

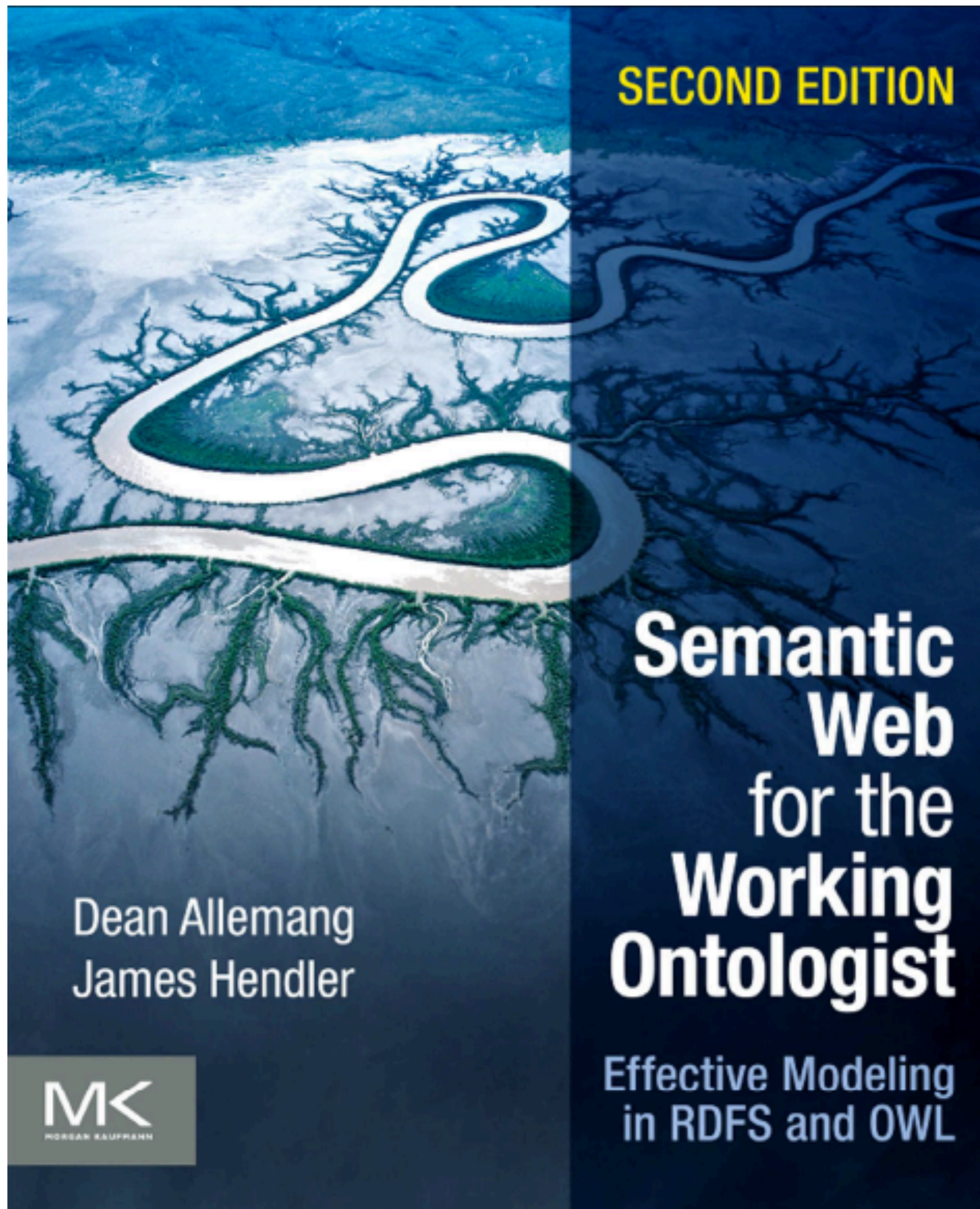
Ingeniería y Gestión del Conocimiento RDFS

Profesor
Andrés Melgar

Agenda

- Inferencia en RDF (subPropertyOf).
- Dominio y rango.
- Unión de conjuntos.
- Intersección de conjuntos.

