FOUNDATIONS OF THE UML

Lecture #7: Realizing local choice MSGs

Joost-Pieter Katoen

MOVES @ RWTH

WS 2007/2008

December 10, 2007

REALIZABILITY

- Recall: mapping a set of MSCs — possibly given by an MSG — onto a distributed implementation (an MPA)
- Basic results: characterization of sets of MSCs (in terms of their traces) that are realizable or even safely realizable
- Checking realizability for a finite set of MSCs is coNP-complete, whereas safe realizability can be checked in PTIME
- A necessary condition for MSGs — that may specify an infinite set of MSCs — for realizability is communication-closedness
- In fact, regular MSC languages are realizable by deterministic MPA, but unfortunately, checking whether an MSG yields such language, is undecidable.

REALIZATION OF LOCAL CHOICE MSG

- Characterizations of (safe) realizability are non-constructive, i.e., they allow to decide whether a set of MSCs is realizable but not how.
- For a limited number of cases, algorithms can be provided for obtaining realizations.
- For example, we (= I2) investigate the usage of learning algorithms; see our tool Smyle
- This lecture: an algorithm for the safe realizability of local choice MSGs, i.e.,

local choice MSG ⇝ algorithm ⇝ deadlock free MPA

RECALL: LOCAL CHOICE

- e is a minimal event wrt. $<$ if

$$\neg\,(\exists e' \in E.\ e' < e)$$

- Process p is active in MSC M if $E_p \neq \emptyset$

p is active in path $\underbrace{v_1 v_2 \dots v_n}_{\pi}$ in MSG G

if $\exists v_i \in \pi.$ p is active in $\underbrace{\lambda(v_i)}$

$= $ MSC of vertex v_i

- MSG $G = (V, \rightarrow, v_0, F, \lambda)$ is local choice if

(1) $\exists$ active $p.$ $\forall \pi \in \text{Paths}(v_0).$

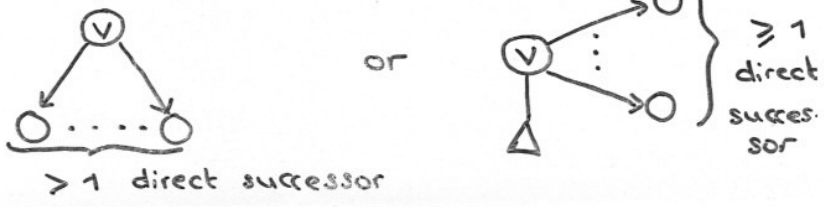
π has a single minimal event e with $e \in E_p$

(2) $\forall$ branching vertex $v \in V.$

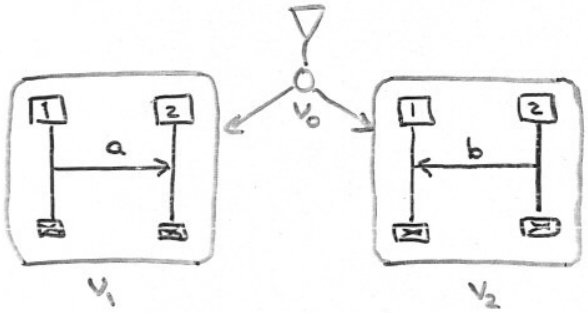
$\exists$ active $p.$ $\forall \pi \in \text{Paths}(w)$ with $v \rightarrow w.$

