SAPIENZA
UNIVERSITÀ DI ROMA

Verifica automatica di proprietà su ontologie DL-Lite$_A$

Ilaria Forte
Valerio Del Grande

Lo scopo del nostro lavoro è mostrare come sia possibile effettuare la verifica di proprietà statiche su un ontologia, al fine di rilevare errori nella progettazione concettuale.

Lo scopo del nostro lavoro è mostrare come sia possibile effettuare la verifica di proprietà statiche su un ontologia, al fine di rilevare errori nella progettazione concettuale.

Ci concentreremo su ontologie espresse nella logica DL-LiteA

Verifica automatica di proprietà su ontologie DL-Lite$_A$

MOTIVAZIONI:
allo stato attuale non esistono strumenti di reasoning intensionale su DL-LiteA

IDEA: DL-LiteA è un sottoinsieme decidibile della FOL

QUINDI: possiamo tradurre un'ontologia espressa in DL-LiteA in una base di conoscenza della FOL

VANTAGGI: possiamo utilizzare strumenti software preesistenti come Otter e mace per verificare l'implicazione logica di proprietà qualsiasi esprimibili in FOL

Ontologia

Un'ontologia è una concettualizzazione del dominio di interesse di un sistema informativo espressa in un qualche linguaggio formale

Description Logics

Sono una famiglia di formalismi particolarmente adatti per rappresentare conoscenza strutturata e ragionare su di essa

DL-LiteA

Sviluppata presso il DIS è, tra tutte le Description Logics, un formalismo pensato appositamente per catturare modelli concettuali di dati.

Il trade-off tra potere espressivo e complessità computazionale (nella dimensione dei dati) la rende l'unico formalismo realmente interessante a livello pratico in sistemi OBDA che lavorano su grandi quantità di dati.

Il query answering su un'ontologia DL-LiteA è LOGSPACE nella dimensione dei dati (come SQL!)

Ontologia DL-LiteA

E' composta da:

TBox:
insieme di formule che rappresentano il livello intensionale

ABox:
insieme di formule che rappresentano il livello estensionale

TBox: elementi

- Concepts: rappresentano una collezione di oggetti dello stesso tipo
- Value-Domains: denota un insieme di valori
- Concept Attributes: relazione binaria tra un concept e un valore
- Roles: relazione binaria tra due concepts
- Role Attributes: relazione ternaria tra due concepts e un valore

- Intensional assertion: esprime una condizione a livello intensionale che deve essere soddisfatta a livello estensionale

TBox: elementi

- Concepts: rappresentano una collezione di oggetti dello stesso tipo
- Value-Domains: denota un insieme di valori
- Concept Attributes: relazione binaria tra un concept e un valore
- Roles: relazione binaria tra due concepts
- Role Attributes: relazione ternaria tra due concepts e un valore

- Intensional assertion: esprime una condizione a livello intensionale che deve essere soddisfatta a livello estensionale

Una TBox è un insieme finito di *intensional assertions*

TBox: intensional assertions

Inclusion Assertions:

$$
\begin{array}{rcll}
B & \sqsubseteq & C & (concept\ inclusion\ assertion) \\
Q & \sqsubseteq & R & (role\ inclusion\ assertion) \\
E & \sqsubseteq & F & (value\text{-}domain\ inclusion\ assertion) \\
U_C & \sqsubseteq & V_C & (concept\ attribute\ inclusion\ assertion) \\
U_R & \sqsubseteq & V_R & (role\ attribute\ inclusion\ assertion)
\end{array}
$$

Functionality assertions:

$$
\begin{array}{ll}
(funct\ P) & (role\ functionality\ assertion) \\
(funct\ P^-) & (inverse\ role\ functionality\ assertion) \\
(funct\ U_C) & (concept\ attribute\ functionality\ assertion) \\
(funct\ U_R) & (role\ attribute\ functionality\ assertion)
\end{array}
$$

Identification assertions:

