchGroup, ResearchProject)
- Who are the alumni of the university?
  - Classes: Person, University
  - Relations: alumnus(University, Person)

References:

Stanford Publications. (2000). Ontology development 101: A guide to creating your first Ontology https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Q2 – Linked Open Data (LOD) in Biomedical domain

Starting Point: Disorder - Autism Spectrum Disorder (ASD)

Disease Ontology (DO):

- Autism Spectrum Disorder (ASD) is classified in Disease Ontology (DO) under the term "Autism spectrum disorder."

- Additional information includes synonyms, related disorders, and associated genes.

ICD9 (International Classification of Diseases):

- Autism Spectrum Disorder is classified under ICD9 code 299.0.

- It is categorized under the broader group of pervasive developmental disorders.

ICD10 (International Classification of Diseases):

- Autism Spectrum Disorder is classified under ICD10 code F84.

- ICD10 provides further sub-classifications for different subtypes and severity levels of ASD.

MeSH (Medical Subject Headings):

- Autism Spectrum Disorder is indexed under MeSH with various subheadings, including classification, diagnosis, epidemiology, and therapy.

- It provides additional insights into aspects such as diagnostic criteria, treatment options, and epidemiological data related to ASD.

SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms):

- SNOMED CT represents Autism Spectrum Disorder with various concepts and relationships, including subtypes, manifestations, and associated conditions.

- It offers detailed clinical information, such as specific symptoms, diagnostic criteria, and treatment modalities associated with ASD.

UMLS (Unified Medical Language System):

- UMLS integrates and links various biomedical vocabularies and ontologies.

- For Autism Spectrum Disorder, UMLS provides mappings to concepts from MeSH, SNOMED CT, and ICD9/ICD10.

- It facilitates interoperability and integration of information by establishing cross-references between different terminologies.

Additional Information Provided by Each Ontology:

- ICD9 and ICD10 provide standardized codes for classifying Autism Spectrum Disorder, enabling uniform reporting and analysis of disease data.

- MeSH offers indexing terms and descriptors, facilitating literature search and retrieval related to ASD.

- SNOMED CT provides a comprehensive clinical terminology, allowing for detailed representation of clinical concepts and relationships associated with ASD.

- UMLS acts as a bridge between different terminologies, enabling interoperability and integration of information from diverse sources related to ASD.

Reference:

Disease Ontology Consortium. (2023). *Do*. Disease Ontology. https://disease-ontology.org/

Biomedical Ontologies and Controlled Vocabularies (2018), University of Michigan, https://guides.lib.umich.edu/ontology/ontologies

Q3 – Google Knowledge Graphs (GKG) and Search Engines

1. Define what Knowledge Panels/Cards are and why they are used?

These are cards/panels presented when a user searches for some information on search engines like Google. For Google, they are generated using the information from Google's knowledge graph. These panels display key information about entities directly in the search results, so that the user does not have to click on a link and visit another web page.

2. Describe the relation between GKGs and Schema.org?

GKG uses structured data, including Schema.org markup on web pages, to understand the relationships between entities and to populate its knowledge graph (Berkel, 2023). This structured information helps Google to create Knowledge Panels and improve search relevance and user experience.

3. In the second article, section 2 (Use schema markup on your site), look at the “sameAs” in the example. This “sameAs” link is fundamental in LOD. Why? What does it do?

The "sameAs" link is crucial for establishing connections between an entity on a webpage and equivalent entities across different databases or platforms. It unambiguously establishes the identity of an entity by connecting it to sources like its Wikipedia pages, official websites, etc. (Jones, 2023). If, for example, a new website is created for a business, the sameAs link can help search engines like Google understand that the new website is still referring to the business.

4.

a. The superclass of Organization is Thing, and some properties are inherited from Thing. What are some of these properties? Do they seem relevant to Organization?

Some of the properties inherited from Thing include alternateName, description, url, and other properties which every "Thing" must have. Since Organization is a subclass of Thing, these properties are relevant to Organization. From a purely semantic perspective, it makes sense that a schema markup of an organization should specify an alternate name, description, URL, and other properties which are just as important (if not more) than the properties from the subclass.

b. Organization can be the range for which properties?