Texto guía



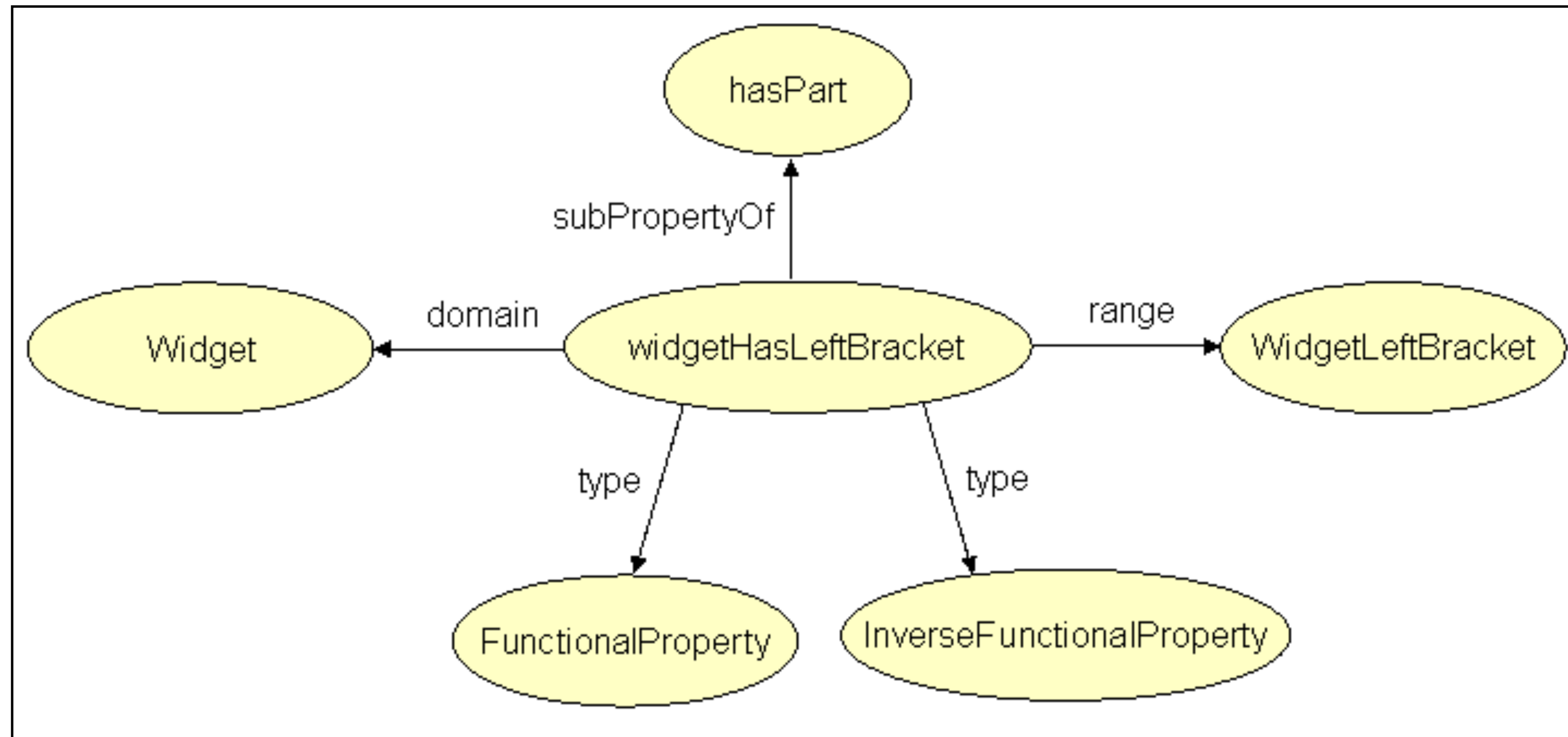
CHAPTER

RDF schema

7

Inferencia en RDF

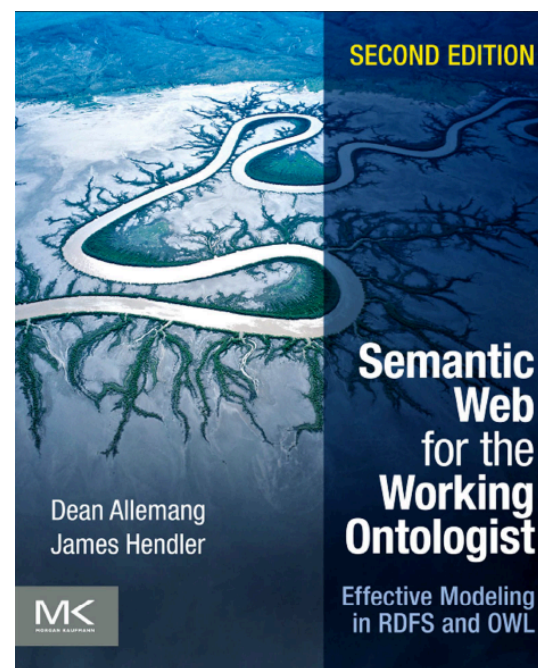
subPropertyOf



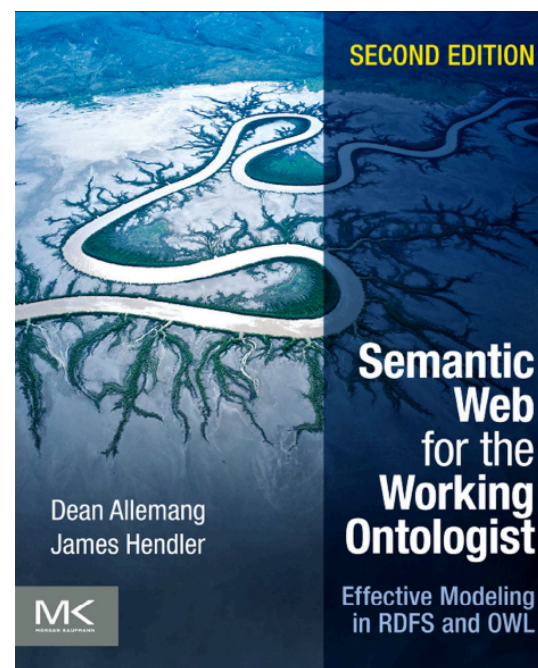
Inferencia en RDF

- La intuición básica detrás del uso de `rdfs:subPropertyOf` es que una terminología incluye tanto verbos como sustantivos, y que muchos de los mismos requisitos para mapear sustantivos de una fuente a otra se aplicarán a las relaciones.
- Los ejemplos simples abundan en el lenguaje ordinario.
 - La relación **hermano** (*brother*) es más específica que la relación **pariente** (*sibling*); si alguien es mi **hermano** (*brother*), entonces él también es mi **pariente** (*sibling*).
- Esto se formaliza en RDFS con `rdfs:subPropertyOf`.

```
CONSTRUCT {?x ?r ?y .}  
WHERE {?x ?q ?y .  
       ?q rdfs:subPropertyOf ?r }
```



Inferencia en RDF



- Una gran empresa involucra a varias personas en diversas capacidades y tiene una variedad de formas de administrar estas relaciones.
- Algunas personas son empleadas directamente por la empresa, mientras que otras son contratistas.
- Entre estos contratistas,
 - algunos de ellos están directamente contratados por la empresa como freelance,
 - otros tienen un contrato a largo plazo y a través de una empresa intermedia.
- Se podría decir que todas estas personas trabajan para la empresa.

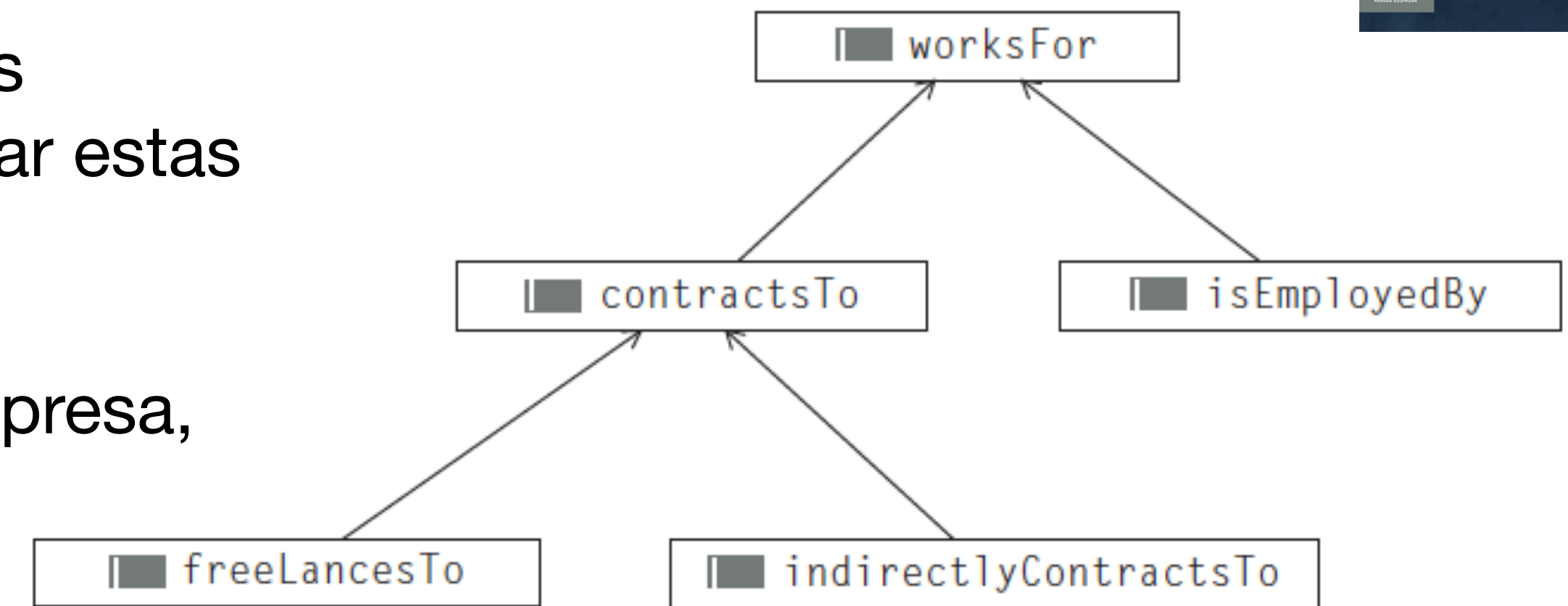
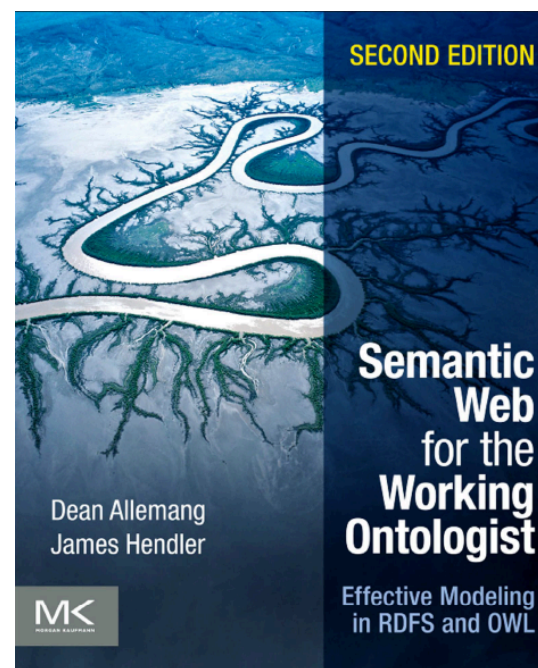


FIGURE 7.1

`rdfs:subPropertyOf` relations for workers in the firm.