$$
(id\ B\ I_1, ..., I_n) \qquad (identification\ assertion)
$$

TBox: espressioni

1. Concept expressions:

$$B \ ::= \ A \mid \exists Q \mid \delta(U_C)$$

$$C \ ::= \ \top_C \mid B \mid \neg B$$

2. Value-domain expressions:

$$E \ ::= \ \rho(U_C) \mid \rho(U_R)$$

$$F \ ::= \ \top_D \mid T_1 \mid \cdots \mid T_n$$

3. Role expressions:

$$Q \ ::= \ P \mid P^- \mid \delta(U_R) \mid \delta(U_R)^-$$

$$R \ ::= \ Q \mid \neg Q$$

4. Attribute expressions:

$$V_C \ ::= \ U_C \mid \neg U_C$$

$$V_R \ ::= \ U_R \mid \neg U_R$$

TBox: espressioni

1. Concept expressions:

$$B \ ::= A \mid \exists Q \mid \delta(U_C)$$

$$C \ ::= \ \top_C \mid B \mid \neg B$$

2. Value-domain expressions:

$$E \ ::= \ \rho(U_C) \mid \rho(U_R)$$

$$F \ ::= \ \top_D \mid T_1 \mid \cdots \mid T_n$$

3. Role expressions:

$$Q \ ::= \ P \mid P^- \mid \delta(U_R) \mid \delta(U_R)^-$$

$$R \ ::= \ Q \mid \neg Q$$

4. Attribute expressions:

$$V_C \ ::= \ U_C \mid \neg U_C$$

$$V_R \ ::= \ U_R \mid \neg U_R$$

TBox: espressioni

1. Concept expressions:

$$B \ ::= A \mid \exists Q \mid \delta(U_C)$$

$$C \ ::= \ \top_C \mid B \mid \neg B$$

2. Value-domain expressions:

$$E \ ::= \ \rho(U_C) \mid \rho(U_R)$$

$$F \ ::= \ \top_D \mid T_1 \mid \cdots \mid T_n$$

3. Role expressions:

$$Q \ ::= \ P \mid P^- \mid \delta(U_R) \mid \delta(U_R)^-$$

$$R \ ::= \ Q \mid \neg Q$$

4. Attribute expressions:

$$V_C \ ::= \ U_C \mid \neg U_C$$

$$V_R \ ::= \ U_R \mid \neg U_R$$

TBox: espressioni

1. Concept expressions:

$$B \ ::= A \mid \exists Q \mid \delta(U_C)$$

$$C \ ::= \ \top_C \mid B \mid \neg B$$

2. Value-domain expressions:

$$E \ ::= \ \rho(U_C) \mid \rho(U_R)$$

$$F \ ::= \ \top_D \mid T_1 \mid \cdots \mid T_n$$

3. Role expressions:

$$Q \ ::= P \mid P^- \mid \delta(U_R) \mid \delta(U_R)^-$$

$$R \ ::= \ Q \mid \neg Q$$

4. Attribute expressions:

$$V_C \ ::= \ U_C \mid \neg U_C$$

$$V_R \ ::= \ U_R \mid \neg U_R$$

TBox: espressioni

1. Concept expressions:

$$B \ ::= A \mid \exists Q \mid \delta(U_C)$$

$$C \ ::= \ \top_C \mid B \mid \neg B$$

2. Value-domain expressions:

$$E \ ::= \ \rho(U_C) \mid \rho(U_R)$$

$$F \ ::= \ \top_D \mid T_1 \mid \cdots \mid T_n$$

3. Role expressions:

$$Q \ ::= P \mid P^- \mid \delta(U_R) \mid \delta(U_R)^-$$

$$R \ ::= \ Q \mid \neg Q$$

4. Attribute expressions:

$$V_C \ ::= U_C \mid \neg U_C$$

$$V_R \ ::= \ U_R \mid \neg U_R$$

TBox: espressioni

1. Concept expressions:

$$B \ ::= A \mid \exists Q \mid \delta(U_C)$$

$$C \ ::= \ \top_C \mid B \mid \neg B$$

2. Value-domain expressions:

$$E \ ::= \ \rho(U_C) \mid \rho(U_R)$$

$$F \ ::= \ \top_D \mid T_1 \mid \cdots \mid T_n$$

3. Role expressions:

$$Q \ ::= P \mid P^- \mid \delta(U_R) \mid \delta(U_R)^-$$

$$R \ ::= \ Q \mid \neg Q$$

4. Attribute expressions:

$$V_C \ ::= U_C \mid \neg U_C$$

$$V_R \ ::= U_R \mid \neg U_R$$

ABox: membership assertions

- $A(a)$
- $P(a, b)$
- $U_C(a, c)$
- $U_R(a, b, c)$

dove a e b sono costanti dell'alfabeto in Γ_O e c è una costante in Γ_V.

ABox: membership assertions

- $A(a)$
- $P(a, b)$
- $U_C(a, c)$
- $U_R(a, b, c)$

dove a e b sono costanti dell'alfabeto in Γ_O e c è una costante in Γ_V.

Una ABox è un insieme finito di *membership assertions*

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

name è attributo di person di tipo string

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

ogni persona ha un nome

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

ogni persona ha un solo nome

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

cityOfBirth è un role il cui dominio è Person e codominio City

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

ogni persona ha almeno una città di nascita

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

ogni persona ha al più una città di nascita

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

dateOfBirth è attributo del role di tipo date

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

dateOfBirth è attributo obbligatorio

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

dateOfBirth è attributo funzionale

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

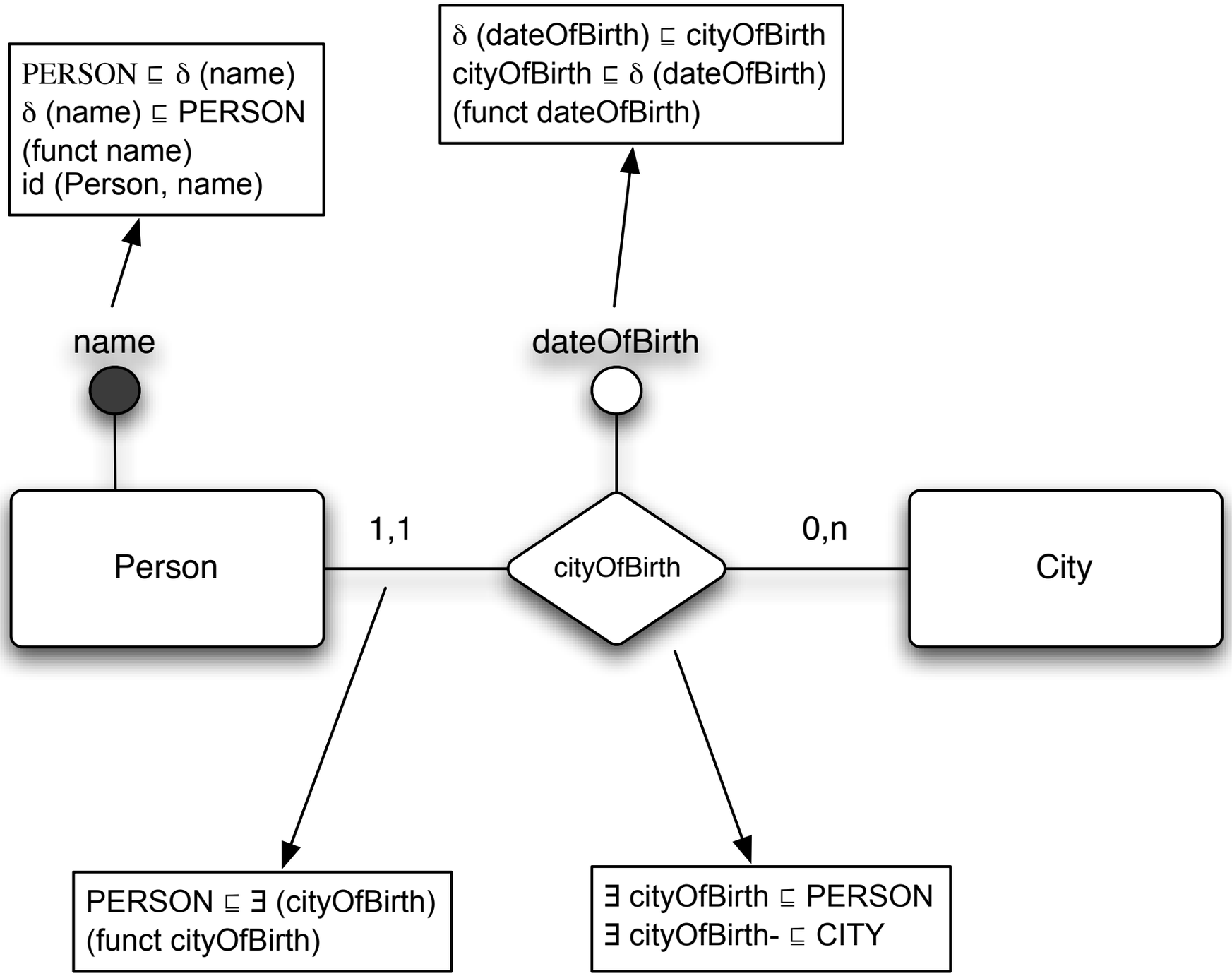
ogni istanza di persona è identificata dall'attributo name

