

# DASAP Project Meeting Milano Feb 2010

Egon Börger

Calculus for Mobile Ambients (Cardelli & Gordon 1998)

An ASM Model

Università di Pisa, Dipartimento di Informatica, Italy

# Mobile Agents (Cardelli & Gordon)

- ambient processes  $n[P]$  interpreted as process  $P$  located to run at  $n$
- $n[P]$  definable in ASM framework (notation: **amb**  $n$  **in**  $P$ )
- tree structure induced by the nesting of ambients:
  - *ambName*, element of a domain *AmbName*, considered as root of the tree induced by **amb**  $n$  **in**  $P$ , which is also identified with  $n$
  - *locAg*( $n$ ): (possibly empty) dynamic set of (non-ambient) processes, say  $P_1, \dots, P_p$ , called **local agents** of the ambient process and viewed as running at  $n$
  - *subAmb*( $n$ ): (possibly empty) dynamic set of **subambients**, say **amb**  $m_1$  **in**  $Q_1, \dots, \text{amb } m_q \text{ in } Q_q$
  - *ambBody*( $n$ ) =  $P$  in **amb**  $n$  **in**  $P$  is interpreted as parallel composition of the elements of *subtrees*( $n$ )  
$$P = P_1 \mid \dots \mid P_p \mid \text{amb } m_1 \text{ in } Q_1 \dots \mid \text{amb } m_q \text{ in } Q_q$$

# ASM Interpreter for ambient changing operations

- ambient process change by three actions: Entry, Exit, Open
- can be viewed as tree operations performed on *curAmbProc*

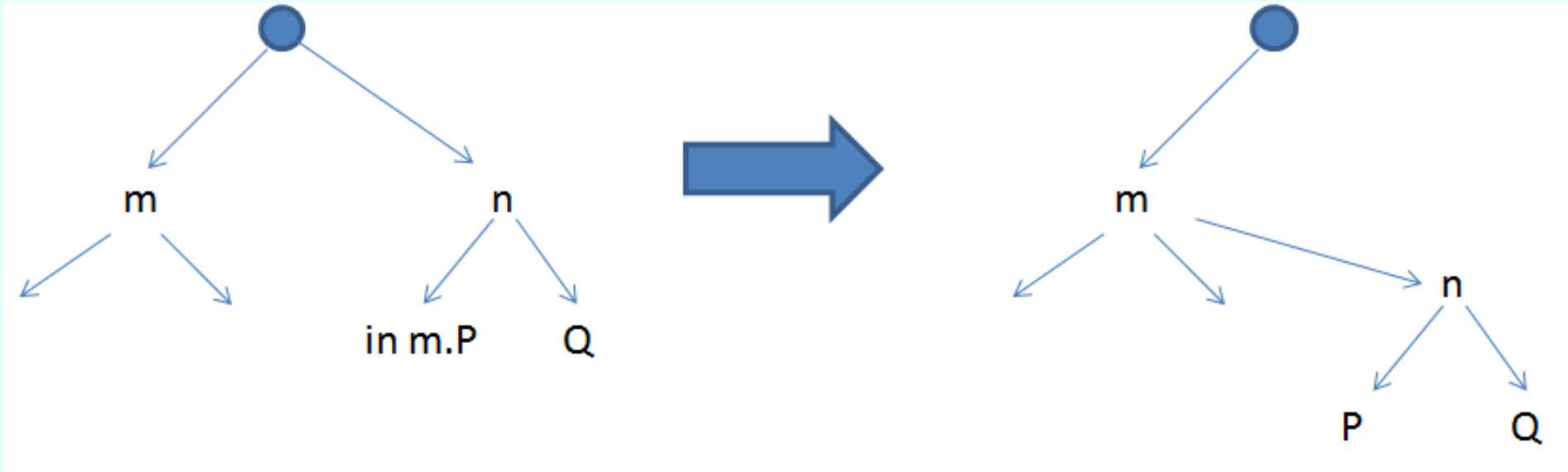
MOBILEAGENTSINTERPRETER =  
choose  $R \in \{\text{ENTRY}, \text{EXIT}, \text{OPEN}\}$  in  
 $R$

- restriction operator definable:  $(\nu n)P = P(n/new(AmbName))$
- none of the remaining 17 structural congruence rules of Cardelli & Gordon needed

# ENTRY into an ambient as subambient

Entry from where?