Inferencia en RDF



- ¿Cómo se puede modelar esta situación en RDFS?
- En primer lugar, se debe considerar las inferencias que deseamos extraer y en qué circunstancias.
- Hay una serie de relaciones que pueden sostenerse entre una persona y la empresa:
 - contractTo
 - freeLancesTo
 - indirectlyContractsTo
 - isEmployedBy
 - worksFor.

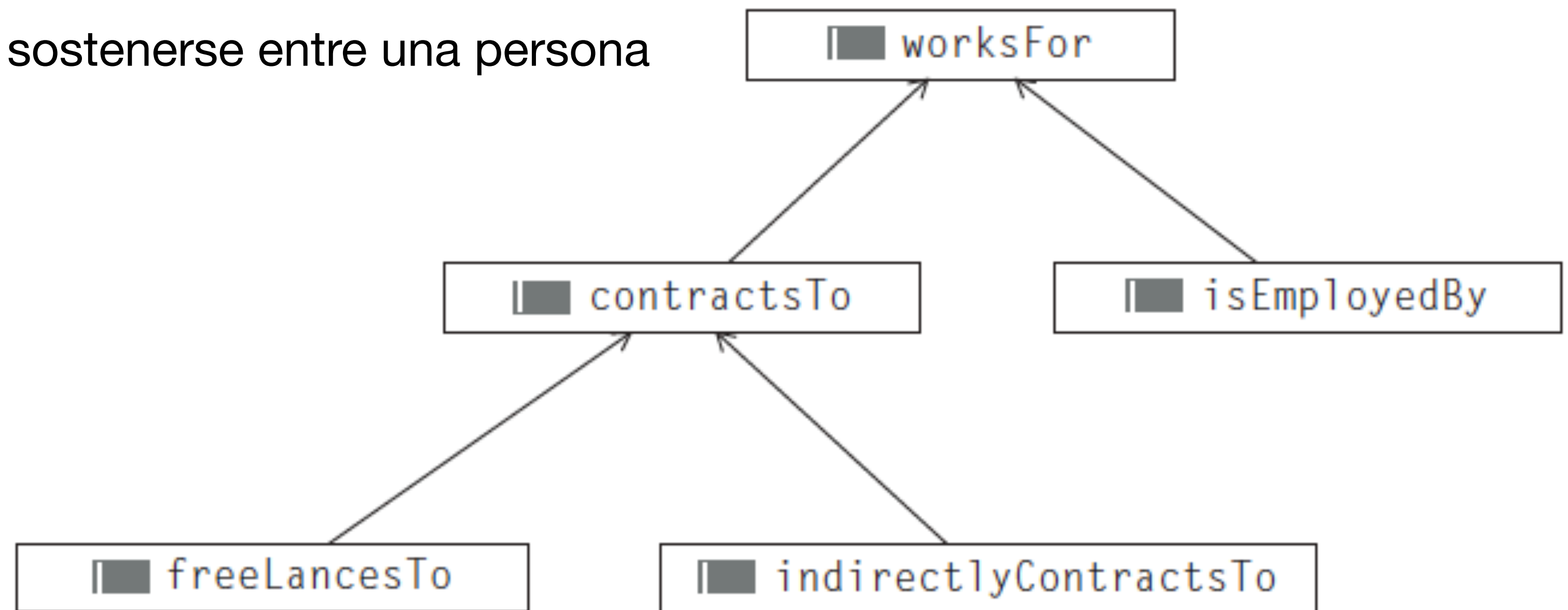
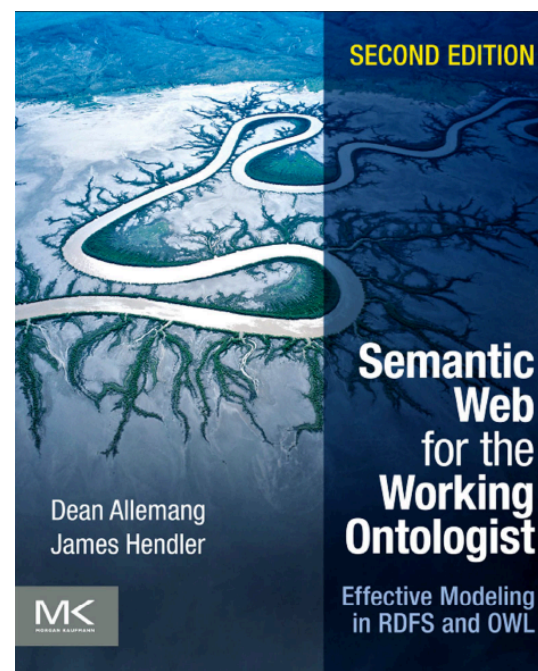


FIGURE 7.1

`rdfs:subPropertyOf` relations for workers in the firm.

Inferencia en RDF



- Si afirmamos alguna de estas declaraciones sobre alguna persona, entonces podremos inferir que esa persona trabaja (*worksFor*) para la empresa.
- Además, hay conclusiones intermedias que podemos extraer, por ejemplo:
 - Tanto un profesional *freeLance* como un contratista indirecto contratan (*contractsTo*) con la empresa y, de hecho, trabajan (*worksFor*) para la empresa.

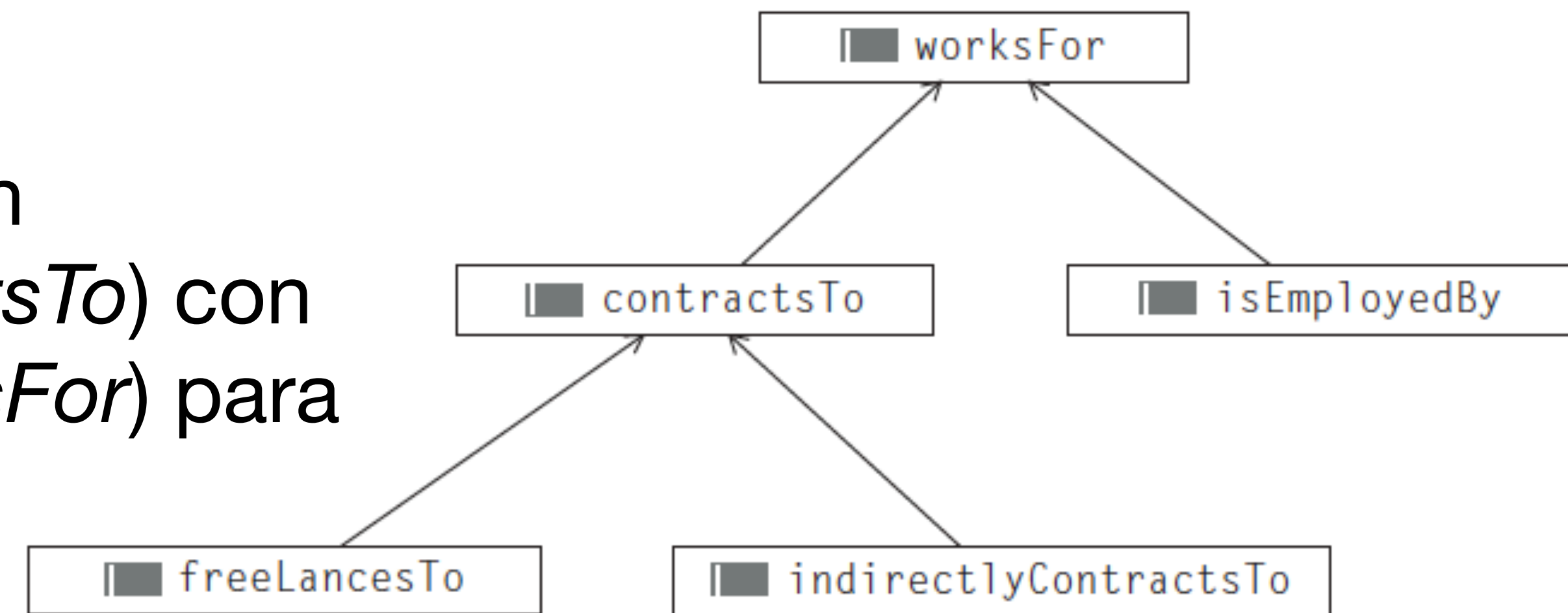


FIGURE 7.1

`rdfs:subPropertyOf` relations for workers in the firm.

