

# Fourth Mandatory Assignment

IN4060

March 14, 2025

## 2 Entailment

### 2.1 sim:Marge rdf:type foaf:Person

The statement `sim:Marge rdf:type foaf:Person` entails the following:

1. `fam:hasSpouse` <sup>*aaa*</sup> `rdfs:range` <sup>*xxx*</sup> `foaf:Person` . - *P*
2. `sim:Homer` <sup>*yyy*</sup> `fam:hasSpouse` <sup>*aaa*</sup> `sim:Marge` <sup>*zzz*</sup> . - *P*
3. `sim:Marge` <sup>*zzz*</sup> `rdf:type` <sup>*xxx*</sup> `foaf:Person` . *rdfs3*, 1, 2

Pattern[1]	If <i>S</i> contains	then <i>S</i> RDFS entails recognizing <i>D</i>
<i>rdfs3</i>	<i>aaa</i> <code>rdfs:range</code> <i>xxx</i> . <i>yyy</i> <i>aaa</i> <i>zzz</i> .	<i>zzz</i> <code>rdf:type</code> <i>xxx</i> .

### 2.2 fam:hasSister rdfs:subPropertyOf fam:isRelativeOf

The statement `fam:hasSister rdfs:subPropertyOf fam:isRelativeOf` entails because, `fam:hasSister`  $\subseteq$  `fam:hasSibling`  $\subseteq$  `fam:isRelativeOf` as proven below:

1. `fam:hasSister` <sup>*xxx*</sup> `rdfs:subPropertyOf` `fam:hasSibling` <sup>*yyy*</sup> . - *P*
2. `fam:hasSibling` <sup>*yyy*</sup> `rdfs:subPropertyOf` `fam:isRelativeOf` <sup>*zzz*</sup> . - *P*
3. `fam:hasSister` <sup>*xxx*</sup> `rdfs:subPropertyOf` `fam:isRelativeOf` <sup>*zzz*</sup> . *rdfs5*, 1, 2

Pattern[1]	If <i>S</i> contains	then <i>S</i> RDFS entails recognizing <i>D</i>
<i>rdfs5</i>	<i>xxx</i> <code>rdfs:subPropertyOf</code> <i>yyy</i> . <i>yyy</i> <code>rdfs:subPropertyOf</code> <i>zzz</i> .	<i>xxx</i> <code>rdfs:subPropertyOf</code> <i>zzz</i> .

### 2.3 sim:Marge rdf:type fam:Woman

Based on the provided RDF graph, we cannot prove the statement `sim:Marge rdf:type fam:Woman` because even though we know that Lisa has two parents and her father is Homer, who is married to Marge; that does not prove that Lisa has a female parent or that Homer's spouse is a wife.

## 2.4 sim:Herb rdf:type fam:Man

The statement `sim:Herb rdf:type fam:Man` entails because:

1. `fam:hasBrother` <sup>*aaa*</sup> `rdfs:range` <sup>*xxx*</sup> `fam:Man` . - *P*
2. <sup>*yyy*</sup> `_:2` `fam:hasBrother` <sup>*aaa*</sup> `sim:Herb` <sup>*zzz*</sup> . - *P*
3. <sup>*zzz*</sup> `sim:Herb` `rdf:type` <sup>*xxx*</sup> `fam:Man` . - *rdfs3, 1, 2*

Pattern[1]	If <i>S</i> contains	then <i>S</i> RDFS entails recognizing <i>D</i>
<i>rdfs3</i>	<i>aaa</i> <code>rdfs:range</code> <i>xxx</i> . <i>yyy</i> <i>aaa</i> <i>zzz</i> .	<i>zzz</i> <code>rdf:type</code> <i>xxx</i> .

## 2.5 sim:Lisa fam:isRelativeOf sim:Homer

The statement `sim:Lisa fam:isRelativeOf sim:Homer` entails because,  
`fam:hasFather`  $\subseteq$  `fam:hasParent`  $\subseteq$  `fam:isRelativeOf` as proven below:

1. `fam:hasFather` <sup>*aaa*</sup> `rdfs:subPropertyOf` <sup>*bbb*</sup> `fam:isRelativeOf` . - *P*
2. <sup>*xxx*</sup> `sim:Lisa` `fam:hasFather` <sup>*aaa*</sup> `sim:Homer` <sup>*yyy*</sup> . - *P*
3. <sup>*xxx*</sup> `sim:Lisa` `fam:isRelativeOf` <sup>*bbb*</sup> `sim:Homer` <sup>*yyy*</sup> . - *rdfs7, 1, 2*