- Sibling ambient chosen as neighbourhood from where to enter into an ambient  $m$
- $S = n[in\ m.P \mid Q]$  becomes  $n[P \mid Q] \in subtrees(m)$ 
  - if  $sibling(S)$  contains a process with  $ambName\ m$



# ENTRY

= **if**  $EntryAction(curAmbProc) \neq \emptyset$  // there is some entry action  
**then choose**

$S = \mathbf{amb} \ n \ \mathbf{in} \ ((in \ m.P) \mid Q) \in EntryAction(curAmbProc)$

**if**  $sibling(S)$  contains a process with ambient name  $m$  **then**

**choose**  $\mathbf{amb} \ m \ \mathbf{in} \ R \in sibling(S)$

DELETE( $S, subtrees(parent(m))$ )

//  $n$  disappears as sibling of target ambient  $m$

INSERT( $\mathbf{amb} \ n \ \mathbf{in} \ (P \mid Q), subtrees(m)$ )

// modified  $n$  becomes subambient of  $m$

**where**

$EntryAction(curAmbProc) =$

$\{n \in curAmbProc \mid ambBody(n) = (in \ m.P) \mid Q\}$

# Exit of a subambient

Exit to where?

- Sibling ambient chosen as neighbourhood where to exit as subambient of an ambient  $m$



EXIT =

**if**  $ExitAction(curAmbProc) \neq \emptyset$  // there is some exit action  
**then choose**

$S = \mathbf{amb} \ n \ \mathbf{in} \ ((out \ m.P) \mid Q) \in ExitAction(curAmbProc)$

**if**  $parent(n) = m$  **then**

DELETE( $S, subtrees(m)$ ) //  $n$  disappears as subambient of  $m$

INSERT( $\mathbf{amb} \ n \ \mathbf{in} \ (P \mid Q), subtrees(parent(m))$ )

// modified  $n$  becomes sibling ambient of  $m$

**where**  $ExitAction(curAmbProc) =$

$\{n \in curAmbProc \mid ambBody(n) = (out \ m.P) \mid Q\}$

## Ambient dissolving action OPEN

Operating at which level upon a process to open its ambient?

- dissolving the boundary of an ambient named  $m$  “located at the same level”
- sibling ambient chosen as neighbourhood
- replaces a subtree pair ( $open\ m.P$ ,  $amb\ m\ in\ Q$ ) of siblings in  $curAmbProc$  by the new siblings pair ( $P$ ,  $Q$ )





## OPEN

= **if**  $AmbDissolvAction(curAmbProc) \neq \emptyset$   
    // there is some ambient dissolving action  
**then choose**  $S_1 = open\ m.P_1 \in AmbDissolvAction(curAmbProc)$   
**if**  $sibling(S_1)$  contains a process with ambient  $m$  **then**  
    **choose**  $S_2 = \mathbf{amb}\ m\ \mathbf{in}\ P_2 \in sibling(S_1)$     **let**  $p = parent(S_1)$   
    **forall**  $i \in \{1, 2\}$   
        DELETE( $S_i, subtrees(p)$ )  
        INSERT( $P_i, subtrees(p)$ )

**where**

$AmbDissolvAction(curAmbProc) =$   
     $\{open\ m.P_1 \mid open\ m.P_1 \in curAmbProc\}$

$X$  contains a process with ambient  $m =$   
    **forsome**  $Q$     **amb**  $m$  **in**  $Q \in X$

# References

- L. Cardelli and A. D. Gordon, Mobile Ambients. FoSSaCS'98, LNCS 1378.
- E. Börger and R. Stärk, *Abstract State Machines*. A Method for High-Level System Design and Analysis. – Springer-Verlag 2003.