Inferencia en RDF

```
:freeLancesTo rdfs:subPropertyOf contractsTo.  
:indirectlyContractsTo rdfs:subPropertyOf contractsTo.  
:isEmployedBy rdfs:subPropertyOf worksFor.  
:contractsTo rdfs:subPropertyOf worksFor.
```

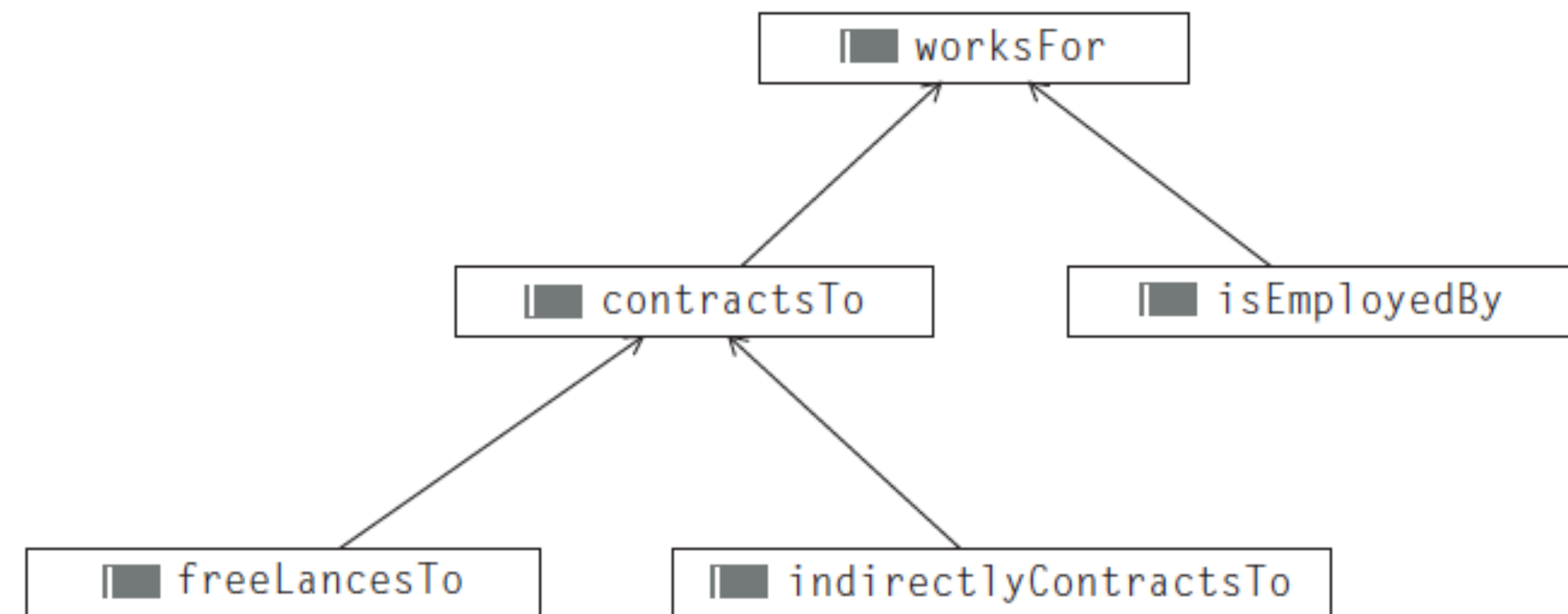
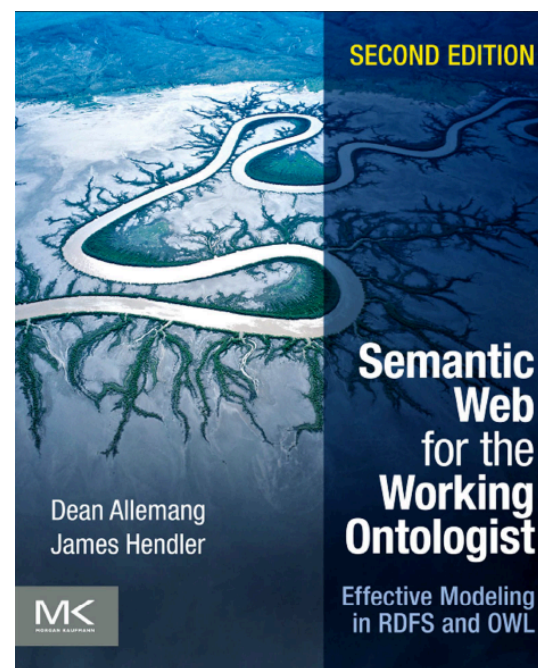


FIGURE 7.1

`rdfs:subPropertyOf` relations for workers in the firm.

Inferencia en RDF



- Dado los siguientes datos

```
:Goldman :isEmployedBy :TheFirm.  
:Spence :freeLancesTo :TheFirm.  
:Long :indirectlyContractsTo :TheFirm.
```

- Se puede inferir que:

```
:Goldman :worksFor :TheFirm.
```

```
:Spence :contractsTo :TheFirm.  
:Long contractsTo :TheFirm.
```

```
:Spence :worksFor :TheFirm.  
:Long :worksFor :TheFirm.
```

