A Deduction Calculus for Concurrent Objects

Didier Buchs

Objectives of the Deduction Calculus

- Propose a concurrent object-oriented formalism
  - data structures
  - operational behavior
  - true concurrency
- Introduce object-oriented specification
- Provide a formal semantics
- Provide a deduction rules and in the future proof techniques

Features of the Object Model

1. Data structures
2. Notion of object
3. Interaction between the objects
4. Concurrency

Data Structures: Syntax

- S is a finite set of sorts.
- a *S-sorted set* A is a family of sets indexed by S, we write $A = (A_s)_{s \in S}$
- $F = (F_{w,s})_{w \in SORTS^*, s \in SORTS}$ is a $(SORTS^* \times SORTS)$-sorted set of function names of $FUNC$.
- An *Adt signature* is a couple $\Sigma = \langle S, F \rangle$
- The set of term is a S set T_Σ and its extension $T_{\Sigma,X}$ on the variable S-set X.
- The axioms are simply equations $t = t'$ such that $t, t' \in T_{\Sigma,X}$.

Data Structures: Deduction relation

- Given a specification $Spec$ with the set of axioms AX
- Deductible equalities $t = t'$ with $t, t' \in T_{\Sigma,X}$
- Deduction relation $AX \vdash t = t'$

Data Structures: Deduction rules

A-Axiom $$\frac{}{AX \cup t = t' \vdash t = t'}$$

A-Reflexivity $$\frac{}{AX \vdash t = t}$$

A-Symmetry $$\frac{AX \vdash t_1 = t_2}{AX \vdash t_2 = t_1}$$

A-Transitivity $$\frac{AX \vdash t_1 = t_2,\ AX \vdash t_2 = t_3}{AX \vdash t_1 = t_3}$$

A-Substitutivity $$\frac{AX \vdash t_1 = t_1' \ ,AX \vdash t_2 = t_2' \ , \ ... \ ,AX \vdash t_n = t_n'}{AX \vdash f(t_1, t_2, t_3, ..., t_n) = f(t_1', t_2', t_3', ..., t_n')}$$

A-conjunction $$\frac{AX \vdash \psi \ ,AX \vdash t = t'}{AX \vdash \psi \cup t = t'}$$

Data Structures: Example of deduction

Object Model: Interfaces

Each object o of a system of objects possesses an *interface* $I_o =< o, M >$ where M is a $\{o\} \times S^*$-sorted set of method names.

Given O a set of object o_i, $0 \leq i \leq n$, and their respective interfaces $I_{o_i} =< o_i, M_i >$ the *global interface* I_O is the union of all I_{o_i} interfaces $o_i, 0 \leq i \leq n$, $I_O =< O, \cup_{0 \leq i \leq n} M_i >$.

Object Model: State Space

Given a signature $\Sigma = (S, F)$ and A a S-sorted set of attributes names, the set of states $States_\Sigma(A)$ is built as follows:

- $\epsilon \in States_\Sigma(A)$
- $\forall s \in S, \forall p \in A_s, \forall t \in (T_\Sigma(X))_s, (p, t) \in States_\Sigma(A)$
- $\forall m, m' \in States_S(A), m + m' \in States_\Sigma(A)$

Object Model: State Space example

A state is in our model a multiset of values for each attribute.

- The couple (p, t) indicates that the attribute p has an additional value t.
- Multiset are used to express the concurrent access of the attributes, i.e. the available ressources.
- A multiset $\{a, a, b\}$ in the place p is noted $(p, a) + (p, a) + (p, b)$.

Object Model: Events

Let $\Sigma = \langle S, F \rangle$ be a complete signature. Given a set of parameter values A, a $(O \times S^*)$-sorted set of methods M, a set of object identifiers O and a S-sorted set of variables X. The set of events $EVENTS_{A,M,O,X}$ is recursively built with the set of synchronization $SYNC_{A,M,O,X}$.