π has a single minimal event e with $e \in E_p$

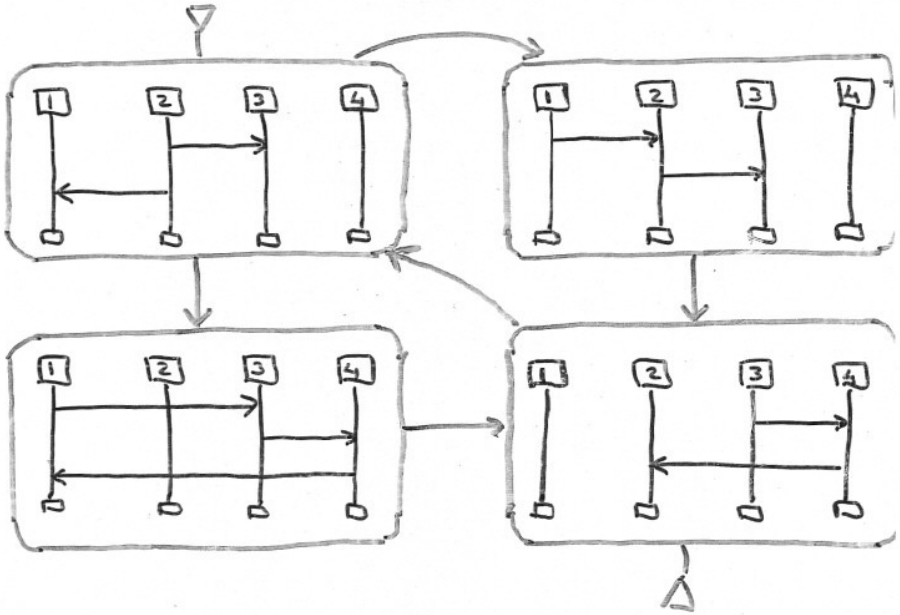
- $v \in V$ is a branching vertex if



LOCAL CHOICE MSGs



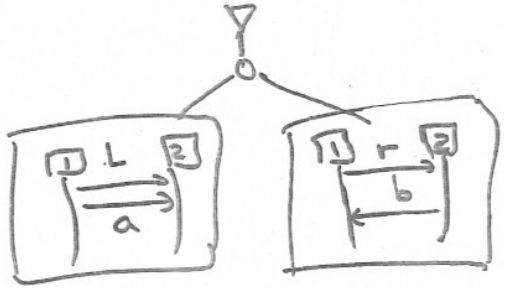
non local choice



local choice

LOCAL CHOICE MSGs

- Checking whether an MSG is local choice can be done in PTIME.
- A non-local choice MSG can be turned into a local choice MSG by adding synchronization messages.



REALIZING LOCAL CHOICE MSGs

MSG G which is local choice is safely realizable with additional data (which is of size linear in G)

Proof let $G = (V, \rightarrow, v_0, F, \lambda)$ be a local choice MSG. We construct the MPA

$$A_G = \left((A_p)_{p \in P}, \mathbb{D}, s_{in}, F' \right)$$

$$\downarrow$$

$$(S_p, \Delta_p)$$

- first consider the local automata A_p (i.e., define S_p and Δ_p)
- define $\mathbb{D}$
- global initial state s_{in}
- the set of global accepting states F'

REALIZING A LOCAL CHOICE MSG

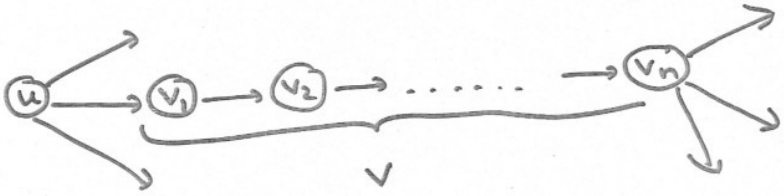
- local choice: at every branch $v \in G$ there is a unique process p, say, such that on every path from v the unique minimal event occurs at p
- Idea of constructing an MPA:
  - Process p determines the successor vertex of v;
  - Process p informs other processes about its decision by adding synchronization data to messages;
  - Synchronization data $\approx$ path from v to next branching vertex.

REALIZING A LOCAL CHOICE MSG

Let $nbp : V \to V^*$ such that $nbp(v)$ is a ~~A.t.~~ maximal non-branching path in MSG G to which v belongs

$$nbp(v) = \begin{cases} v & \text{if } v \in F \text{ or } v \text{ is a branching vertex} \\ v_1 \dots v_n & \text{otherwise } *) \end{cases}$$

*) where $v_i = v$ for some i, $0 < i \leq n$, $v_n \in F$ or is a branching vertex, and v_1 is a direct successor of a branching vertex and $v_2, \dots, v_{n-1} \notin F$ and are non-branching



LOCAL AUTOMATA

Local automaton A_p for process p is :

- $S_p = V \times E_p$, that is,

  state = (vertex v in G , all events that already occurred in MSC $\lambda(v)$ in E_p)

  More precisely :

  $s_p = (v, I)$ with $I \subseteq E_p$ downward-closed wrt. $<_p$ (process order in $\lambda(v)$)

  $e <_p e'$ and $e' \in I \rightarrow e \in I$

  in the MSC $\lambda(v)$

- Δ_p is defined by two inference rules :

(1) $$\frac{e \in E_p}{(v, I) \xrightarrow{\ell(e),\, m}_p (v, I \uplus \{e\})} \quad \text{with } m = nbp(v)$$

Note: since $I \uplus \{e\}$ is down closed wrt. $<_p$, event e is enabled at process p.