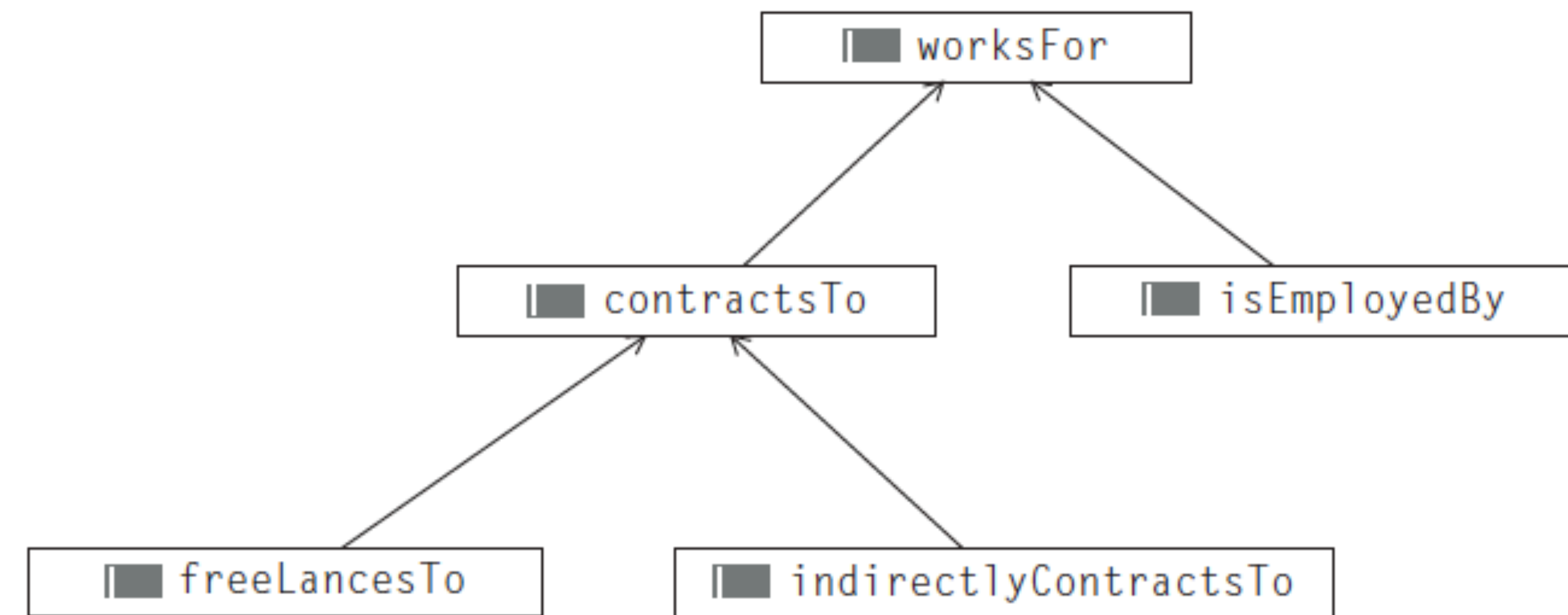
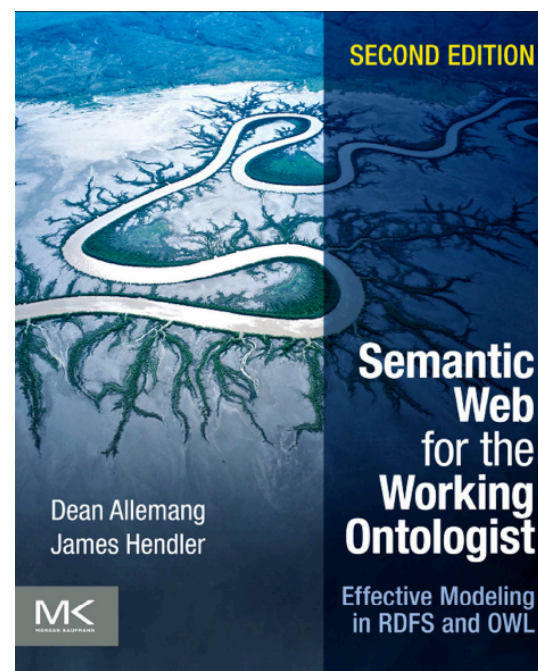


FIGURE 7.1

`rdfs:subPropertyOf` relations for workers in the firm.

Inferencia en RDF



- En general, `rdfs:subPropertyOf` permite a un modelador describir una **jerarquía** de propiedades relacionadas.
- Al igual que en las jerarquías de clase, las propiedades específicas se encuentran en la parte inferior del árbol, y las propiedades más generales están más arriba en el árbol.
- Cada vez que una propiedad en el árbol se mantiene entre dos entidades, también lo hace cada propiedad encima de ella.

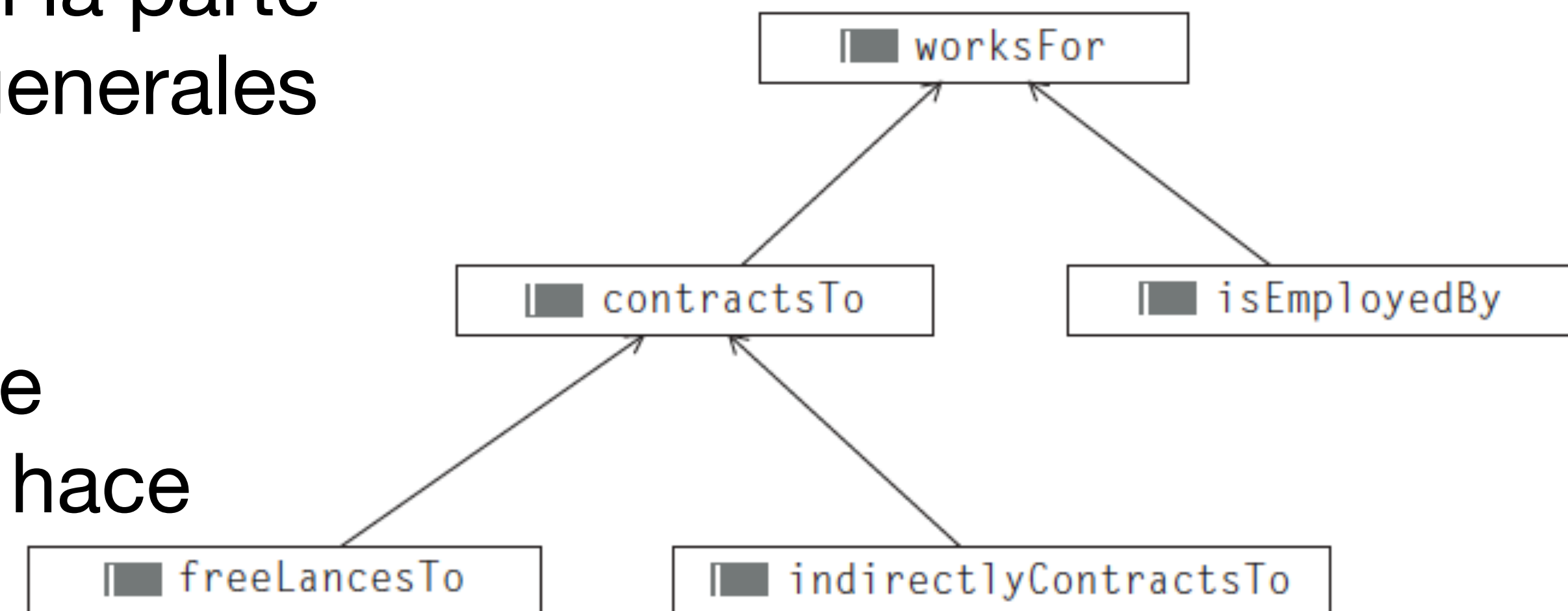
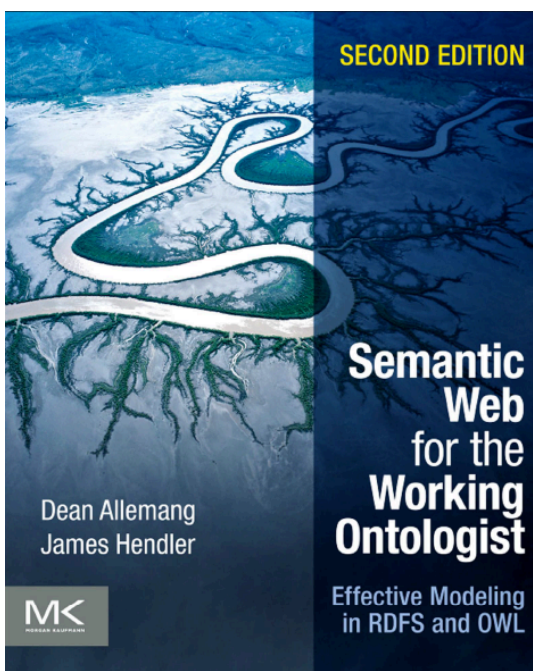


FIGURE 7.1

`rdfs:subPropertyOf` relations for workers in the firm.

