Semantics of UML class diagrams

Till Mossakowski[1]

Otto-von-Guericke Universität Magdeburg, Germany

April 26, 2016

Sets

Definition (Set)

A set is a collection of objects. The basic relation is membership:

$$x \in A \ (x \text{ is a member of } A)$$

The following operations and relations are defined on sets:

- **empty set** $\varnothing$ is the set with no members
- **enumeration set** $\{a_1; \ldots; a_n\}$ contains exactly $a_1; \ldots; a_n$
- **subset** $A \subseteq B$ iff for all x: $x \in A$ implies $x \in B$
- **comprehension** $\{x \in A \mid P(x)\}$ (the set of all $x \in A$ such that $P(x)$ holds)
- **union** $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
- **intersection** $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$
- **difference** $A \setminus B = \{x \mid x \in A \text{ and not } x \in B\}$
- **powerset** $\mathcal{P}(A) = \{B \mid B \subseteq A\}$
- **set of words (strings) over A** $A^* = \{\varepsilon\} \cup \{a_1 \ldots a_n \mid a_i \in A\}$

Relations

Definition (Cartesian product)

$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$
$A \times B \times C = \{(a, b, c) \mid a \in A, b \in B \text{ and } c \in C\}$
etc.

Definition (Relation)

A binary relation R on A and B is given by a set of pairs

$$R \subseteq A \times B$$

$(a, b) \in R$ often is written as $a\ R\ b$.
If $A = B$, then we speak of a binary relation R on A.

Orders

Definition (Partial Order)

A **partial order** $(A, \leq)$ is given by a set A and a binary relation $\leq$ on A, such that

- for all $x \in A$: $x \leq x$ (reflexivity)
- for all $x, y, z \in A$: $x \leq y$ and $y \leq z$ imply $x \leq z$ (transitivity)
- for all $x, y \in A$: $x \leq y$ and $y \leq x$ imply $x = y$ (antisymmetry)

Definition (Total Order)

A partial order $(A, \leq)$ is called a **total order**, if additionally

- for all $x, y \in A$: $x \leq y$ or $y \leq x$ or $x = y$ (trichotomy)

Orders

Example (Sample partial orders)

- the set of natural numbers with the usual ordering $\leq$
- the set of natural numbers with the ordering "x can be divided by y"
- the lexicographic order on strings (used for sorting)
- the prefix order on strings

Which of these are total?

Semantics of UML class diagrams

- The semantics is given by a mathematical function $\mathcal{S}$, representing a snapshot of a system
- A snapshot $\mathcal{S}$ includes all objects of a system, and their relations
- The evolution of a system can be represented by the transition of a system from a snapshot $\mathcal{S}_1$ to a new snapshot $\mathcal{S}_2$
    - evolution of a system is only considered later (state machines)

Semantics of a class hierarchy

- A class hierarchy is given a by a partial order $(C, \leq)$
  - antisymmetry means that cyclic subclasses are forbidden
- Each class $c \in C$ is interpreted as a finite set $\mathcal{S}(c)$
  - $\mathcal{S}(c)$ is the set of objects that are instances of class c
- If $c \leq d$, then $\mathcal{S}(c) \subseteq \mathcal{S}(d)$ must hold
  - hence, "each c **is a** d"

Semantics of a generalisation set

- disjoint$(c_1 \leq d, \ldots, c_n \leq d)$ expresses the condition

$$\mathcal{S}(c_i) \cap \mathcal{S}(c_j) = \varnothing \text{ for } i \neq j$$

- complete$(c_1 \leq d, \ldots, c_n \leq d)$ expresses the condition

$$\mathcal{S}(c_1) \cup \cdots \cup \mathcal{S}(c_n) = \mathcal{S}(d)$$

- no condition for overlapping and incomplete

Semantics of enumeration types and built-in types

- An enumeration type T with literals $l_1, \dots, l_n$ has as its semantics the set of literals:

$$\mathcal{S}(T) = \{l_1, \dots, l_n\}$$

- A built-in type has a predefined semantics, e.g.

$$\mathcal{S}(\text{integer}) = \mathbb{Z}$$

$$\mathcal{S}(\text{string}) = A^*$$

where A is a suitable set of characters

Functions

Definition (Function)

A function f from a set A to a set B, written $f : A \nrightarrow B$, associates with some of the elements $a \in A$ a unique element $b \in B$. This association is symbolically expressed as $f(a) = b$. The elemen $a \in A$ is called the argument and b the value of the function application $f(a)$. If there is no b, then $f(a)$ is undefined.

Note: a function $f : A \nrightarrow B$ can be represented as a binary relation $graph(f) \subseteq A \times B$ as follows:

$$graph(f) = \{(a, f(a)) \mid a \in A\}$$

We have right-uniqueness:

$$(x, y) \in graph(f) \text{ and } (x, z) \in graph(f) \text{ imply } y = z$$

Definition (Total function)

A function $f : A \nrightarrow B$ is **total**, if $f(a)$ is defined for all $a \in A$. In this case, we write $f : A \rightarrow B$.