(2) $$\frac{e \in E_p^w \cap \lambda(w) \qquad v u_0 \dots u_n w \in Path(G) \quad \forall u_i .\, E_p = \emptyset}{(v, E_p^v) \xrightarrow{\ell(e),m}_p (w, \{e\})} \quad m = nbp(w),\ v \neq w$$

↑ .... all events in $\lambda(v)$ at p has occurred

REMAINING COMPONENTS

- $\mathbb{D} = \{ nbp(v) \mid v \in V \}$ set of maximal non-branching paths in MSC G

- s_{in}, global initial state $= \{ (v_0, \emptyset) \}^n$

  where # processes $= n$

- Global final states $F' \subseteq \prod_{p \in P} s_p$ such that $(s_{p_1}, \ldots, s_{p_n}) \in F'$ iff $\forall\, 0 < i \leq n$:

  $s_{p_i} = (v, O_p)$ with $O_p^v \subseteq E_p^v$ such that

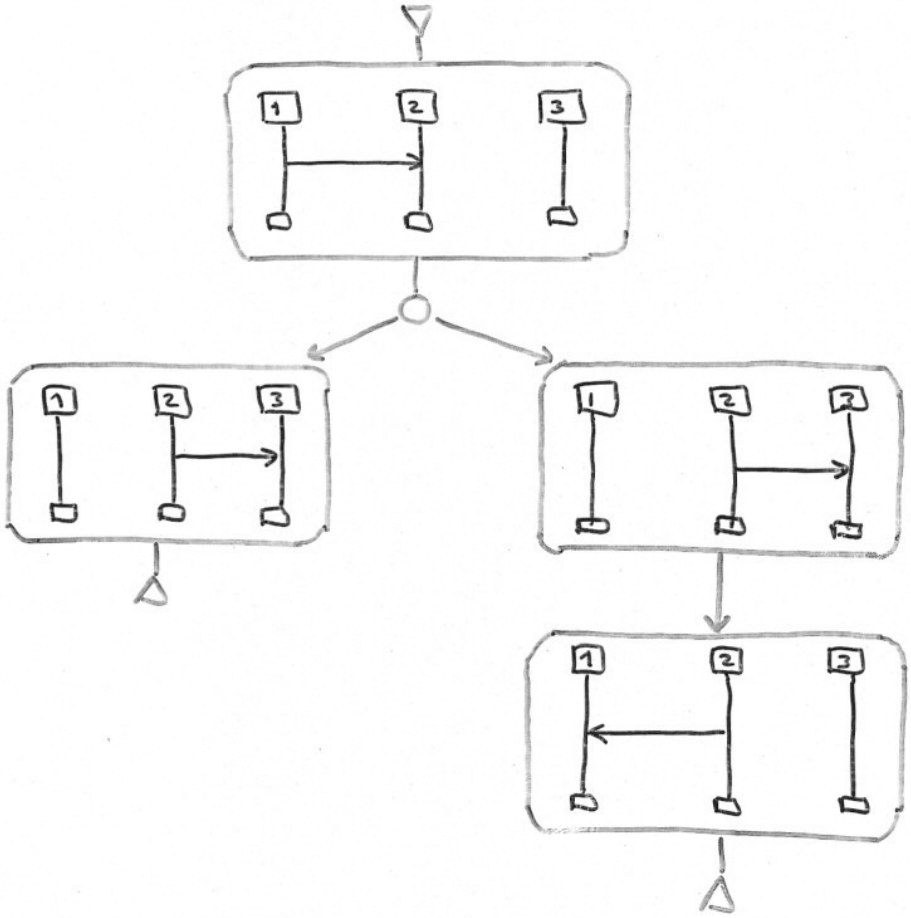
  (1) $v \in F$ and O_p^v contains a maximal event wrt. $<_p$ in MSC $\lambda(v)$

  or

  (2) $v \notin F$ and $\exists v' \in F \,.\, \pi = v \ldots v'$ is a path in G and O_p^v contains a maximal event wrt. $<_p$ in MSC $\lambda(\pi)$.

EXAMPLE

..... on the black board .....