TBox: un esempio

$$\text{Atomic Concepts} = \{Person, City\}$$
$$\text{Atomic Concept Attributes} = \{name\}$$
$$\text{Atomic Roles} = \{cityOfBirth\}$$
$$\text{Atomic Role Attributes} = \{dateOfBirth\}$$
$$\text{Value Domains} = \{String, Date\}$$

$$\delta(name) \sqsubseteq Person \quad (1)$$
$$\rho(name) \sqsubseteq xsd : string \quad (2)$$
$$Person \sqsubseteq \delta(name) \quad (3)$$
$$(funct\ name) \quad (4)$$
$$\exists cityOfBirth \sqsubseteq Person \quad (5)$$
$$\exists cityOfBirth^{-} \sqsubseteq City \quad (6)$$
$$Person \sqsubseteq \exists cityOfBirth \quad (7)$$
$$(funct\ cityOfBirth) \quad (8)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth \quad (9)$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date \quad (10)$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth) \quad (11)$$
$$(funct\ dateOfBirth) \quad (12)$$
$$id(Person, name) \quad (13)$$

TBox vs ER


PERSON ⊑ δ (name)
δ (name) ⊑ PERSON
(funct name)
id (Person, name)
δ (dateOfBirth) ⊑ cityOfBirth
cityOfBirth ⊑ δ (dateOfBirth)
(funct dateOfBirth)
name
dateOfBirth
Person
1,1
cityOfBirth
0,n
City
PERSON ⊑ ∃ (cityOfBirth)
(funct cityOfBirth)
∃ cityOfBirth ⊑ PERSON
∃ cityOfBirth- ⊑ CITY

ABox: un esempio

Alfabeto delle costanti per gli oggetti: $\Gamma_O = \{pers1, pers2, city1, city2...\}$

Alfabeto delle costanti per i valori: $\Gamma_V = \{Valerio, Ilaria, 02/02/1983, 06/08/1983, ...\}$

$$Person(\mathbf{pers1}) \quad (1)$$
$$Person(\mathbf{pers2}) \quad (2)$$
$$name(\mathbf{pers1}, Valerio) \quad (3)$$
$$name(\mathbf{pers2}, Ilaria) \quad (4)$$
$$cityOfBirth(\mathbf{pers1}, \mathbf{city2}) \quad (5)$$
$$cityOfBirth(\mathbf{pers2}, \mathbf{city1}) \quad (6)$$
$$City(\mathbf{city1}) \quad (7)$$
$$City(\mathbf{city2}) \quad (8)$$
$$dateOfBirth(\mathbf{stud1}, \mathbf{city2}, 06/08/1983) \quad (9)$$
$$dateOfBirth(\mathbf{stud2}, \mathbf{city1}, 02/02/1983) \quad (10)$$

Traduzione TBox ⇝ FOL

Simboli di predicato:

Atomic Concept: per ogni Atomic Concept $C \in A$ nell'alfabeto allora esiste un simbolo di predicato unario $C/1$.

Atomic Concept Attribute: per ogni Atomic Concept Attribute $a \in U_C$ nell'alfabeto allora esiste un simbolo di predicato binario $a/2$.

Atomic Role: per ogni Atomic Role $R \in P$ nell'alfabeto allora esiste un simbolo di predicato binario $r/2$.

Atomic Role Attribute: per ogni Atomic Role Attribute $b \in U_R$ nell'alfabeto allora esiste un simbolo di predicato ternario $b/3$.

Value Domain: per ogni Value Domain T_i esiste un simbolo di predicato unario con il nome del corrispondente data type $T_i/1$.

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd : string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd : string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Attributo di concept:

$$\forall XY\ name(X, Y) \rightarrow Person(X)$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd : string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Tipo di attributo di concept:

$$\forall XY\ name(X, Y) \rightarrow String(Y)$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd : string$$
$$\boxed{Person \sqsubseteq \delta(name)}$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Molteplicità minima 1 attributo di concept:

$$\boxed{\forall X\ Person(X) \rightarrow \exists Y\ name(X, Y)}$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd:string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd:date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Molteplicità massima 1 attributo di concept:

$$\forall XYZ\ name(X,Y) \wedge name(X,Z) \rightarrow Y = Z$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd:string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$

$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$

$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd:date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Role tra due concepts:

$$\forall XY\ cityOfBirth(X,Y) \rightarrow Person(X) \wedge City(Y)$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd:string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$\boxed{Person \sqsubseteq \exists cityOfBirth}$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd:date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Cardinalità minima 1 del role:

$$\boxed{\forall X\ Person(X) \rightarrow \exists Y\ cityOfBirth(X,Y)}$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd:string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd:date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Cardinalità massima 1 del role:

$$\forall XYZ\ Person(X) \wedge cityOfBirth(X,Y) \wedge cityOfBirth(X,Z) \rightarrow Y = Z$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd : string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Attributo di role e tipo:

$$\forall XYZ\ dateOfBirth(X, Y, Z) \rightarrow cityOfBirth(X, Y) \wedge Date(Z)$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd:string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd:date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Molteplicità minima 1 attributo di role:

$$\forall XY\ cityOfBirth(X,Y) \rightarrow \exists Z\ dateOfBirth(X,Y,Z)$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd:string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd:date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Molteplicità massima 1 attributo di role:

$$\forall XYZW\ cityOfBirth(X,Y) \wedge dateOfBirth(X,Y,Z) \wedge \wedge dateOfBirth(X,Y,W) \rightarrow Z = W$$

Traduzione TBox ➠ FOL

$$\delta(name) \sqsubseteq Person$$
$$\rho(name) \sqsubseteq xsd : string$$
$$Person \sqsubseteq \delta(name)$$
$$(funct\ name)$$
$$\exists cityOfBirth \sqsubseteq Person$$
$$\exists cityOfBirth^{-} \sqsubseteq City$$
$$Person \sqsubseteq \exists cityOfBirth$$
$$(funct\ cityOfBirth)$$
$$\delta(dateOfBirth) \sqsubseteq cityOfBirth$$
$$\rho(dateOfBirth) \sqsubseteq xsd : date$$
$$cityOfBirth \sqsubseteq \delta(dateOfBirth)$$
$$(funct\ dateOfBirth)$$
$$id(Person, name)$$

Identifying property:

$$\forall XYZ\ Person(X) \wedge Person(Y) \wedge name(X, Z) \wedge name(Y, Z) \rightarrow X = Y$$

Software OntologyConverter

E' uno strumento software che traduce una TBox DL-LiteA in una teoria della FOL.

Software OntologyConverter

E' uno strumento software che traduce una TBox DL-LiteA in una teoria della FOL.

INPUT: file XML che descrive una TBox rispettando la DTD di QuOnto

Software OntologyConverter

E' uno strumento software che traduce una TBox DL-LiteA in una teoria della FOL.

