FOUNDATIONS OF THE UML

Lecture 10 :

Statecharts Semantics (2)

Joost-Pieter Katoen

MOVES

moves.rwth-aachen.de / i2 / 192

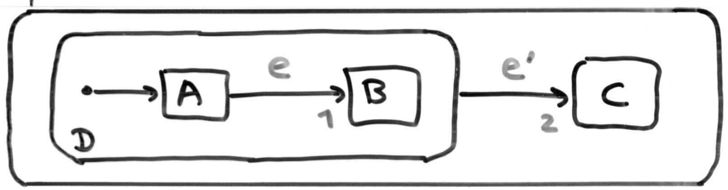
WS 2007/2008

January 11, 2008

PRIORITIES

To rule out certain nondeterminism between edges, priorities are used.

Example



in configuration $\{ root, D, A \}$ and $\{e, e'\} \subseteq I$ alive there is a nondeterministic choice between taking edge 1, edge 2, or both.

Assumption: there is a partial-order relation, $\lhd$

$\lhd \subseteq Edges \times Edges$ defined by:

irreflexive, anti-symmetric, transitive

$\underbrace{ed \lhd ed'}_{\text{ed' has a higher priority than ed}}$ if $scope(ed) \geqslant scope(ed')$

ancestor relation in node hierarchy

Example above:

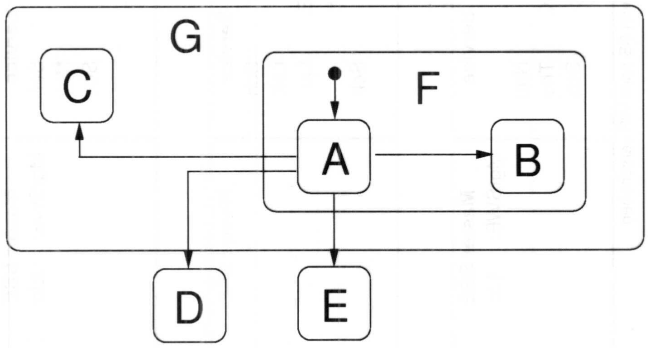
$2 \lhd 1$ since $\underbrace{scope(2)}_{root} \geqslant \underbrace{scope(1)}_{D}$

Priority

Priority is a partial order on edges; different priority schemes exist, e.g.

$$\textit{prio}(ed) \rhd \textit{prio}(ed') \quad \text{if} \quad \textit{scope}(ed) \lneq \textit{scope}(ed') \qquad \text{UML}$$

$$\textit{prio}(ed) \lhd \textit{prio}(ed') \quad \text{if} \quad \textit{scope}(ed) \lneq \textit{scope}(ed') \qquad \text{STATEMATE}$$



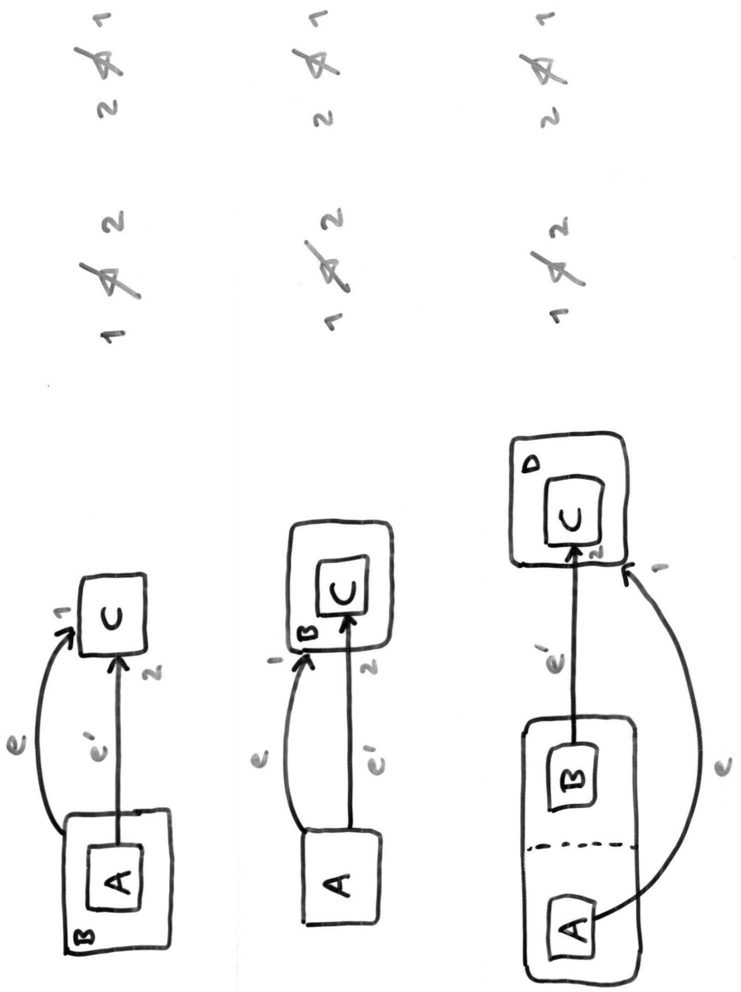
$\textit{prio}(A \rightarrow C) \rhd \textit{prio}(A \rightarrow D)$ since $\textit{scope}(A \rightarrow C) = G \lneq \textit{root} = \textit{scope}(A \rightarrow D)$