SUMMARY OF MSC-PART

- MSCs: scenario-based, graphical language
- Used in requirements engineering
- Standardized by ITU, UML, ...
- Formally: MSC $M = (P, E, \mathbb{C}, \ell, m, <)$

  $< \subseteq E \times E$ is a partial order ("visual order")

  $$< = \bigcup_{p \in P} <_p \cup \{ (e, m(e)) \mid e \in E_! \}$$

  + FIFO property is guaranteed
- There is a one-to-one correspondence between MSCs and their linearizations
- M has a race if visual order $\neq$ possible order

  Checking whether M has a race is in PTIME (variant of Floyd-Warshall)
- Sets of MSCs can be conveniently described by Message Sequence Graphs

SUMMARY OF MSGs

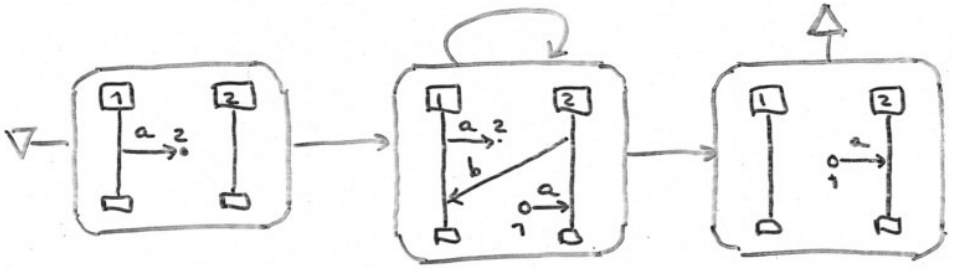
- An MSG is a digraph with a designated set of final vertices, and each vertex is labeled with an MSC.
- The MSC language of MSG G is given by:

  $$L(G) = \{ M(\pi) \mid \pi \text{ is accepting path in } G \}$$
- Checking whether an MSG has a race is undecidable.
- State space of an MSG is context-sensitive
- Checking whether MSG is local choice is in PTIME
- The decision problem "is $L(G_1) \cap L(G_2) = \emptyset$" is undecidable.
- There are (regular) MSCs that cannot be represented by an MSG

  $\Rightarrow$ compositional MSGs (and MSCs)

SUMMARY OF CMSGs

- A CMSC allows $m: E_! \to E_?$ to be partial
- A CMSG is like an MSG where vertices are labeled with CMSCs.



- Path π in CMSG G is safe if $M(\pi)$ is an MSC
- The existence of a safe path in G is undecidab
  (accepting)
- Checking whether all accepting paths in G are safe can be done in PTIME

SUMMARY OF MPAs

- An MPA is a set of finite-state automata with a finite set of communication channels of unbounded capacity (but FIFO)
- The MSC language of MPA A is $L(A)$
- Checking whether $L(A) = \emptyset$ for MPA A is undecidable.
- MPA A is $\forall B$-bounded if all its Linearizations (= runs) are B-bounded ($B \in \mathbb{N}$)
- MPA A is $\exists B$-bounded if some of its runs are B-bounded ($B \in \mathbb{N}$)
- MPAs with $|\mathbb{D}| = 1$ are less expressive than MPAs with $|\mathbb{D}| > 1$.
- For every MPA A which is $\forall B$-bounded: Lin(A) is a regular (word) language

SUMMARY OF REALIZABILITY

- MSG G is realizable if $L(G) = L(A)$ for some MPA A
- MSC language $L \subseteq Act^*$ is realizable iff

$$\forall w \in Act^*. \left(\forall p \in P. \exists v \in L. \; w\lceil_p = v\lceil_p \right) \rightarrow w \in L .$$

- Checking whether L is realizable is coNP-complet

$\Rightarrow$ this all refers to finite set of MSCs

- MSG G is safely realizable if it is realizabl by a deadlockfree MPA
- MSC language $L \subseteq Act^*$ is safely realizable iff

$$\forall w \in pref(L). \left(\forall p \in P. \exists v \in L. \; w\lceil_p = v\lceil_p \right) \rightarrow w \in L$$

and

$$\forall w \in pref(L). \left(\forall p \in P. \exists v \in pref(L). \; w\lceil_p = v\lceil_p \right) \rightarrow w \in pref(L)$$

- Checking whether L is safely realizable can be done in PTIME

SUMMARY OF REGULARITY

- MSG G is regular if Traces (G) is a regular language
- Checking whether MSG G is regular is undecidable
- Checking "is regular $L \subseteq Act^*$ an MSC language" is decidable
- Any regular MSC language is realizable by a $\forall$-bounded, deterministic MPA
- MSG G is communication-closed if for any loop in G its comm. graph is strongly connected
- G is communication-closed $\rightarrow$ G is regular
- Checking realizability of a comm.-closed MSG is undecidable; safe realizability is EXPSPACE-complete
- A set D of MSCs is realizable iff there exists a connected regular expression α with $L(\alpha) = D$