INPUT: file XML che descrive una TBox rispettando la DTD di QuOnto

OUTPUT: file di testo in formato OTTER/mace contenete la traduzione in FOL della TBox

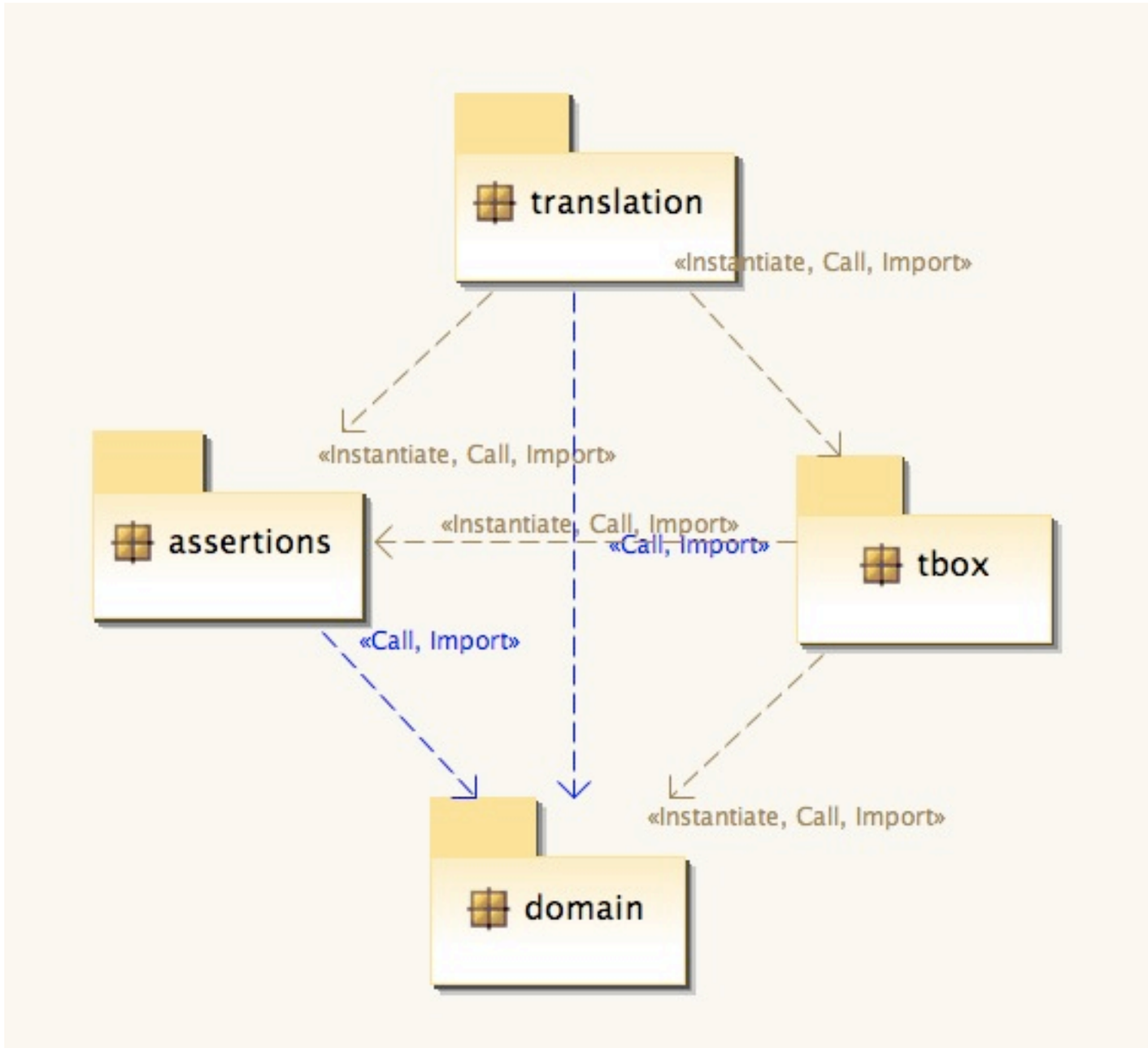
OntologyConverter: input XML

Frammento di DTD:

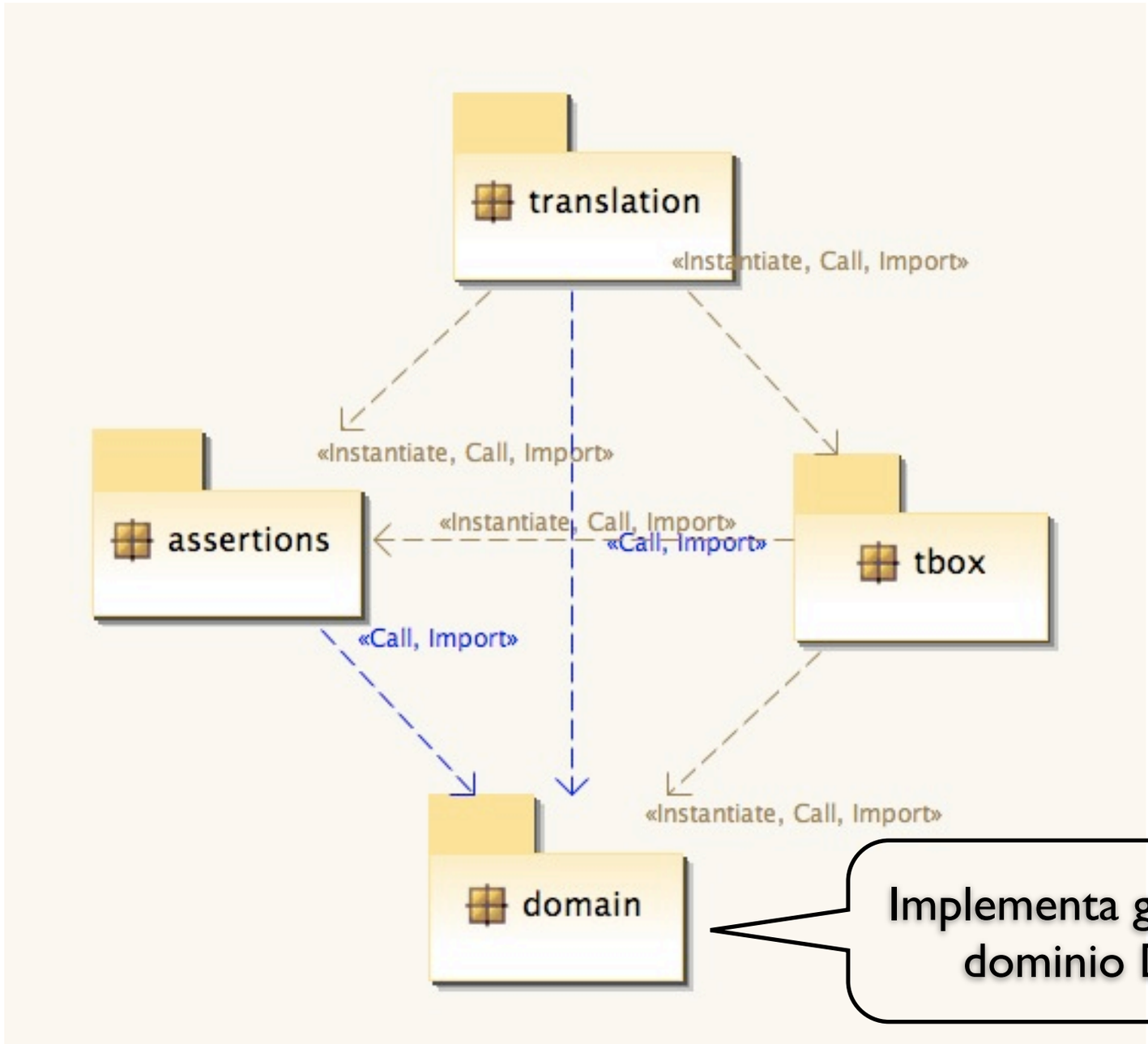
```
<!ELEMENT ontology (alphabet, tbox)>

<!ELEMENT alphabet (atomicC | atomicV | atomicR |
        atomicCA | atomicRA)+>
<!ELEMENT tbox (inclusionAssertion | funct)*>

<!ELEMENT atomicC (#PCDATA)>
<!ELEMENT atomicV (#PCDATA)>
<!ELEMENT atomicR (#PCDATA)>
<!ELEMENT atomicCA (#PCDATA)>
<!ELEMENT atomicRA (#PCDATA)>
<!ELEMENT inclusionAssertion ((basicC, generalC+) |
        (basicV, generalV+) | (basicR, generalR+) |
        (atomicCA, generalCA+) | (atomicRA, generalRA+))>
<!ELEMENT funct (basicR | atomicCA | atomicRA)>
```

```
<!-- PERSON ⊑ δ (name) -->
        <inclusionAssertion>
            <basicC>
                <atomicC>Person</atomicC>
            </basicC>
            <generalC>
                <signedC sign="positive">
                    <basicC>
                        <CADomain>
                            <atomicCA>name</atomicCA>
                        </CADomain>
                    </basicC>
                </signedC>
            </generalC>
        </inclusionAssertion>
<!-- δ (name) ⊑ PERSON -->
        <inclusionAssertion>
            <basicC>
                <CADomain>
                    <atomicCA>name</atomicCA>
                </CADomain>
            </basicC>
            <generalC>
                <signedC sign="positive">
                    <basicC>
                        <atomicC>Person</atomicC>
                    </basicC>
                </signedC>
            </generalC>
        </inclusionAssertion>
```