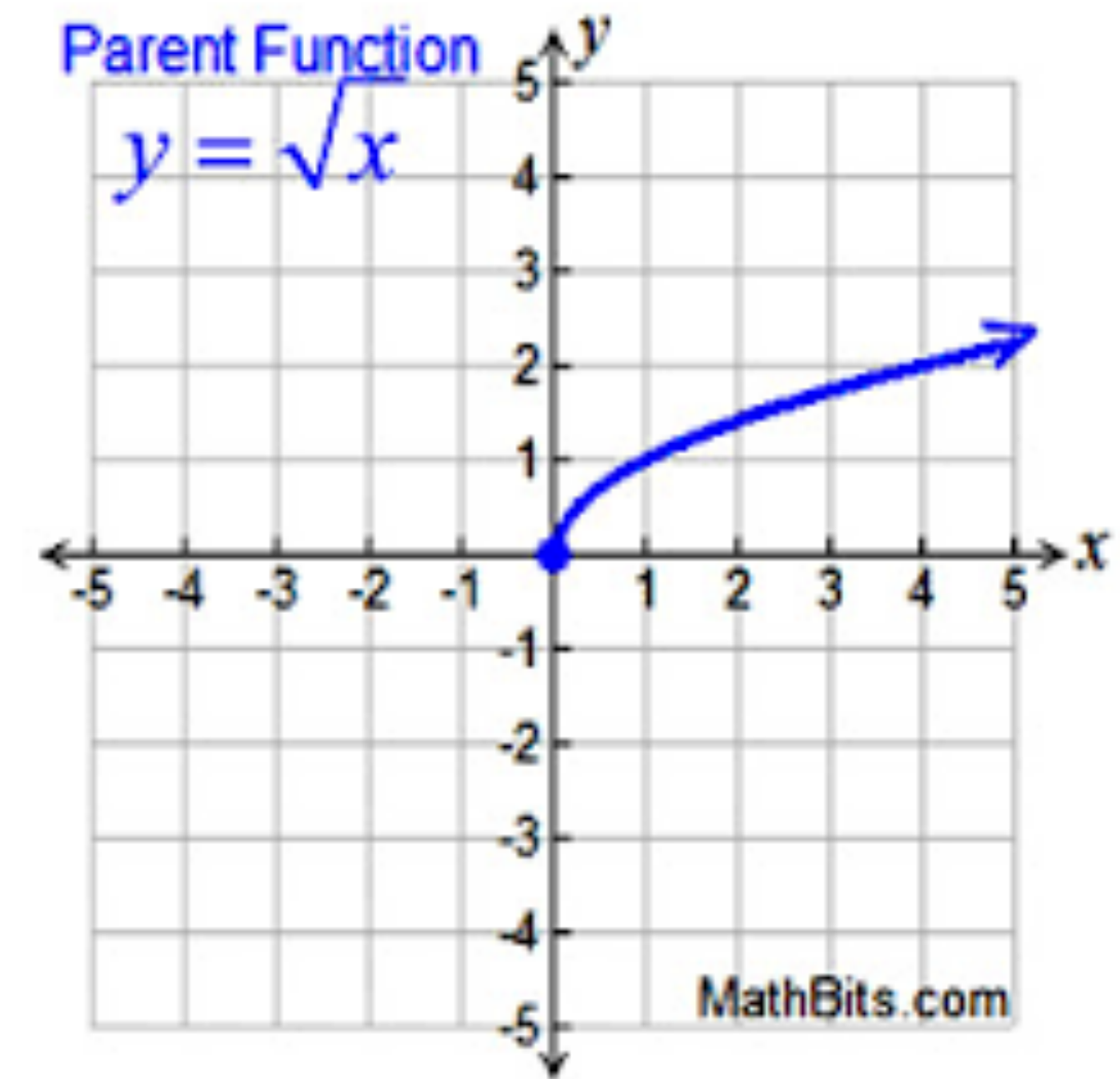
Dominio y rango



- Se ha visto cómo las inferencias alrededor de `rdfs:subPropertyOf` se puede utilizar para describir cómo dos propiedades se relacionan entre sí.
- Pero cuando se describe el uso de términos en los datos, también nos gusta representar cómo se usa una **propiedad** en **relación** con las **clases** definidas.
- En particular, podríamos decir que:
 - cuando se usa una propiedad, el sujeto de la tripla proviene de (es decir, tiene `rdf:type`) una cierta clase y
 - que el objeto proviene de algún otro tipo.
- Estas dos estipulaciones se expresan en RDFS con los recursos (palabras clave) `rdfs:domain` y `rdfs:range`, respectivamente.

Dominio y rango

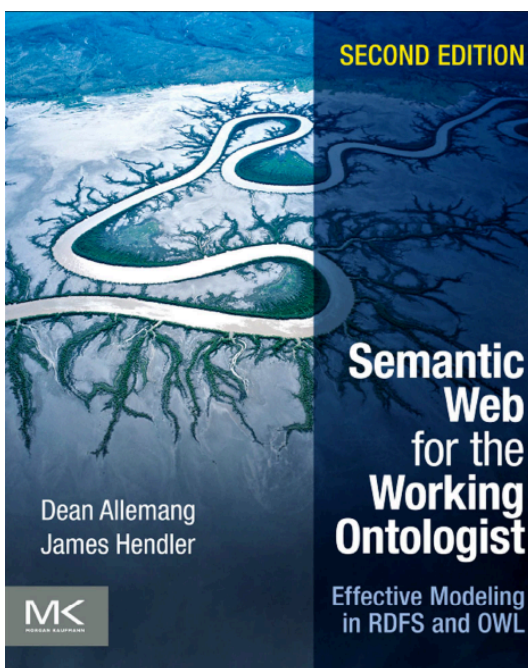
- En matemáticas, las palabras dominio y rango se usan para referirse a cómo se puede usar una función (o más generalmente, una relación).
- El **dominio** de una función es el conjunto de valores para el que está definido.
- El **rango** es el conjunto de valores que puede tomar.
- Por ejemplo, la relación raíz cuadrada:
 - tiene los números positivos como el dominio (ya que los números negativos no tienen raíces cuadradas en los reales),
 - y los números positivos como el rango.



Dominio y rango

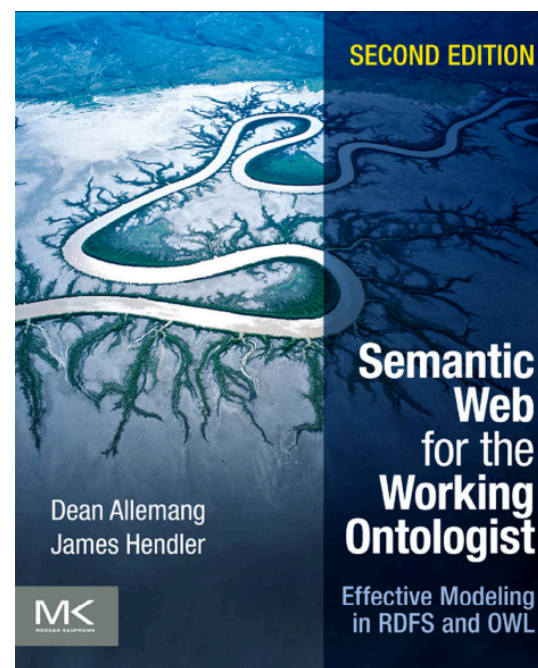
- En RDFS, las propiedades `rdfs:domain` y `rdfs:range` tienen significados inspirados en los usos matemáticos de estas palabras.
- Una propiedad `p` puede tener un `rdfs:domain` y un `rdfs:range`.
- Estos se especifican, como todo en RDF, a través de triplas:

```
:p rdfs:domain :D.  
:p rdfs:range :R.
```