There are several properties for which Organization can be the range. As per the schema.org webpage for Organization, the full list of these properties are: acquiredFrom, affiliation, agent, alumniOf, attendee, attendees, authenticator, author, bccRecipient, bookingAgent, branchOf, brand, broadcastAffiliateOf, broadcaster, broker, buyer, carrier, ccRecipient, claimInterpreter, composer, contributor, copyrightHolder, creator, creditedTo, customer, department, employmentUnit, endorsee, endorsers, followee, fundedItem, funder, grantee, hiringOrganization, hostingOrganization, issuedBy, landlord, legislationPassedBy, legislationResponsible, lender, maintainer, manufacturer, member, memberOf, members, merchant, offeredBy, organizer, parentOrganization, participant, performer, performers, producer, productionCompany, provider, publishedBy, publisher, publisherImprint, recipient, recognizedBy, recognizingAuthority, recordLabel, reviewedBy, sdPublisher, seller, sender, serviceOperator, sourceOrganization, spokenByCharacter, sponsor, subOrganization, toRecipient, translator, underName, vendor, worksFor.

c. Why is a type like CreativeWork used so much as a possible type for various properties?

CreativeWork is often used as a type for various properties because it encompasses the most generic kind of creative work including books, movies, photographs, software, etc. which are relevant to many organizational activities. For example, the ethicsPolicy property of Organization could be a policy document, which is a creative work.

d. Why is Text like “Text or PostalAddress” often used as possibly type for various properties? What are the advantages and disadvantages of using Text as a property type?

Location data can either be represented as simple unstructured text or as structured data of the type PostalAddress. Both representations are equally valid and hence text like "Text or PosalAddress" is often used as possible types for many properties related to addresses and location. The primary disadvantage of using Text over a PostalAdress is that using a structured data type reduces any ambiguity. The main advantages of using Text is that it offers additional flexibility and simplicity. Also, if an address does not neatly fit into the structure defined in PostalAdress, using Text would be useful, for e.g., if an address was located in a areas where street names are not commonly used (e.g. many rural areas), or regions where postal codes are not used (e.g. Hong Kong).

5. Some classes don't have the same status in Schema.org, for example, look at VirtualLocation. What's its status? What does that mean in the context of ontology development? Is Schema.org curated? If yes, by whom? Are they to be trusted?

VirtualLocation's status is "new", i.e., it is susceptible to change based on implementation feedback and the ways that other applications and websites adopt it. This status reflects the development stage within Schema.org's ontology development, based on community feedback and evolving web standards. Schema.org is curated by two main groups, the Community Group and the Steering Group (About Schema.org, 2024). The larger Community Group handles day-to-day activities including the evolution of the schemas and related discussions and integrations. High-level oversight of the project is curated by the smaller Steering Group involving participants from Yahoo, Yandex, Microsoft, and Google. Since the steering group members are from organizations in whose best interests it is to have proper schema definitions, it is reasonable to assume that they can be trusted.

6. A PostalAddress is a subclass of ContactPoint, why would an ontology want to make that distinction?

Differentiating PostalAddress from ContactPoint allows for a more detailed ontology, enabling the representation of postal addresses as a particular type of contact point among others (e.g., email addresses, phone numbers), reflecting the diverse ways that entities can be contacted or located.

7. Go in ListenAction. Look at the examples (in JSON-LD, it's easy to read). They differentiate John listened to Pink. John listened to Star 101.3. How is such differentiation done?

The differentiation between listening to different entities (a music group vs. a radio station) in ListenAction examples is achieved through specifying the type of object that the agent is listening to. Within Schema.org, the MusicGroup and RadioStation types refer to the types that the entities Pink! and Star 101.3 are instances of respectively.

8. Let say you wanted to express that Bora went to the National Art Center in Ottawa to hear a concert by the NAC Orchestra playing music by Beethoven, what classes could be assigned to “Bora”, “National Art Center”, “NAC Orchestra”, “Ottawa”, “Beethoven”. What are the properties between them? Inspired by the JSON-LD format shown in the ListenAction page examples, represent the sentence above using that format. You can make-up classes if you do not find the appropriate ones in Schema.org.