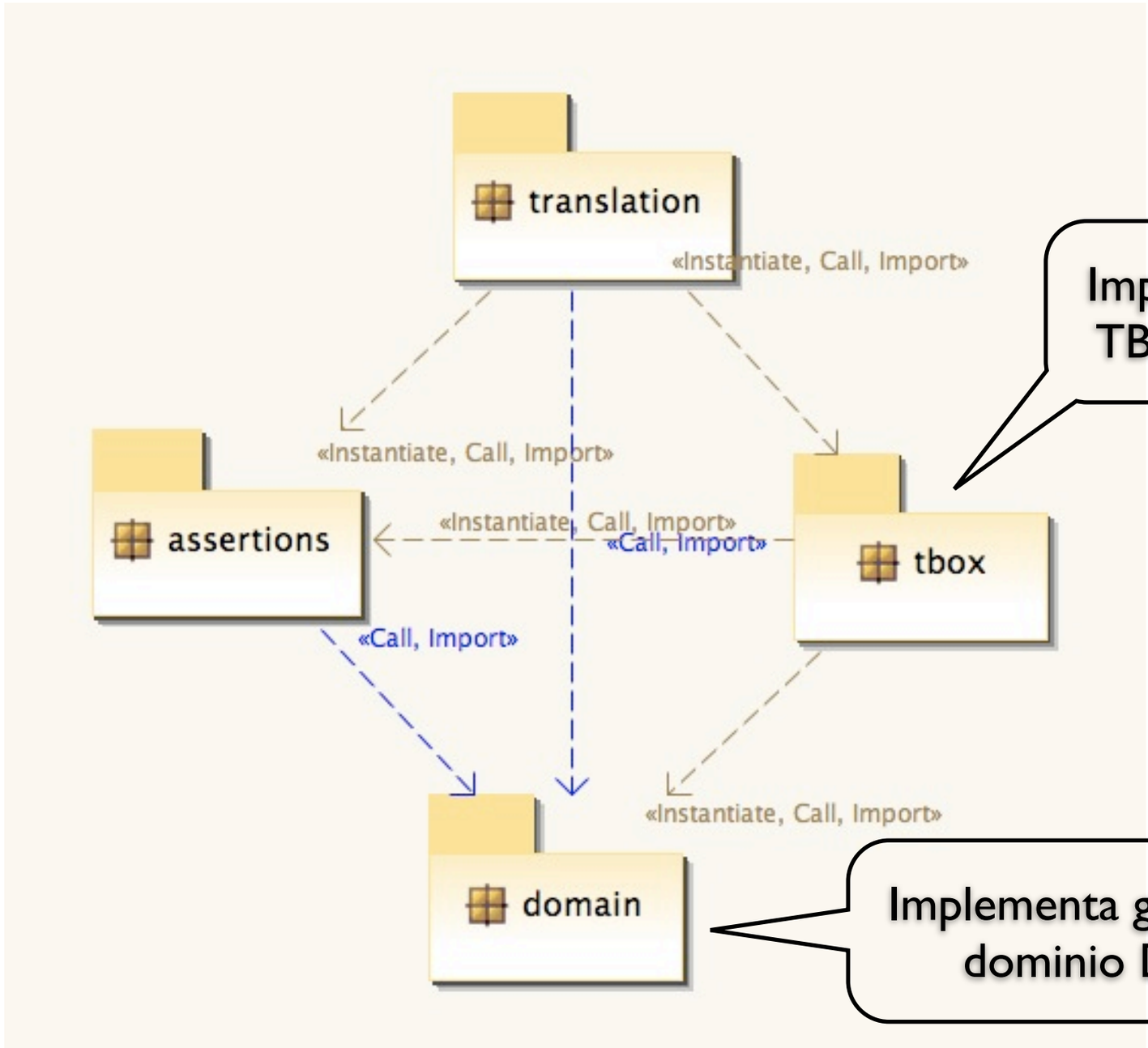
OntologyConverter: package diagram



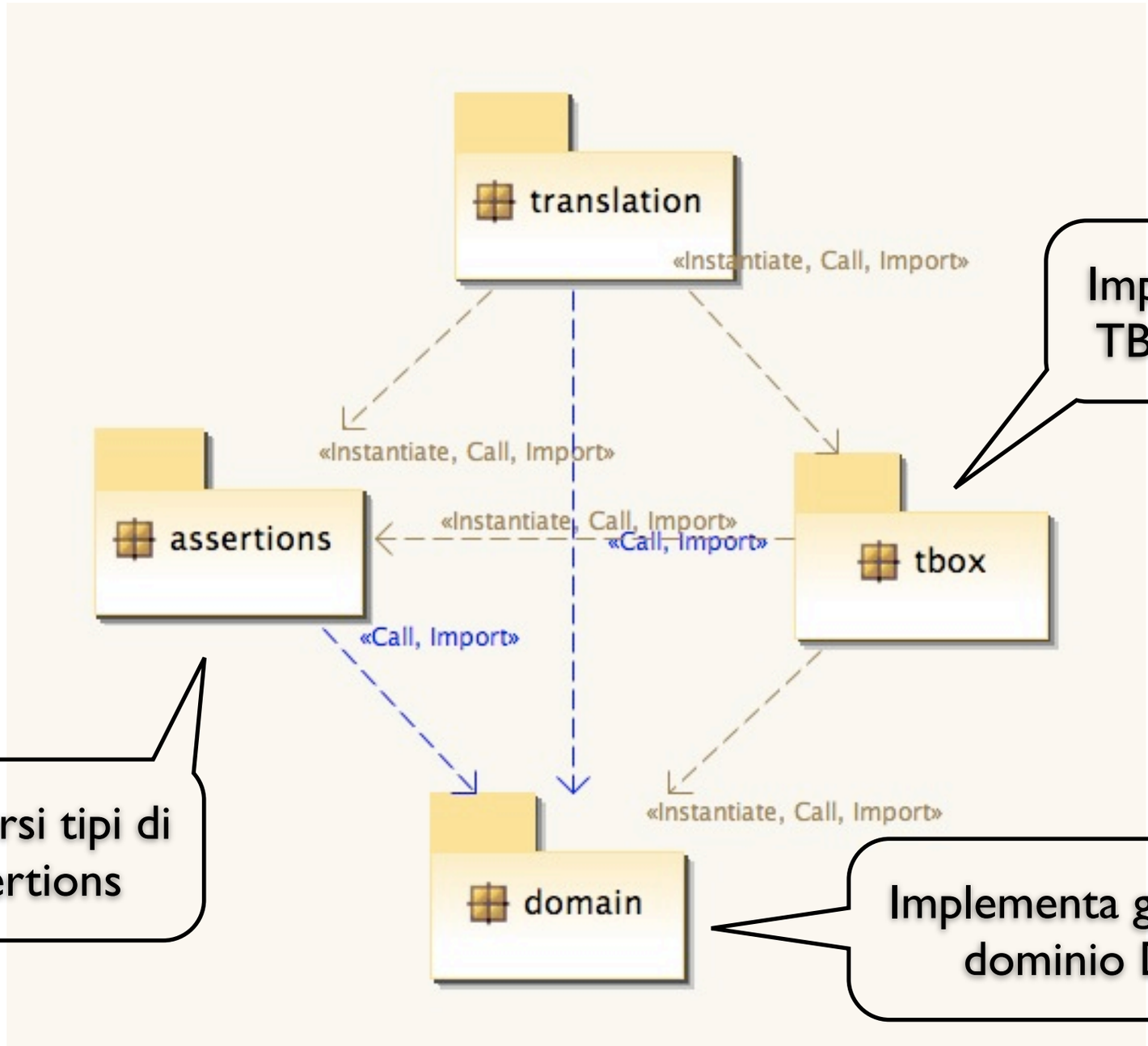
OntologyConverter: package diagram