The set of events $EVENTS_{A,M,O,X}$ is the least set such that:

- $\forall o \in O, \forall m \in M_{o,s_1,\ldots,s_n}, o.m(a_1, \ldots, a_n) \in EVENTS_{A,M,O,X}$
- $\forall o \in O, \forall m \in M_{o,s_1,\ldots,s_n}, \forall sync \in SYNC_{A,M,O,X}, o.m(a_1, \ldots, a_n)$ **with** $sync \in EVENTS_{A,M,O,X}$
- $\forall o \in O, \forall m \in M_{o,s_1,\ldots,s_n}, o.m(a_1, \ldots, a_n) \in SYNC_{A,M,O,X}$
- $\forall sync, sync' \in SYNC_{A,M,O,X}, sync \mathbin{//} sync' \in SYNC_{A,M,O,X}$
- $\forall sync, sync' \in SYNC_{A,M,O,X}, sync \mathinner{..} sync' \in SYNC_{A,M,O,X}$

Object Model : example of Events

The observable event "$o.m(a_1, a_2)$ **`with`** $o_1.m_1(a_1) \mathbin{//} o_2.m_2(a_2)$" represents the simultaneous synchronization of the method m of an object o with both the methods m_1 and m_2 of two objects o_1 and o_2.

Object Model : Behavioral Axioms

Let $\Sigma = \langle S, F \rangle$ be a complete signature. Given a $(O \times S^*)$-sorted set of methods M, a S-sorted set of place P, and a S-sorted set of variables X and a set of object O. A *behavioral axiom* is an expression

$$Cond \Rightarrow Event : Pre \rightarrow Post$$

defined as follows:

- $Event \in EVENTS_{(T_{\Sigma,X}),M,O,X}$;
- $Cond$ is an expression $e_1 \wedge \cdots \wedge e_n$ where all e_i $(1 \leq i \leq n)$ are equations over Σ and X;
- $Pre, Post \in States_{\Sigma}(P)$

The set of all behavioral axioms is noted $Beh(\Sigma, M, O, P, X)$.

Object Model : Example of Behavioral Axioms

Object Model : Specification

Given O a set of object, M a set of methods, an object o, $o \in O$ and its respective interfaces $I_o =< o, M_o >$ and $\Sigma = \langle S, F \rangle$ be a complete signature, X a set of variable and P a set of places, the specification of one object o is defined as $Spec_{o,\Sigma} =< I_o, P, BAX_o >$ where $BAX_o \in Beh(\Sigma, M, O, P, X)$.

Deduction on Object Model : Transition systems

Let *Spec* be a CO-OPN specification with a set of places P. A *transition system* is defined as

$$TS_{Spec} = \langle States_{\Sigma}(P) \times EVENTS_{Spec,M(Spec)} \times States_{\Sigma}(P) \rangle$$

We note a transition e between two states m and m' by $m \xrightarrow{e} m'$.

Deduction on Object Model: Deduction relation

- Given a specification $Spec$ with the set of axioms AX and $SPEC_O$ with a set of axioms BAX.
- Deductible transitions $m_1 \stackrel{e}{\rightarrow} m_2$
- Deduction relation $BAX, AX \vdash\ m_1 \stackrel{e}{\rightarrow} m_2$

Deduction rules for states

Multiset-emptyl $$\frac{BAX, AX \vdash\ m_1 + \epsilon \stackrel{e}{\rightarrow} m_2}{BAX, AX \vdash\ m_1 \stackrel{e}{\rightarrow} m_2}$$

Multiset-emptyr $$\frac{BAX, AX \vdash\ m_1 \stackrel{e}{\rightarrow} m_2 + \epsilon}{BAX, AX \vdash\ m_1 \stackrel{e}{\rightarrow} m_2}$$

Multiset-comml $$\frac{BAX, AX \vdash\ m_1 + m_2 \stackrel{e}{\rightarrow} m_3}{BAX, AX \vdash\ m_2 + m_1 \stackrel{e}{\rightarrow} m_3}$$

Multiset-commr $$\frac{BAX, AX \vdash\ m_1 \stackrel{e}{\rightarrow} m_2 + m_3}{BAX, AX \vdash\ m_1 \stackrel{e}{\rightarrow} m_3 + m_2}$$

Multiset-assl + opp $$\frac{BAX, AX \vdash\ (m_1 + m_2) + m_3 \stackrel{e}{\rightarrow} m_4}{BAX, AX \vdash\ m_1 + (m_2 + m_3) \stackrel{e}{\rightarrow} m_4}$$