To represent Bora attending a concert by the NAC Orchestra at the National Art Center in Ottawa featuring Beethoven's music, the classes that could be assigned are as follows, along with associated properties:

Entity	**Class**	**Properties**
Bora	Person	name:Bora
National Art Center	PerformingArtsTheater	name: National Art Center containedInPlace: Ottawa
NAC Orchestra	PerformingGroup	name: NAC Orchestra
Ottawa	Place	name: Ottawa
Beethoven	Person	name: Beethoven

In order to construct the JSON-LD representation, we also use two additional classes including MusicEvent and MusicComposition, and their associated properties as shown below:

The sentence can be represented in JSON-LD format as follows:

```
<!-- Bora went to the National Art Center in Ottawa to hear a concert
by the NAC Orchestra playing music by Beethoven-->

<script type="application/ld+json">
```

```
{
  "@context": "http://schema.org",
  "@type": "MusicEvent",
  "name": "Beethoven Concert by NAC Orchestra",
  "location": {
    "@type": "PerformingArtsTheater",
    "name": "National Art Center",
    "containedInPlace": {
      "@type": "Place",
      "name": "Ottawa"
    }
  },
  "performer": {
    "@type": "PerformingGroup",
    "name": "NAC Orchestra"
  },
  "workPerformed": {
    "@type": "MusicComposition",
    "composer": {
      "@type": "Person",
      "name": "Beethoven"
```

```
    }
  },
  "attendee": {
    "@type": "Person",
    "name": "Bora"
  }
}
</script>
```

References

Berkel, M. van. (2023, August 25). How schema markup enhances your google knowledge panel. Schema App Solutions. https://www.schemaapp.com/schema-markup/how-schema-markup-helps-you-gain-or-enhance-a-google-knowledge-panel/

Jones, G. (2023, March 6). Using the sameas schema.org markup tag in Seo. InLinks. https://inlinks.com/schema/using-the-sameas-schema-org-markup-tag-in-seo

About Schema.org. About Page. (2024, February 12). https://schema.org/docs/about.html

Logics

Q4 – Propositional logic and proofs

The four selected exercises are e,f,g, and h. As per standard notation, all the steps above the horizontal line are the premises, and those below are the inferences.

e. To show : $\neg Q$

1.	$(S \rightarrow \neg Q)$	premise
2.	$(P \rightarrow S)$	premise
3.	$\neg\neg P$	Premise
4.	P	Double negation (3)
5.	S	Modus ponens (4,2)
6.	$\neg Q$	Modus ponens (1,5)

f. To show : $\neg Q$

1.	$T \rightarrow P$	premise
2.	$Q \rightarrow S$	premise
3.	$S \rightarrow T$	premise
4.	$\neg P$	premise
5.	$\neg T$	Modus tollens (4,1)
6.	$\neg S$	Modus tollens (5, 3)
7.	$\neg Q$	Modus tollens (6, 2)

g. To show: Q

1.	R	premise
2.	P	premise
3.	$P \to (R \to Q)$	premise
4.	$R \to Q$	modus ponens (2,3)
5.	Q	modus ponens (1,4)

h. To show: V

1.	$(R \to S) \to Q$	premise
2.	$\neg Q$	premise
3.	$\neg(R \to S) \to V$	premise
4.	$\neg(R \to S)$	modus tollens (1, 2)
5.	V	modus ponens (3, 4)

Q5 – Predicate logic

1. All graduate students have an undergraduate degree.

$\forall x:(GraduateStudent(x) \rightarrow HasUndergraduateDegree(x))$

2. Anyone with an undergraduate degree studied in at least one university.

$\forall x \ \exists y(HasUndergraduateDegree(x) \rightarrow (University(y) \land StudiedIn(x,y)))$

3. A university, at any particular year, delivers courses to many students.

This statement cannot be expressed in first-order logic because it uses the quantifier "many", which is not fully supported within standard first-order logic without additional non-standard definitions. The closest that one could express this concept is with interpreting "many" as "there exists" (i.e., at least one), but this is not equivalent to "many".