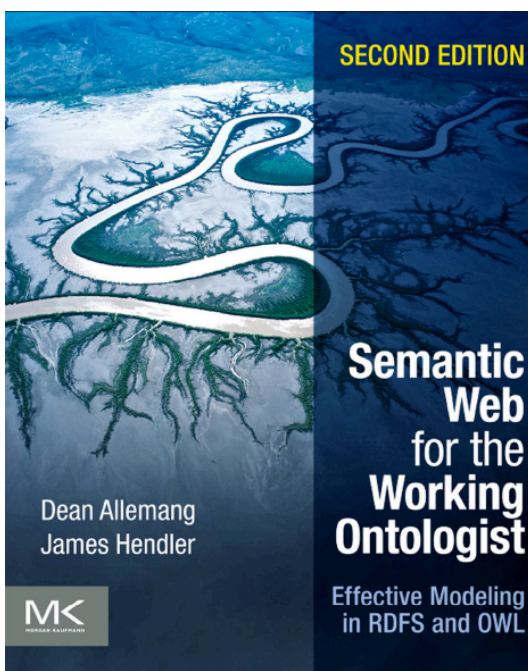


Dominio y rango

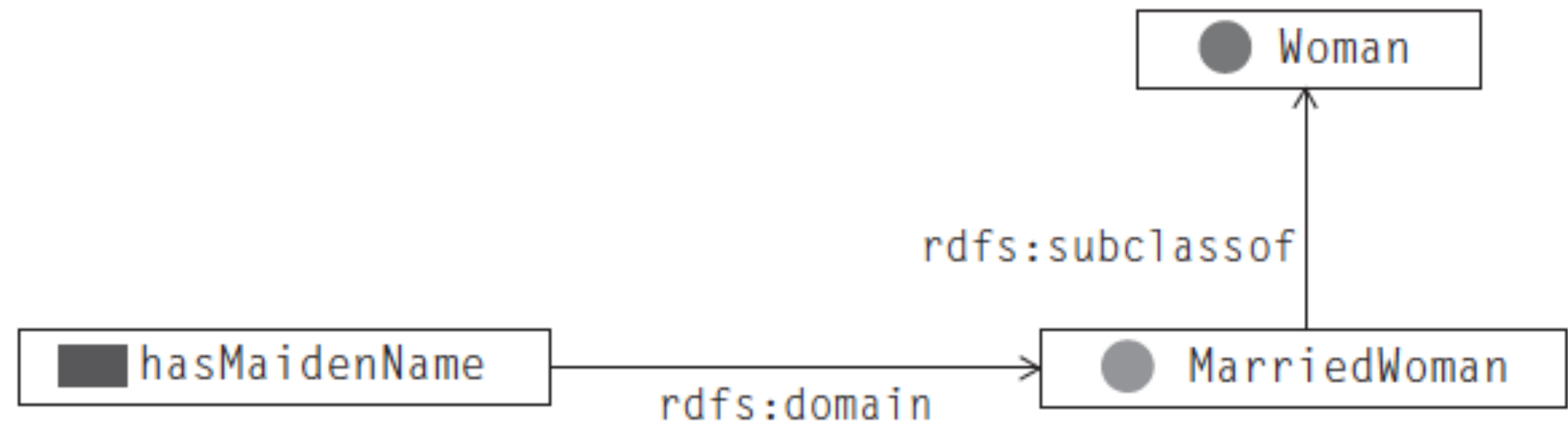
- En RDFS, el dominio y rango proporcionan cierta información sobre cómo se usará la propiedad P;
 - **dominio** se refiere al sujeto de cualquier tripleta que usa P como su predicado,
 - **rango** se refiere al objeto de dicha tripleta.
- Cuando afirmamos que la propiedad P tiene un dominio D (respectivamente, rango R), estamos diciendo que siempre que usemos la propiedad P, podemos inferir que el sujeto (respectivamente objeto) de esta tripla es un miembro de la clase D (respectivamente R)
- En resumen, el dominio y el rango nos dicen cómo se usará P.
 - En lugar de señalar un error si P se utiliza de una manera aparentemente inconsistente con esta declaración, RDFS deducirá la información de tipo necesaria para que P cumpla con sus declaraciones de dominio y rango.



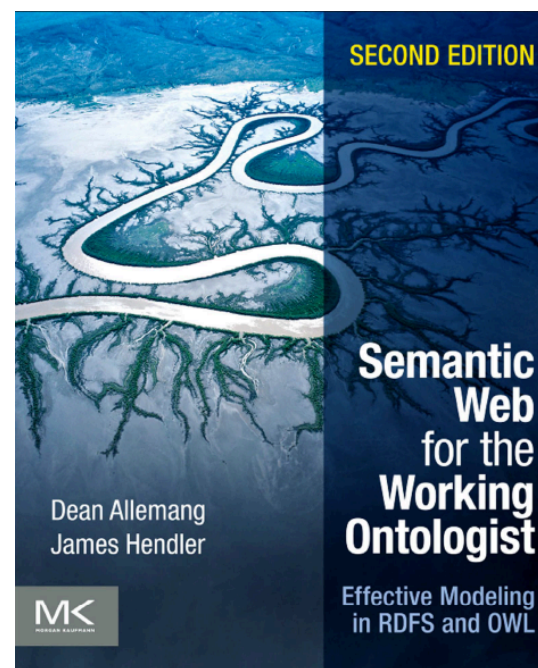
Inferencia en RDF



```
:MarriedWoman rdfs:subClassOf :Woman.  
:hasMaidenName rdfs:domain :MarriedWoman.  
:Karen :hasMaidenName "Stephens".
```



Inferencia en RDF



```
:MarriedWoman rdfs:subClassOf :Woman.
```

```
:hasMaidenName rdfs:domain :MarriedWoman.
```

```
:Karen :hasMaidenName "Stephens".
```

```
?X :hasMaidenName ?Y .
```

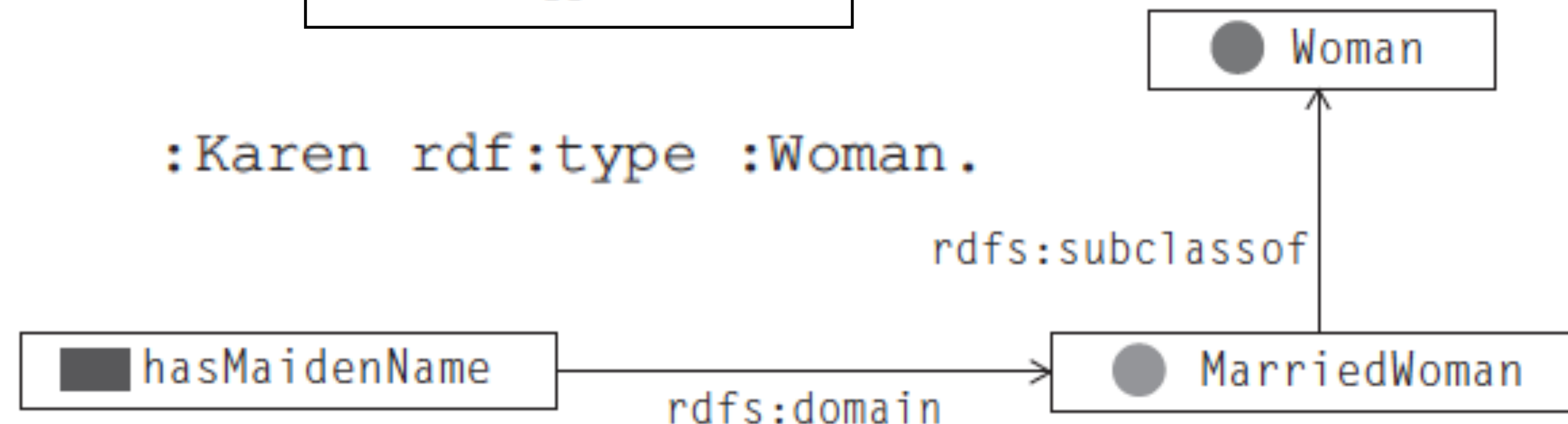
Se infiere

```
:hasMaidenName rdfs:domain :Woman.
```

