expr = \( \cdot \rightarrow \tau \rightarrow /BV \) 0 \( \leq \)

and

or, with a little renaming:

\( n \in \{ \cdot \} \) evaluates

\( \tau (\cdot \rightarrow /llbracket I \rrbracket = n) \)

\( \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)

\( n \tau \times \cdot \rightarrow /llbracket I \rrbracket = n \)
OCL Inconsistency Example

We write \( \varphi \in I^\sigma \) if and only if there exists a system state \( \sigma \) which satisfies all of them, i.e. if \( \varphi \) is called over constraints on \( I \) we write \( \varphi \in I^\sigma \) where \( \varphi \) is unrealizable (or inconsistent and \( \Sigma \in DS_n \)).

Context: \( \text{name} : \text{String}, \text{age} : \text{Integer} \)

\( \text{Meeting} \) - 05 – 2014

- Location: String
- Description: Integer
- Title: String
- Number of Participants: Integer
- Start Date: Date
- Duration: Time
• Whether a set of OCL constraints is satisfiable or not is in general not as obvious as in the made-up example.

• Wanted: A procedure which decides the OCL satisfiability problem.

• Unfortunately: in general undecidable. Otherwise we could, for instance, solve diophantine equations:

\[ c_1 x_1^1 + \cdots + c_m x_m^m = d. \]

Encoding in OCL:

```
allInstances C->exists (w: C | c_1 * w.x^1 + \cdots + c_m * w.x^m = d).
```

• And now?

Options:

• Constrain OCL, use a less rich fragment of OCL.

• Revert to finite domains — basic types vs. number of objects.

---

**Expressive Power**

"Pure OCL expressions only compute primitive recursive functions, but not recursive functions in general."

[Cengarle and Knapp, 2001]

**Evolution over Time**

"finally self.x > 0"

Proposals for fixes e.g. [Flake and Müller, 2003]. (Or: sequence diagrams.)

**Real-Time**

"Objects respond within 10s"

Proposals for fixes e.g. [Cengarle and Knapp, 2002].

**Reachability**

"After insert operation, node shall be reachable."

Fix: add transitive closure.

---

**Concrete Syntax**

"The syntax of OCL has been criticized — e.g., by the authors of Catalysis [...] — for being hard to read and write.

• OCL's expressions are stacked in the style of Smalltalk, which makes it hard to see the scope of quantified variables.

• Navigations are applied to atoms and not sets of atoms, although there is a collect operation that maps a function over a set.

• Attributes, [...], are partial functions in OCL, and result in expressions with undefined value."

[Jackson, 2002]
A graph \( G \) is represented graphically as follows:

\[ (\Sigma, \tau, \sigma) = (\{\Sigma_1, \Sigma_2\}, \tau, \sigma) = \emptyset \]

where \( \Sigma \) is the set of vertexes, \( \tau \) is the set of edges, and \( \sigma \) is the set of hyperedges. A graph is a directed graph (digraph) if \( \tau \neq \emptyset \), and an undirected graph if \( \tau = \emptyset \).

Example: Graph

\[ (\{x, y, z\}, \{\{x, y\}, \{y, z\}\}, \{\{x, y, z\}\}) \]

The edge set \( \tau \) consists of the pairs \( \{x, y\} \) and \( \{y, z\} \), and the hyperedge set \( \sigma \) consists of the set \( \{x, y, z\} \).

Graphs can be used to represent various relationships, such as

- Social networks
- Road networks
- Electrical circuits
- Chemical structures
- Computer networks
In other words: then we can uniquely reconstruct the corresponding system state.

Claim:

Each object diagram

•

\[ \mathcal{G} \]

\[ \mathcal{C} \]

\[ \mathcal{N} \]

partial

\[ \mathcal{U} \]

\[ \mathcal{T} \]

\[ \mathcal{F} \]

Complete vs. Partial Object Diagram

Objects Diagrams: More Examples?
Discussion

We slightly deviate from the standard (for reasons):

• In the course, $C_0$, $C_1$ and $C^*$-typed attributes only have sets as values. UML also considers multisets, that is, they can have $u_1:C_1, u_2:C_n$. (This is not an object diagram in the sense of our definition because of the requirement on the edges $E$). Extension is straightforward but tedious.

• We allow to give the valuation of $C_0, C_1$- or $C^*$-typed attributes in the values compartment.

• Allows us to indicate that a certain $r$ is not referring to another object.

• Allows us to represent "dangling references", i.e. references to objects which are not alive in the current system state.

• We introduce a graphical representation of $\emptyset$ values.

References