4. For a professor to teach a course, there must be a student attending it.

$\forall p \forall c \exists s:(Professor(p) \land Course(c) \land Teaches(p,c)) \rightarrow (Student(s) \land Attends(s,c))$

5. Classmates are students taking the same course.

$\forall x \forall y \forall c: (Student(x) \land Student(y) \land Course(c) \land Attends(x,c) \land Attends(y,c) \rightarrow Classmates(x,y)$

6. A graduate student takes less courses than an undergraduate student.

This statement cannot be expressed in first-order logic because it uses the quantifier "less than", which is not fully supported within standard first-order logic without additional non-standard definitions. The closest that one could express this concept is with defining another predicate like TakesLessCourses(x,y) would need to be defined by some mechanism outside of basic predicate logic to compare the number of courses.

Q6 – Description Logics

Description Logics

41

T-Box

Blue Box $\sqsubseteq$ Box

Black Box $\sqsubseteq$ Box

Blue Box $\doteq$ Box $\sqcap$ (Glass $\sqcup$ Metal)

Black Box $\doteq$ Box $\sqcap$ (Paper $\sqcup$ Cardboard)

~~Glass~~

Recyclable $\doteq$ Glass $\sqcup$ Metal $\sqcup$ Paper $\sqcup$ Cardboard

Glass $\sqsubseteq$ Recyclable

Metal $\sqsubseteq$ Recyclable

Paper $\sqsubseteq$ Recyclable

Cardboard $\sqsubseteq$ Recyclable

A-Box

Glass (Wine Bottle)

Paper (News Paper)

40

b) T-Box Add 2 more rules

Large Box $\sqsubseteq$ Box $\sqcap$ (SizeVolume < 1000 gms)

Small Box $\sqsubseteq$ Box $\sqcap$ (SizeVolume > 1000 gms)

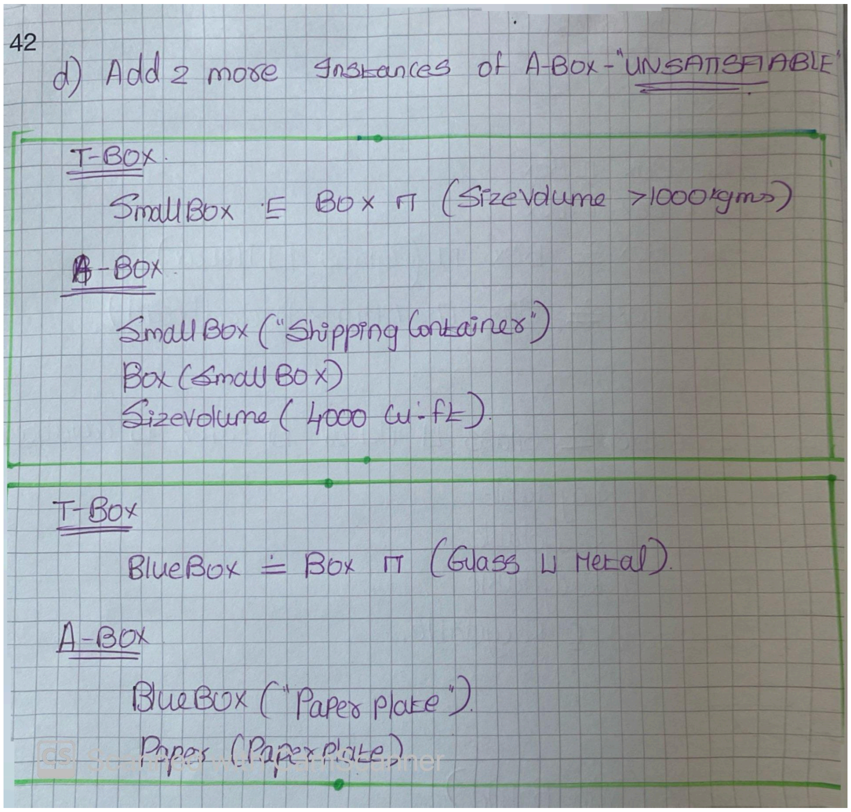
SizeVolume $\sqsubseteq$ Size

c) Add 5 more instances to A-Box - "SATISFIABLE"

A-Box

1. Metal (Aluminium Can)
2. Glass (Glass Jar)
3. Cardboard (Cardboard Tube)
4. ~~Plastic~~ (

Paper (Magazine)



42

d) Add 2 more instances of A-Box - "UNSATISFIABLE"

T-BOX

SmallBox $\sqsubseteq$ Box $\sqcap$ (SizeVolume > 1000 kgms)

A-BOX

SmallBox ("Shipping Container")
Box (Small Box)
SizeVolume (4000 cu-ft).