Multiset-assr + opp $\dfrac{BAX, AX \vdash \; m_1 \xrightarrow{e} (m_2 + m_3) + m_4}{BAX, AX \vdash \; m_1 \xrightarrow{e} m_2 + (m_3 + m_4)}$

Rules for Objects (Local Behavior)

Monotonicity $\dfrac{BAX, AX \vdash \; m_1 \xrightarrow{e} m_2}{BAX, AX \vdash \; m_1 + m \xrightarrow{e} m_2 + m}$

Evaluation $\dfrac{\phi \implies a \; With \; sync : Pre \rightarrow Post \in BAX, \; AX \vdash \sigma(\phi)}{BAX, AX \vdash \; \sigma(Pre) \xrightarrow{\sigma(a) \; With \; \sigma(sync)} \sigma(Post)}$

Substitution $\dfrac{BAX, AX \vdash \; m_1 \xrightarrow{e} m_2}{BAX, AX \vdash \; \sigma(m_1) \xrightarrow{\sigma(e)} \sigma(m_2)}$

Rules for Objects (Substitution)

Event-Replacement
$$\frac{AX \vdash a = a',\ BAX, AX \vdash\ \ m_1 \xrightarrow{a\ With\ sync} m_2}{BAX, AX \vdash\ \ m_1 \xrightarrow{a'\ With\ sync} m_2}$$

Synchro-Replacement
$$\frac{AX \vdash sync = sync',\ BAX, AX \vdash\ \ m_1 \xrightarrow{a\ With\ sync} m_2}{BAX, AX \vdash\ \ m_1 \xrightarrow{a\ With\ sync'} m_2}$$

Marking-Replacementl
$$\frac{AX \vdash t = t',\ BAX, AX \vdash\ \ m_1 + (p, t) \xrightarrow{e} m_2}{BAX, AX \vdash\ \ m_1 + (p, t') \xrightarrow{e} m_2}$$

Marking-Replacementr
$$\frac{AX \vdash t = t',\ BAX, AX \vdash\ \ m_1 \xrightarrow{e} m_2 + (p, t)}{BAX, AX \vdash\ \ m_1 \xrightarrow{e} m_2 + (p, t')}$$

Rules for object specification

Sequential-Synchro
$$\frac{BAX, AX \vdash\ \ m_1 \xrightarrow{e} m_2,\ BAX, AX \vdash\ \ m_2 \xrightarrow{e'} m_3}{BAX, AX \vdash\ \ m_1 \xrightarrow{e..e'} m_3}$$

Simultaneous-Synchro
$$\frac{BAX, AX \vdash\ \ m_1 \xrightarrow{e} m_2,\ BAX, AX \vdash\ \ m_3 \xrightarrow{e'} m_4}{BAX, AX \vdash\ \ m_1 + m_3 \xrightarrow{e\&e'} m_2 + m_4}$$

Resolve-Synchro
$$\frac{BAX_o, AX \vdash\ \ m_1 \xrightarrow{a\ With\ sync} m_2,\ BAX_{\{o_1,\ldots,o_n\}}, AX \vdash\ \ m_3 \xrightarrow{sync} m_4, o \geq \{o_1, ..., o_n\}}{BAX_o \cup BAX_{o_1,\ldots,o_n}, AX \vdash\ \ m_1 + m_3 \xrightarrow{a} m_2 + m_4}$$

Extension: Modeling Exceptions and Failures

- The transitions will be: $m_1 \xrightarrow{e}_f m_2$, where e is the event set and f is the failure set $f \in \{\perp, ex_1, ex_2, ...\}$
- Add implicit axioms for failure detection, $\perp$ is the predicate failure indicator (we would have full definition of method call, i.e $\forall m_1$, $\forall e$, $\exists f, m_2$, such that $BAX, AX \vdash \; m_1 \xrightarrow{e}_f m_2$).
- Add axioms for failure recovery, on $\perp$ or exception $\{ex_1, ex_2, ...\}$ indicators.
- Determine negation on terms $(t \neq t')$ ($\Rightarrow$ Problems with models).
- Revise the deduction rules on transitions by adding negative behavior

Comments about the Object Model

➤ **Use of an Object-Based Model instead of Object-Orientation**

- evaluation of the use of object orientation mechanisms such as inheritance and subtyping
- Subtyping for specification enrichment

➤ **Concurrency**

- for distribution
- inside objects