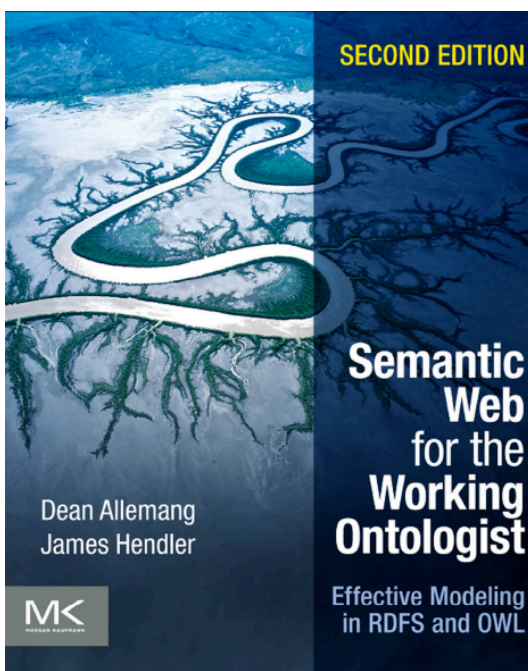
```
:Karen rdf:type :MarriedWoman.
```

```
?X rdf:type :Woman.
```

```
:Karen rdf:type :Woman.
```



Intersección de conjuntos



```
:C rdfs:subClassOf :A.  
:C rdfs:subClassOf :B.
```

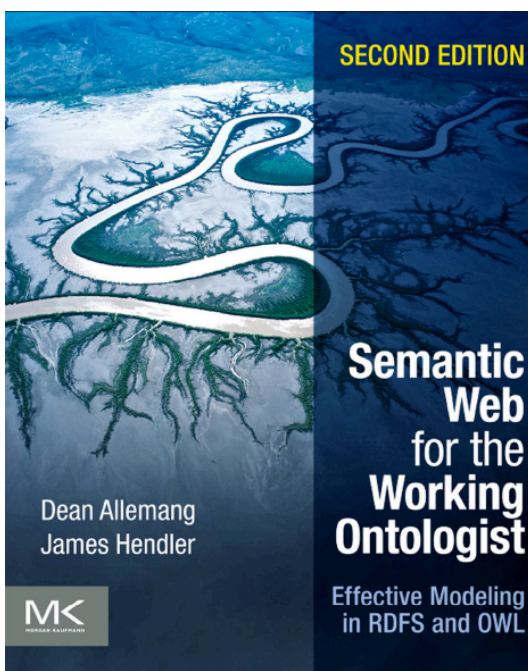
```
?x rdf:type :C.
```

We can infer

```
?x rdf:type :B.
```

```
?x rdf:type :A.
```

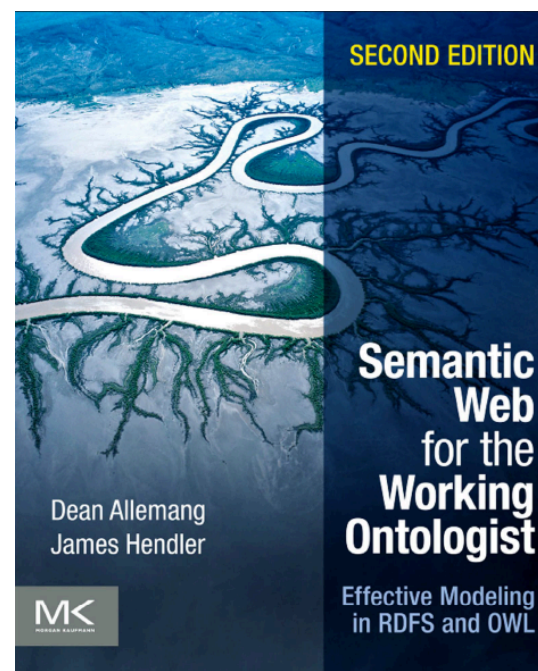
Unión de conjuntos



```
:A rdfs:subClassOf :C.  
:B rdfs:subClassOf :C.
```

```
?x rdf:type :A.  
or  
?x rdf:type :B.  
implies  
?x rdf:type :C.
```

Inferencia en conjuntos



- Dado los siguientes datos:

```
:Surgeon rdfs:subClassOf :Staff.  
:Surgeon rdfs:subClassOf :Physician.  
:Kildare rdf:type :Surgeon.
```

- Se infiere que:

```
:Kildare rdf:type :Staff.  
:Kildare rdf:type :Physician.
```

- $C \subseteq A \cap B$ (by making `C rdfs:subClassOf` both `A` and `B`)
- $C \supseteq A \cup B$ (by making both `A` and `B rdfs:subClassOf C`).

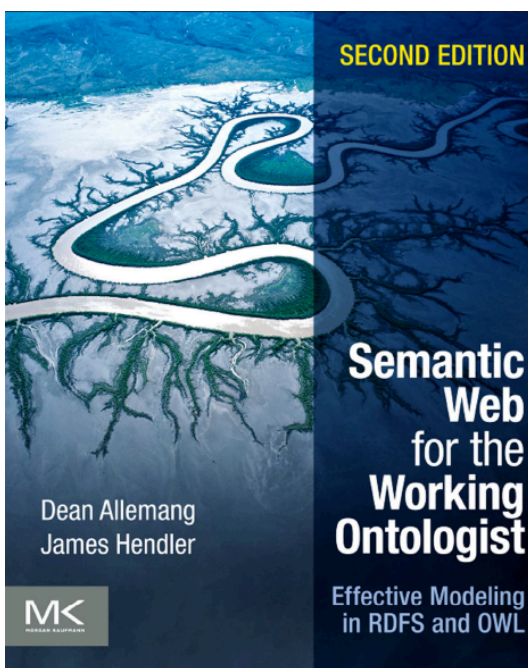
Inferencia en propiedades

- Dado los siguientes datos:

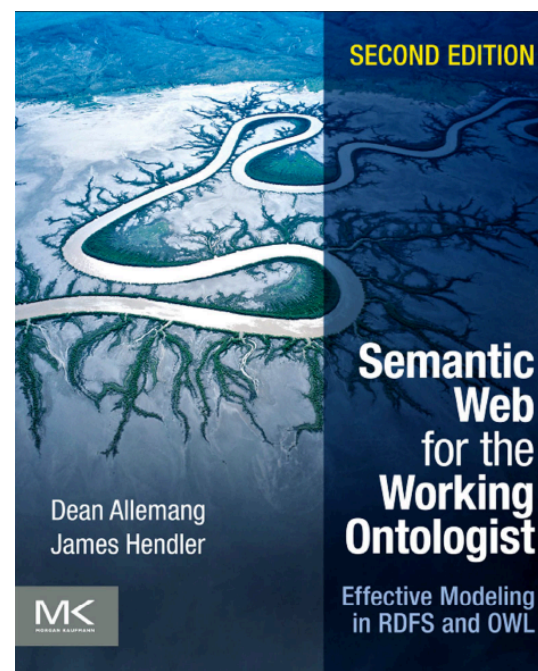
```
:lodgedIn rdfs:subPropertyOf :billedFor.  
:lodgedIn rdfs:subPropertyOf :assignedTo.  
  
:Marcus :lodgedIn :Room101.
```

- Se infiere que:

```
:Marcus :billedFor :Room101.  
:Marcus :assignedTo :Room101.
```



Inferencia en conjuntos



- Dado los siguientes datos:

```
:MVP rdfs:subClassOf :AllStarCandidate.  
:TopScorer rdfs:subClassOf :AllStarCandidate.  
  
:Reilly rdf:type :MVP.  
:Kaneda rdf:type :TopScorer.
```

- Se infiere que:

```
:Reilly rdf:type :AllStarCandidate.  
:Kaneda rdf:type :AllStarCandidate.
```

- $C \subseteq A \cap B$ (by making `C rdfs:subClassOf` both `A` and `B`)
- $C \supseteq A \cup B$ (by making both `A` and `B rdfs:subClassOf` `C`).