OntologyConverter: package diagram



OntologyConverter: package diagram



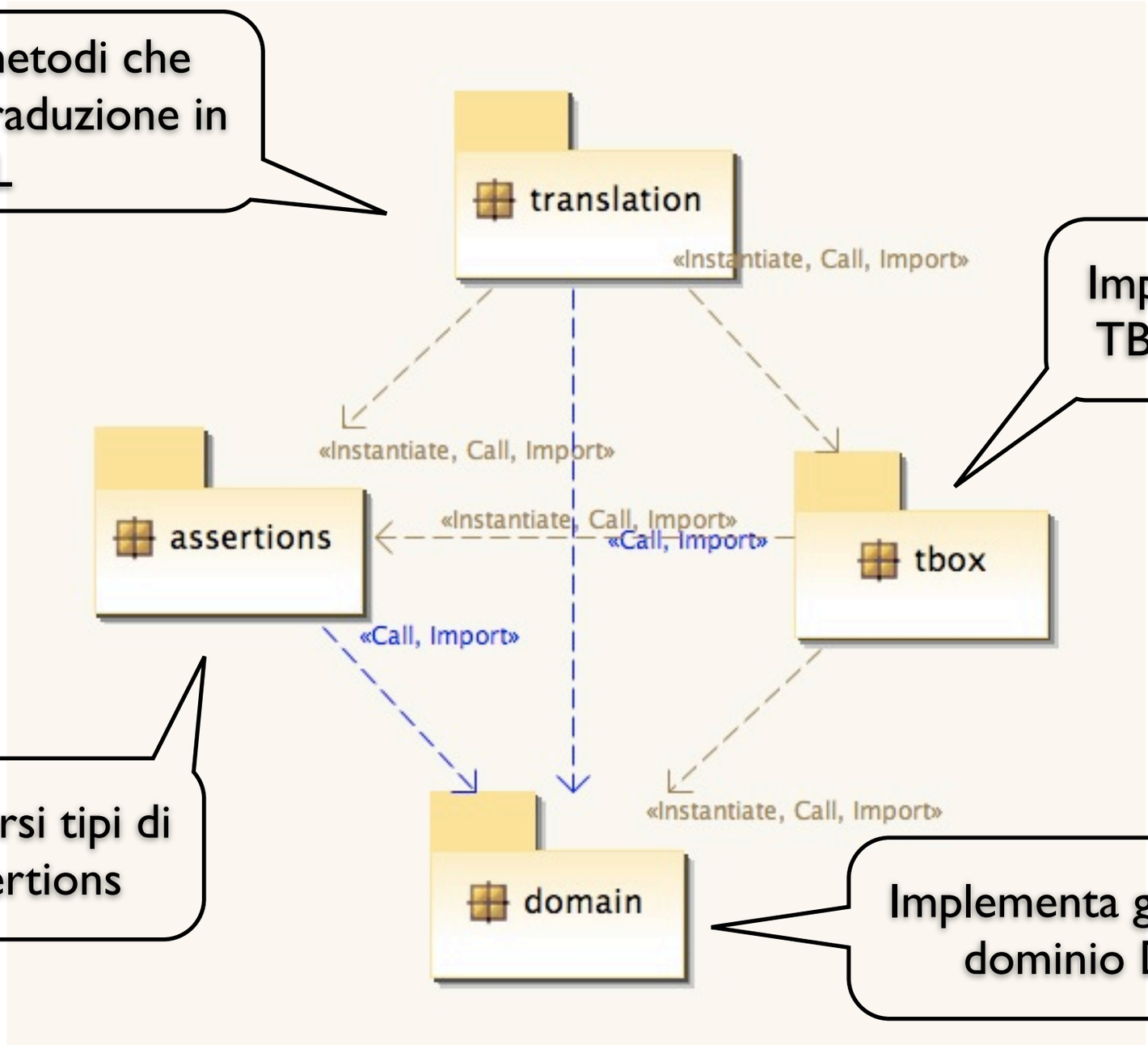
Implementa il concetto di TBox e parsing dell'XML