4

30

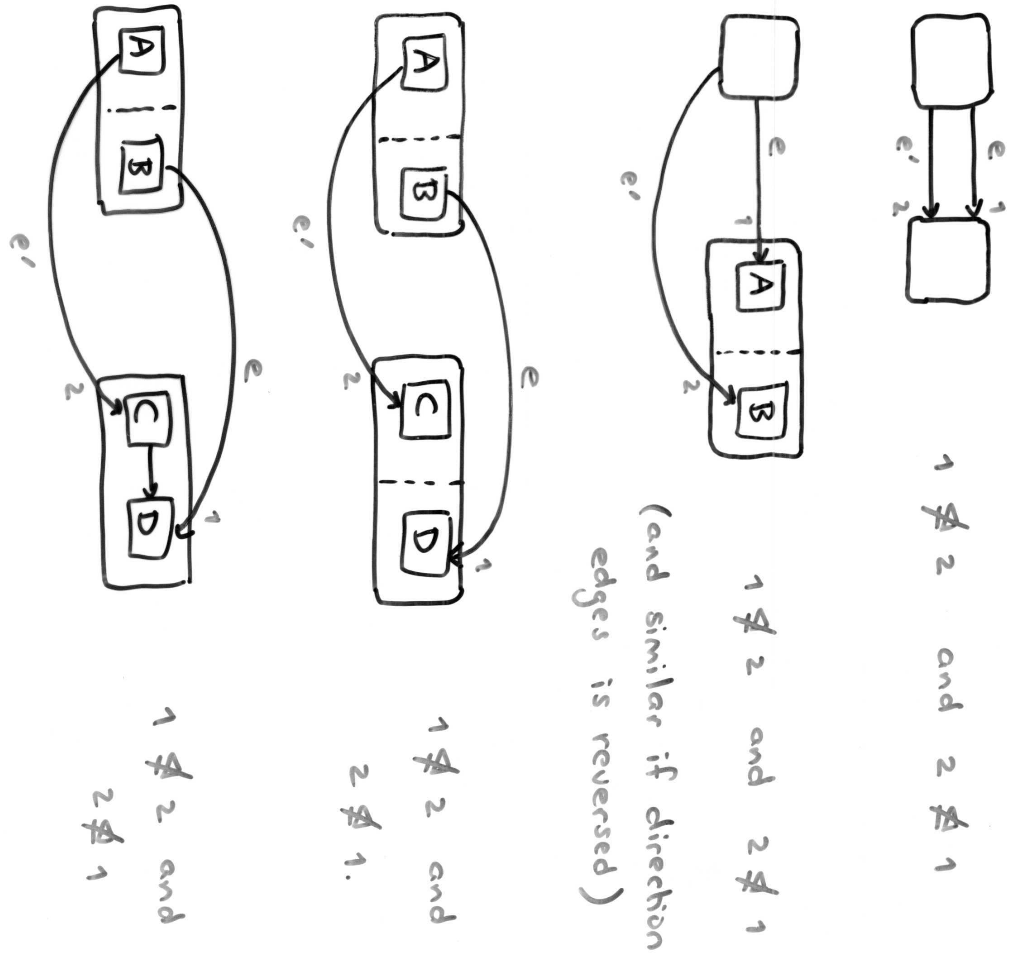
3

PRIORITY EXAMPLES



Note: different priority schemes do exist;
this one is adopted in UML statecharts
(we believe);
here hierarchy (= scope) is used to
define priorities, but other principles
could be exploited as well.

NONDETERMINISM



So: priorities rule out some nondeterminism but not all!

What is now a step?

A step is a *set of enabled edges* such that:

- all edges in a step are enabled
- all edges in a step are pairwise consistent
  - they are identical or
  - scopes are (descendants of) different children of the same AND-node
- enabled edge ed is not in step implies /* respect priorities */

  $\exists ed' \in$ step. $(ed$ inconsistent with ed' and $\neg(\underbrace{prio(ed') < prio(ed)}_{ed' \lhd ed}))$
- a step must be maximal (wrt. set inclusion)

Computing the set T of steps

Initialize $T := \varnothing$

While

there is an enabled edge in *En*$(C, I, V) - T$
which is consistent with all edges in T

