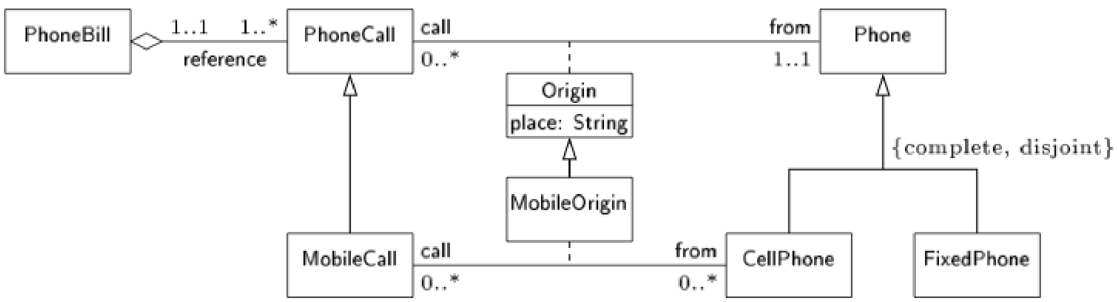
Knowledge Representation and Reasoning

Solutions to Modelling and Reasoning with UML Class diagrams

1 Converting from UML to First-Order Logic

Consider the following UML class diagram about different kinds of phones, and phone bills they belong to.



The diagram shows that a MobileCall is a particular kind of PhoneCall and that the Origin of each PhoneCall is one and only one Phone. Additionally, a Phone can be only of two different kinds: a Fixed Phone or a Cell Phone. Mobile calls originate (through the association MobileOrigin) from cell phones. The association MobileOrigin is contained in the binary association Origin: hence MobileOrigin inherits the attribute place of association class Origin. Finally, a PhoneCall is referenced in one and only one PhoneBill, whereas a PhoneBill contains at least one PhoneCall.

1. Convert the UML diagram into Description Logics.

Answer:

$$\begin{aligned}
\exists place &\sqsubseteq Origin \\
\exists place^- &\sqsubseteq String \\
Origin &\sqsubseteq \exists place \sqcap (\leq 1 place) \\
\exists reference &\sqsubseteq PhoneBill \\
\exists reference^- &\sqsubseteq PhoneCall \\
PhoneBill &\sqsubseteq (\geq 1 reference) \\
PhoneCall &\sqsubseteq (\geq 1 reference^-) \sqcap \\
&\quad (\leq 1 reference^-) \\
\exists callO &\sqsubseteq Origin \\
\exists callO^- &\sqsubseteq PhoneCall \\
\exists fromO &\sqsubseteq Origin \\
\exists fromO^- &\sqsubseteq Phone \\
Origin &\sqsubseteq \exists callO \sqcap (\leq 1 callO) \sqcap \\
&\quad \exists fromO \sqcap (\leq 1 fromO)
\end{aligned}$$

$$\begin{aligned}
PhoneCall &\sqsubseteq (\geq 1 callO^-) \sqcap (\leq 1 callO^-) \\
\exists callMO &\sqsubseteq MobileOrigin \\
\exists callMO^- &\sqsubseteq MobileCall \\
\exists fromMO &\sqsubseteq MobileOrigin \\
\exists fromMO^- &\sqsubseteq CellPhone \\
MobileOrigin &\sqsubseteq \exists callMO \sqcap (\leq 1 callMO) \sqcap \\
&\quad \exists fromMO \sqcap (\leq 1 fromMO) \\
MobileOrigin &\sqsubseteq Origin \\
callMO &\sqsubseteq callO \\
fromMO &\sqsubseteq fromO \\
MobileCall &\sqsubseteq PhoneCall \\
CellPhone &\sqsubseteq Phone \sqcap \neg FixedPhone \\
FixedPhone &\sqsubseteq Phone \\
Phone &\sqsubseteq CellPhone \sqcup FixedPhone
\end{aligned}$$

2. Suppose you add a generalization to the diagram asserting that each CellPhone is a FixedPhone. Which classes become inconsistent (i.e. they cannot be populated) and which pairs of classes become equivalent?

> **Answer:**
> First, the class CellPhone is inconsistent, i.e., it has no instances. Indeed, the disjointness constraint asserts that there are no cell phones that are also fixed phones, and since the empty set is the only set that can be at the same time disjoint from and contained in the class FixedPhone, the class CellPhone must have it as extension. Second, since the class Phone is made up by the union of classes CellPhone and FixedPhone, and since CellPhone is inconsistent, the classes Phone and FixedPhone are equivalent, hence one of them is redundant. Finally, since there are no cell phones, there are no pairs in the association MobileOrigin, and so it is inconsistent too. The class MobileCall is not inconsistent since it can be populated by instances that do not participate to association MobileOrigin.