Implementa i diversi tipi di intensional assertions

Implementa gli oggetti del dominio DL-LiteA

OntologyConverter: package diagram

Contiene i metodi che effettuano la traduzione in FOL

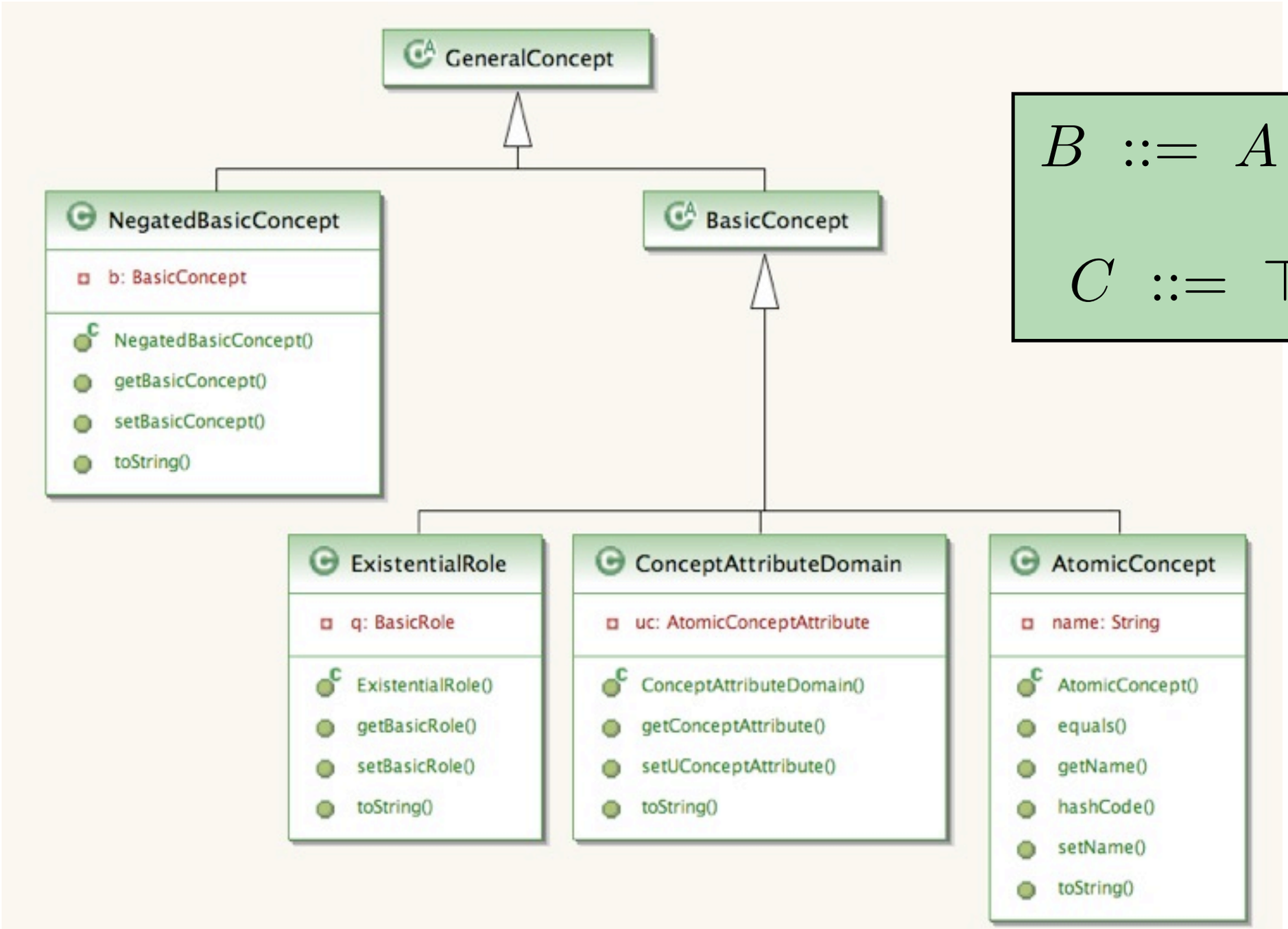


Implementa il concetto di TBox e parsing dell'XML

Implementa i diversi tipi di intensional assertions

Implementa gli oggetti del dominio DL-LiteA

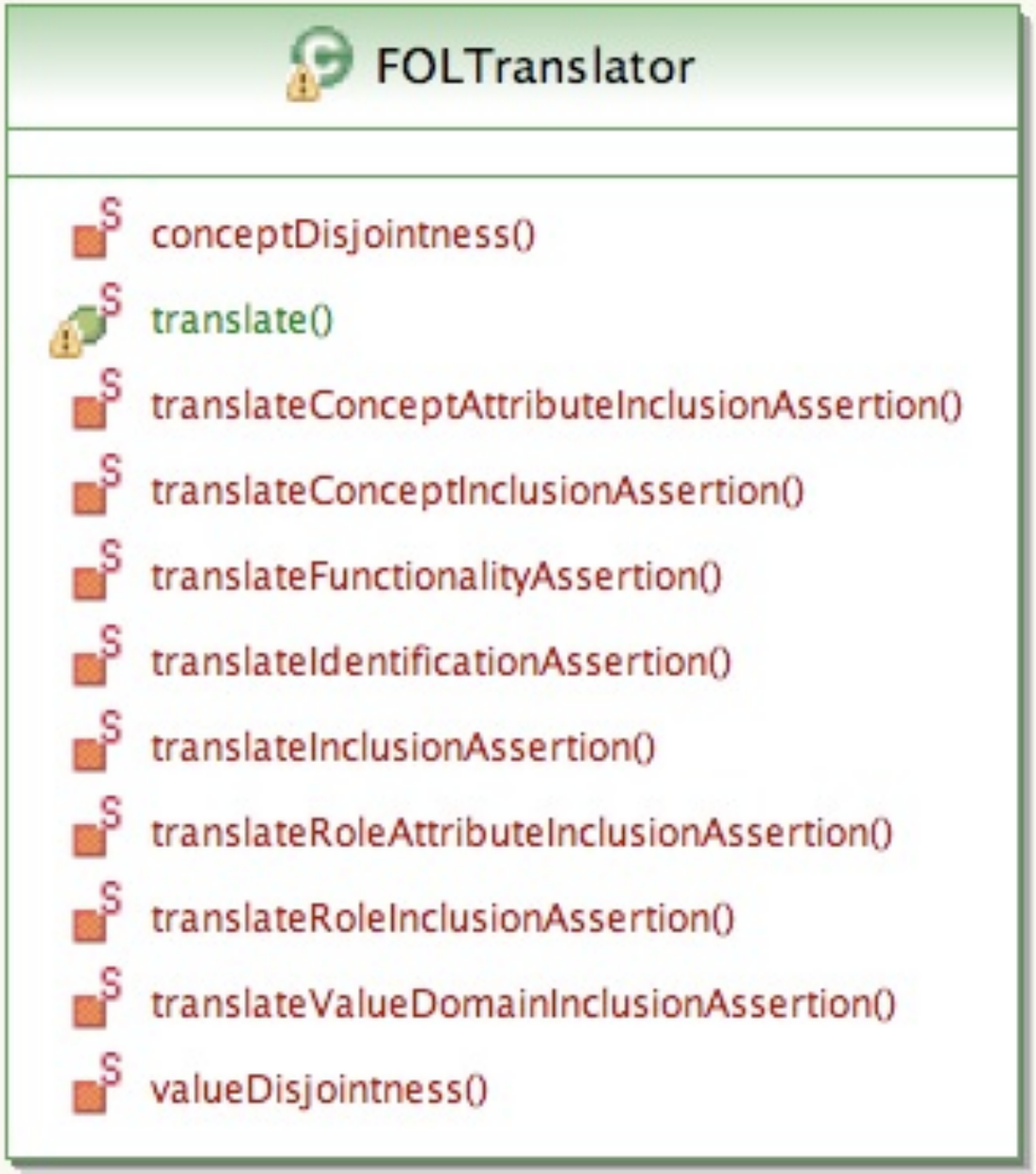
OntologyConverter: domain



$$B \ ::= \ A \mid \exists Q \mid \delta(U_C)$$

$$C \ ::= \ \top_C \mid B \mid \neg B$$

OntologyConverter: translation




```
public class FOLTranslator {
    public static String translate(TBox t) {
        ...
    }
```

Prende in input un oggetto TBox e per ogni asserzione chiama la specifica funzione di traduzione.

OntologyConverter: translation

```
private static String translateConceptInclusionAssertion(TBox t, InclusionAssertion ass) {
    Included left = ass.getLeft();
    Including right = ass.getRight();
    if ((left instanceof ConceptAttributeDomain)&(right instanceof AtomicConcept)) {
        ConceptAttributeDomain deltaUc = (ConceptAttributeDomain)left;
        AtomicConcept a = (AtomicConcept)right;
        return "all XY ("+deltaUc.getConceptAttribute().getName() +"(X,Y) -> "+a.getName()+"(X)).";
    }
    else if ((left instanceof AtomicConcept)&(right instanceof ConceptAttributeDomain)) {
        ConceptAttributeDomain deltaUc = (ConceptAttributeDomain)right;
        AtomicConcept a = (AtomicConcept)left;
        return "all X ("+a.getName() +"(X) -> (exists Y "+deltaUc.getConceptAttribute().getName()+"(X,Y))).";
    }
    else if ((left instanceof AtomicConcept)&(right instanceof AtomicConcept)) {
        AtomicConcept sx = (AtomicConcept)left;
        AtomicConcept dx = (AtomicConcept)right;
        t.addISA(new ISA(sx,dx));
        return "all X ("+sx.getName() +"(X) -> "+dx.getName()+"(X)).";
    }
    else if ((left instanceof AtomicConcept)&(right instanceof NegatedBasicConcept)) {
        AtomicConcept sx = (AtomicConcept)left;
        NegatedBasicConcept dx = (NegatedBasicConcept)right;
        BasicConcept b = dx.getBasicConcept();
        if (b instanceof AtomicConcept) {
            AtomicConcept a = (AtomicConcept)b;
            return "all X ("+sx.getName() +"(X) -> -"+a.getName()+"(X)).";
        }
```

δ(Uc)⊆ A

A⊆δ(Uc)

A⊆A’

A⊆¬A’

OntologyConverter: translation

```