Do

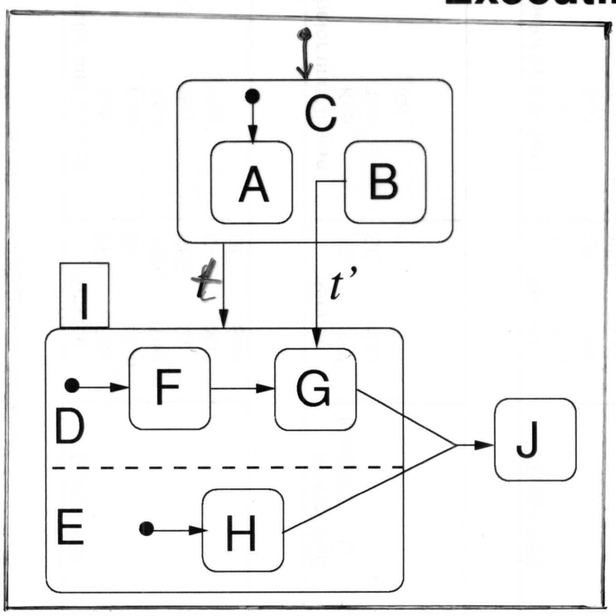
choose one of these edges which has maximal priority
and add this edge to T

Return T

Steps(*En*(C, I, V)) denotes the set of steps in state (C, I, V)

7

Executing a step



$\{$ *root*, $A, C \} \xrightarrow{\{t\}} \{$ *root*, $I, D, E, F, H \}$

i.e., the *default completion of* $\{$ *root*, $I \}$

Notation: *Exec*$(C_{1..n}, T_{1..n}) = ((C'_1, I'_1, V'_1), \ldots, (C'_n, I'_n, V'_n))$

8

STEP EXECUTION

- For a single statechart, executing a step results in performing the actions of the edges in the step + changing "control" to the target nodes thereof

- Actions in SC_j may influence the sets of events of other SCs, e.g. SC_i $(i \neq j)$

  e.g. if action is send i.e

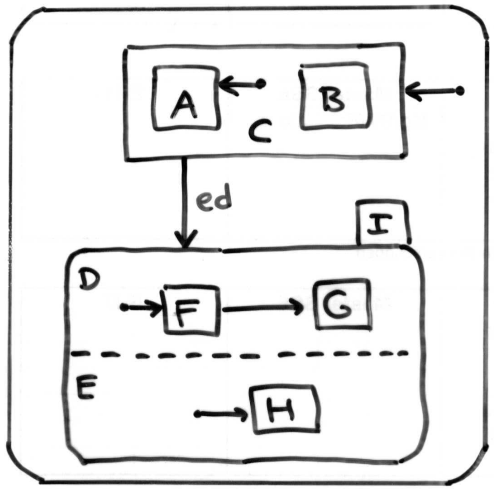
- Therefore: execution of steps is considered on the complete collection of statecharts

STEP EXECUTION

- The default completion C' of some set C of nodes is the canonical superset of C such that C' is a configuration.

- If C' contains an OR-node x and $\text{children}(x) \cap C = \emptyset \rightarrow \text{default}(x) \in C'$

Example:



default completion of $C = \{\text{root}, I\}$ is $C' = \{\text{root}, I\} \cup \{D, E, F, H\}$

default completion of $C = \{\text{root}, C\}$ is $\{\text{root}, C\} \cup \{A\}$

STEP EXECUTION

- let C_j be current configuration of SC_j
- let T_j be the (a) step for SC_j
  - T_j is thus a <u>set</u> of edges in SC_j
- The new state (C_j', I_j', V_j') of SC_j is given by:
  - C_j' is the default completion of

$$\underbrace{\bigcup_{X \xrightarrow{e[g]/A} Y \in T_j} Y}_{\text{all target nodes of edges in } T_j} \quad \cup \quad \underbrace{\left\{ x \in C_j \;\middle|\; \begin{array}{l} \forall X \longrightarrow Y \in T_j . \\ \quad x \notin \text{scope}(X,Y) \end{array} \right\}}_{\text{all unaffected nodes}}$$

- $$I_j' = \underbrace{\bigcup_{k=1}^{n} \left\{ e \mid \exists X \xrightarrow{e[g]/A} Y \in T_k .\ \text{send}\ j.e \in A \right\}}_{\text{all events that have been sent in steps } T_1, \ldots, T_n \text{ to } SC_j}$$

- $$V_j'(v) = \begin{cases} V_j(v) & \text{if } \forall X \xrightarrow{/A} Y \in T_j .\ v := \ldots \notin A \\ \text{val}(expr) & \text{if } \exists X \xrightarrow{/A} Y \in T_j .\ \underline{\underline{v := expr}} \in A \end{cases}$$

MEALY MACHINE SEMANTICS

- Recall a mealy machine:

$$(\underbrace{Q}_{\text{states}}, \underbrace{q_0}_{\text{initial state } \in Q}, \underbrace{\Sigma}_{\text{input alphabet}}, \underbrace{\Gamma}_{\text{output alphabet}}, \underbrace{\delta}_{\text{input function}}, \underbrace{\omega}_{\text{output function}})$$

$$\delta : Q \times \Sigma \longrightarrow 2^Q$$

$$\omega : Q \times \Sigma \longrightarrow \Gamma$$

- Map a set of statecharts $\longrightarrow$ (nondeterm.) Mealy machine
  - States q = Tuples of states of $SC_1, \ldots, SC_n$
  - Input Σ = sets of events
  - Output Γ = sets of events
  - δ : defines the effect of step execution
  - ω : defines all events send to some $SC \notin \{SC_1, \ldots, SC_n\}$