T-Box

BlueBox $\doteq$ Box $\sqcap$ (Glass $\sqcup$ Metal).

A-Box

BlueBox ("Paper plate").
Paper (Paper Plate)

The unsatisfiable logic **example 1** where the shipping container is declared as Small box and its volume is more than the description of small box makes it unsatisfiable as per the T-Box logic of the small box belongs to Box class and its volume size to be less than 1000 kgs.

For **example2** : The T-Box for blue box mentions it belongs to Box and it should only contain Metal or Glass. The A-Box assertion where the paper plate in the blue box is unsatisfiable as it should be in black box.

References:

(Meyer, 2010), *Description Logics* ,
https://www.cse.unsw.edu.au/~cs4415/2010/resources/comp4418-DL.pdf

Q7 – Fuzzy Logic

Fuzzy logic is an approach based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. As a result, fuzzy logic is well-suited for the following:

- Engineering for decisions without clear certainties and uncertainties, or with imprecise data -- such as with natural language processing technologies; and
- Regulating and controlling machine outputs, according to multiple inputs/input variables -- such as with temperature control systems.

(a) Applications of Fuzzy logic:

Seiji et al (Seiji, 1983) paper explores fuzzy logic's application in estimating train braking distance, addressing the complexity of factors like speed and braking force. Using a fuzzy logic system informed by expert knowledge, it **models real-world scenarios to determine braking distance accurately**. Sensitivity analysis ensures result stability. Fuzzy logic proves valuable in railway infrastructure design decisions. Examples from the literature highlight its efficacy in automatic train control systems.

Another paper (Abdullah,2021) discusses the transition from conventional farming to smart farming utilizing IoT technology. It introduces a framework **leveraging advanced fuzzy logic to optimize pump control based on user-defined variables**, focusing on air humidity, temperature, and soil moisture. By integrating sensors as inputs and IoT as output, the system reduces water consumption and watering time significantly compared to manual handling. This innovation holds promise for enhancing crop monitoring and management efficiency in agriculture.

(b) Fuzzy logic in rice cookers and washing machine

Yes, fuzzy logic is being used in rice coolers (Mohan, 2015) and washing machines.

Fuzzy Logic in Rice Cookers: The article explores the integration of fuzzy logic technology into rice cookers, allowing for precise cooking by adjusting parameters based on rice type and desired texture. This innovation has led to the development of rice cookers capable of producing perfectly cooked rice consistently.

Fuzzy Logic in washing Machines: The article explains the concept of fuzzy logic in washing machines, highlighting its role in optimizing washing performance. Fuzzy logic algorithms adjust wash parameters based on factors like load size and dirt level, ensuring efficient cleaning. This technology enhances washing machine functionality by reducing water and energy consumption while improving washing quality. Additionally, fuzzy logic minimizes spinning noise by balancing the washing load effectively.

(c) Japanese Train:

The Sendai Subway 1000 series was the world's first train type to use fuzzy logic to control its speed.

- Fuzzy logic in trains starts with calculating distances between stations and activating the braking system about 500 meters from a station based on speed and distance input.
- It operates by converting numerical values into fuzzy sets using fuzzification, applying rules or inference based on these sets, and then translating them back into numerical values through defuzzification to control the train's braking system

References:

Yasunobu, Seiji & MIYAMOTO, Shoji & Ihara, Hirokazu. (1983). A Fuzzy Control for Train Automatic Stop Control. Transactions of the Society of Instrument and Control Engineers.

N. Abdullah et al. (2021), "Towards Smart Agriculture Monitoring Using Fuzzy Systems,"

Pradhan, Mohan. (2015). An Intelligent Fuzzy Based Technique Of Making Food Using Rice Cooker. 10.13140/RG.2.1.3491.7283.