Semantics of attributes

The semantics of an attribute $a : T$ of a class c is given by a partial function

$$\mathcal{S}(a) : \mathcal{S}(c) \nrightarrow \mathcal{S}(T)$$

The function is needed to be partial because the value of the attribute may not have been initialised yet.

If an attribute $a : T$ has multiplicity other than 1, the semantics is given by

$$\mathcal{S}(a) : \mathcal{S}(c) \nrightarrow \mathcal{P}(\mathcal{S}(T))$$

Semantics of query operations

The semantics of a query operation $op(x_1 : T_1; \ldots x_n : T_n) : T$ of a class c is given by a partial function

$$\mathcal{S}(op) : \mathcal{S}(c) \times \mathcal{S}(T_1) \times \cdots \times \mathcal{S}(T_n) \nrightarrow \mathcal{S}(T)$$

Non-query operations will lead to a new snapshot and are not considered here.

Semantics of associations

- Given two classes c and d, the semantics of an association $c \overset{a}{\rule{2em}{0.4pt}} d$ is given by a binary relation

$$\mathcal{S}(a) \subseteq \mathcal{S}(c) \times \mathcal{S}(d)$$

- An association $c \overset{a}{\rule{2em}{0.4pt}} d$ satisfies the multiplicity

$$c \underset{m..n}{\overset{a}{\rule{4em}{0.4pt}}} d$$

  if for all $y \in \mathcal{S}(d)$,

$$m \leq |\ \{x \mid (x,y) \in \mathcal{S}(a)\}\ | \leq n$$

Here, $|\ X\ |$ is the number of elements in X (also called cardinality of X).

Consistency

- A UML class diagram is **consistent**, if there is at least one snapshot satisfying all its conditions. Otherwise, it is **inconsistent**.
- A UML class diagram is **strongly consistent**, if there is at least one snapshot **intepreting all classes as non-empty sets** satisfying all its conditions.

Semantics of association ends

- The semantics of an association end

$$c \; \overset{\mathsf{e} \qquad a}{\rule{4cm}{0.4pt}} \; d$$

is given by a function $\mathcal{S}(e) : \mathcal{S}(d) \to \mathcal{P}(\mathcal{S}(c))$:

$$\text{for } y \in \mathcal{S}(d),\ \mathcal{S}(e)(y) = \{x \mid (x, y) \in \mathcal{S}(a)\}$$

- The semantics of an association end with multiplicity 0..1

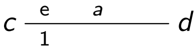
$$c \; \underset{0..1}{\overset{\mathsf{e} \qquad a}{\rule{4cm}{0.4pt}}} \; d$$

is given by a partial function $\mathcal{S}(e) : \mathcal{S}(d) \nrightarrow \mathcal{S}(c)$:

$$\text{for } y \in \mathcal{S}(d),\ \mathcal{S}(e)(y) = \text{ the } x \text{ with } (x, y) \in \mathcal{S}(a) \text{ (if existing)}$$

Semantics of association ends

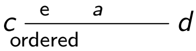
- The semantics of an association end with multiplicity 1



is given by a total function $\mathcal{S}(e) : \mathcal{S}(d) \to \mathcal{S}(c)$:

$$\text{for } y \in \mathcal{S}(d),\ \mathcal{S}(e)(y) = \text{ the } x \text{ with } (x, y) \in \mathcal{S}(a) \text{ (always exists)}$$

- The semantics of an association end



is given by a function $\mathcal{S}(e) : \mathcal{S}(d) \to (\mathcal{S}(c))^*$ with:

$$\text{for } y \in \mathcal{S}(d),\ \ x \in \mathcal{S}(e)(y) \text{ iff } (x, y) \in \mathcal{S}(a)$$

Semantics of aggregations and compositions

- Aggregations and compositions are associations and inherit their semantics
- Aggregations and compositions represent **part-whole** relationships; hence they are **irreflexive** (and so are their transitive closures)
- For compositions, there is a condition on evolution of snapshots: if an object of the composite class is deleted, so must be all associated elements of component classes

Definition (transitive closure)

Given a binary relation $R \subseteq A \times A$, its **transitive closure** is the least relation $R^* \subseteq A \times A$ with

- $R \subseteq R^*$
- R^* is transitive

Semantics of object diagrams

- An object diagram specifies some objects to be part of the snapshot $\mathcal{S}$
- If an object o is specified to have class c, then it must hold that
$$\mathcal{S}(o) \in \mathcal{S}(c)$$
- If there is an association a between objects o and p, then it must hold that
$$(\mathcal{S}(o), \mathcal{S}(p)) \in \mathcal{S}(a)$$

Different stages of “goodness” of UML class diagrams

1. Syntactically ill-formed
2. Syntactically well-formed, but static-semantically ill-formed
3. Syntactically and static-semantically well-formed, but inconsistent
4. Syntactically and static-semantically well-formed and consistent, but methodologically doubtful
5. Syntactically and static-semantically well-formed, consistent and methodologically clean