

# Task 1

---

## A. Is the graph a valid RDF document?

Yes, from a purely technical, syntax-checking point of view, the graph is a valid RDF document.

### Syntactically Correct:

Looking at the structure itself (i.e. to read it in Turtle format), nothing breaks the basic rules of RDF. Each statement is formed correctly in **subject–predicate–object** format. It uses URIs for things like classes and properties (**foaf:Person**, **foaf:knows** etc.) and literals for data (**"Alice"^^xsd:string**, **"twenty-five"^^xsd:string**). An RDF parser would have no issue with this, throwing no errors from a purely syntactic point of view.

### Semantic Issues Appear Later:

Nevertheless, valid RDF syntax does not necessarily guarantee semantic correctness or logical consistency with the chosen vocabulary (e.g., FOAF and Schema.org). For instance, using **foaf:knows** with a simple text string (instead of linking to another **foaf:Person**) does not impact syntactic validity, yet it signals potential issues with vocabulary use and data modelling. We will address these concerns in part B-D.

Therefore, the graph is syntactically valid RDF. However, this means that “valid RDF” is a fairly low bar – it just means that the structure follows the basic rules of RDF. It doesn’t mean the meaning is correct or the terms are being used as intended in other vocabularies, such as FOAF and Schema.org. The real problems lie in how effectively it represents the intended information using these vocabularies, where these issues will be addressed in the following questions.

---

## B. Are there any domain and range violations?

Yes, there are various domain and range violations. FOAF and Schema.org define expected “domains” (what class of resource can appear as a subject) and “ranges” (what class or datatype can appear as an object) for their properties:

### 1. **foaf:knows** used with a literal

- FOAF expects **foaf:knows** to link one **foaf:Person** to another **foaf:Person**. In the graph, Charlie **foaf:knows** **"Alice"^^xsd:string**. That is a domain/range violation

because **foaf:knows** is not meant to link a person to a string literal.

“Alice” should be a resource typed **foaf:Person**.

## 2. foaf:age with spelt-out string

- “**twenty-five**” is typed as an **xsd:string**, which is not typical.

Usually, **foaf:age** is expected to be a numeric value (ideally **xsd:integer**).

Similarly, “**thirty-five**” is also just a string with no datatype. This then leads to questionable usage.

## 3. schema:location with integers or untyped strings

- The diagram shows **schema:location** → “**New York**”^^**xsd:integer**. That is a mismatch because “**New York**” is a city name, not an integer. Also, “**San Francisco**” is an untyped string but might be used as if it were a **schema:Place**. The correct approach would typically be a resource typed **schema:Place** or a string typed as **xsd:string**.

## 4. schema:researchArea typed as an integer?

- There is a mention of “**Cybersecurity**”^^**xsd:integer** for Eve. That is a mismatch if we are trying to say the research area is “**Cybersecurity**.” Instead, a string literal or some resource should be used. Not an integer.

Hence, we see domain and range errors: the relevant vocabularies expect certain classes and datatypes, but the diagram violates those expectations.

---

# C. Errors in the class hierarchy or property declarations.

## 1. Person vs literal usage

- ‘Alice’ is never stated as a **foaf:Person**; rather, it is a literal. This means it cannot be reasonably asserted that **foaf:knows** is relating two people. We should come up with a resource like :Alice of a type **foaf:Person** and use **foaf:name** “**Alice**”^^**xsd:string**.

## 2. Confusion between FOAF and Schema.org

- Some **foaf:Person** are associated with schema: affiliation though it is a property of **schema:Person** → **schema: Organization**. Not every use is incompatible – some can be used in combination, but the domain and range must be appropriate in both.

## 3. Organisational classes

- “**Tech University**”, “**Cybersecurity Research Lab**”, “**University of AI**”, and “**Charlie Research Institute**” are presumably **schema:Organization**, but the diagram is not always strict on declaring all of them as **schema:Organization**. Some of them are classified, while others are not classified. It is recommended that each organisation entity should have **rdf:type schema:Organization**.

## 4. Property collisions/usage

- To give a name to a person, **foaf:name** is used in FOAF. To provide an age, **foaf:age** can be used as a simple literal (usually an integer).

Schema.org defines affiliation as **schema:affiliation**, which is a property of **schema:Person** and must take a **schema:Organization** as its value.

This leads to modelling errors if the subject or object is improperly typed.

In conclusion, several sources of confusion result from how classes and properties are defined or employed.

---

## D. Inferred facts and classification ambiguities

RDF reasoners can infer class memberships from domain/range constraints. For example:

- If **foaf:knows** has domain **foaf:Person** and range **foaf:Person**, then "**Alice**"<sup>^^xsd:string</sup> being the object would cause a reasoner to infer that "**Alice**"<sup>^^xsd:string</sup> is a person, which is contradictory (it's a literal, not a resource).
- If **schema:location** expects a **schema:Place** or at least a textual place name, then linking it to a typed integer "**New York**" is inconsistent. A reasoner might try to infer "**New York**" is a **schema:Place**, but the data literal says it is an integer, which is nonsensical for a location name.

Finally, some resources are untyped. For instance, "**San Francisco**" is just a string in the diagram. Is it a **schema:Place**? The graph is not explicit. The data is ambiguous.

- Several facts that should be inferred (like "Alice is a Person" or "New York is a Place") are blocked or contradicted by the modelling errors.
- The graph's classification of resources is incomplete/ambiguous, leaving open questions about each resource's nature (class).