private static String translateConceptInclusionAssertion(TBox t, InclusionAssertion ass) {
    ...
    else if ((left instanceof ExistentialRole)&(right instanceof AtomicConcept)) {
        ExistentialRole esisteQ = (ExistentialRole)left;
        AtomicConcept a = (AtomicConcept)right;
        BasicRole q = esisteQ.getBasicRole();
        if (q instanceof DirectRole) {
            DirectRole p = (DirectRole)q;
            return "all XY ("+p.getRole().getName() +"(X,Y) -> "+a.getName()+"(X)).";
        }
        else if (q instanceof InverseRole) {
            InverseRole pmeno = (InverseRole)q;
            return "all XY ("+pmeno.getRole().getName() +"(X,Y) -> "+a.getName()+"(Y)).";
        }
    }
    else if ((left instanceof AtomicConcept)&(right instanceof ExistentialRole)) {
        AtomicConcept a = (AtomicConcept)left;
        ExistentialRole esisteQ = (ExistentialRole)right;
        BasicRole q = esisteQ.getBasicRole();
        if (q instanceof DirectRole) {
            DirectRole p = (DirectRole)q;
            return "all XY ("+a.getName() +"(X) -> (exists Y "+p.getRole().getName()+"(X,Y))).";
        }
        else if (q instanceof InverseRole) {
            InverseRole pmeno = (InverseRole)q;
            return "all XY ("+a.getName() +"(X) -> (exists Y "+pmeno.getRole().getName()+"(Y,X))).";
        }
```

$\exists Q \sqsubseteq A$

$A \sqsubseteq \exists Q$

OntologyConverter: ESEMPIO D'USO

IMDB
The Internet Movie Database

http://www.imdb.com

OntologyConverter: IMDB

Dimensioni TBox:

11 Atomic Concept
56 Atomic Concept Attributes
57 Atomic Roles
38 Atomic Role Attributes

470 Intensional Assertions!

OntologyConverter: IMDB

OntologyConverter: IMDB

- Parsing XML e creazione oggetto TBox

OntologyConverter: IMDB

- Parsing XML e creazione oggetto TBox

- Traduzione Intensional Assertions da DL-LiteA in FOL

OntologyConverter: IMDB

- Parsing XML e creazione oggetto TBox

- Traduzione Intensional Assertions da DL-LiteA in FOL

- P: proprietà da verificare

OntologyConverter: IMDB

- Parsing XML e creazione oggetto TBox

- Traduzione Intensional Assertions da DL-LiteA in FOL

- P: proprietà da verificare

- Se P logicamente implicata ➟ OTTER

OntologyConverter: IMDB

- Parsing XML e creazione oggetto TBox

- Traduzione Intensional Assertions da DL-LiteA in FOL

- P: proprietà da verificare

- Se P logicamente implicata ➟ OTTER

- altrimenti ➟ MACE

OntologyConverter: IMDB

Vediamo l'esempio in pratica...