This works because `fam:hasFather` is a sub-property to `fam:isRelativeOf`:

1. `fam:hasFather` <sup>*xxx*</sup> `rdfs:subPropertyOf` <sup>*yyy*</sup> `fam:hasParent` . - *P*
2. <sup>*yyy*</sup> `fam:hasParent` `rdfs:subPropertyOf` <sup>*zzz*</sup> `fam:isRelativeOf` . - *P*
3. <sup>*xxx*</sup> `fam:hasFather` `rdfs:subPropertyOf` <sup>*zzz*</sup> `fam:isRelativeOf` . - *rdfs5, 1, 2*

Pattern[1]	If <i>S</i> contains	then <i>S</i> RDFS entails recognizing <i>D</i>
<i>rdfs5</i>	<i>xxx</i> <code>rdfs:subPropertyOf</code> <i>yyy</i> . <i>yyy</i> <code>rdfs:subPropertyOf</code> <i>zzz</i> .	<i>xxx</i> <code>rdfs:subPropertyOf</code> <i>zzz</i> .
<i>rdfs7</i>	<i>aaa</i> <code>rdfs:subPropertyOf</code> <i>bbb</i> . <i>xxx</i> <i>aaa</i> <i>yyy</i> .	<i>xxx</i> <i>bbb</i> <i>yyy</i> .

## 2.6 sim:Lisa fam:hasMother sim:Marge

The statement `sim:Lisa fam:hasMother sim:Marge` does not entail because according to the provided RDF graph, Lisa has two parents, but does not specify that one of them is a woman or a mother.

## 2.7 sim:Patty rdf:type foaf:Person

We already know that:

1. `_:1` `fam:hasSister` `sim:Patty` .
2. `fam:hasSister` `rdfs:subPropertyOf` `fam:hasSibling` .
3. `fam:isRelativeOf` `rdfs:range` `foaf:Person` .

Therefore, the statement `sim:Patty rdfs:type foaf:Person` entails because:

Starting with the *rdfs5* pattern:

1. `fam:hasSister rdfs:subPropertyOf fam:hasSibling . - P`
2. `fam:hasSibling rdfs:subPropertyOf fam:isRelativeOf . - P`
3. `fam:hasSister rdfs:subPropertyOf fam:isRelativeOf . - rdfs5, 1, 2`

This proves that we can replace `_:1 fam:hasSister sim:Patty .` with `_:1 fam:isRelativeOf sim:Patty .`, which we use with the *rdfs3* pattern:

1. `fam:isRelativeOf rdfs:range foaf:Person . - P`
2. `_:1 fam:isRelativeOf sim:Patty . - P`
3. `sim:Patty rdfs:type foaf:Person . - rdfs3, 1, 2`

Pattern[1]	If <i>S</i> contains	then <i>S</i> RDFS entails recognizing <i>D</i>
rdfs3	<code>aaa rdfs:range xxx .</code> <code>yyy aaa zzz .</code>	<code>zzz rdfs:type xxx .</code>
rdfs5	<code>xxx rdfs:subPropertyOf yyy .</code> <code>yyy rdfs:subPropertyOf zzz .</code>	<code>xxx rdfs:subPropertyOf zzz .</code>

## 2.8 `_:a fam:hasParent _:b .` `_:b fam:hasSister sim:Patty .`

The statement is satisfactory due to blank node equivalence as illustrated in the RDF graph:

1. `sim:Lisa fam:hasParent _:1 .`
2. `_:1 fam:hasSister sim:Patty .`

In other words, it entails without the use of RDFS patterns.

## 2.9 `_:d fam:hasBrother _:e .` `_:d fam:hasBrother _:f .`

The statement `_:d fam:hasBrother _:e .` `_:d fam:hasBrother _:f .` is entailed as illustrated by the RDF graph:

1. `_:2 fam:hasBrother sim:Herb .`
2. `_:2 fam:hasBrother sim:Herb .`

We know from the RDF graph that a blank node resource has a brother `sim:Herb`, and even though the two triples in the statement refer to two different blank nodes `_:e` and `_:f`, they can still be the same person `sim:Herb` due to non-unique name assumption.

## References

- [1] Patrick Hayes, Florida IHMC, and Peter Patel-Schneider (eds.) *RDF 1.1 Semantics*. World Wide Web Consortium. [Online; accessed 13-March-2025]. Feb. 2014. URL: <https://www.w3.org/TR/rdf11-mt/#patterns-of-rdfs-entailment-informative>.