Dynamic Analysis of ECATNets

Noura Boudiaf * & Allaoua Chaoui**
Département d’Informatique, Université d’Oum El Bouaghi, Algérie
boudiafn@yahoo.com

**Department of Software Engineering, Faculty of IT, University of Philadelphia, Jordan
chaoui@philadelphia.edu.jo
Outline

- ECATNets
- Motivations
- Study of Unbounded Places Case
- Dynamic Analysis
- Example
- Conclusion & future work
ECATNets

- Characteristics
  - Sound semantics based on rewriting logic
  - Simulation
  - Model checking verification
  - Static analysis
  - Presence of tool based Maude system
**ECATNets**

## Syntax

- Integration of:
  - Abstract data types & Petri Nets

\[
IC(p, t), \ DT(p, t) \text{ and } CT(p', t) : \text{multi-sets of algebraic terms}
\]

\[
TC(t) : \text{boolean expression}
\]
Semantic

- A transition $t$ is enabled if:
  - $IC(p, t)$ is included in $M(p)$ for every input place $p$
  - $Tc(t)$ is true
  - the addition of $CT(t, p')$ to each output place $p'$ must not result in $p'$ exceeding its capacity when this capacity is finite
- If $t$ is fired then:
  - $DT(p, t)$ is removed from the input place $p$
  - $CT(t, p')$ is added to the output place $p'$
Semantic : Rewriting Rules

We use the notation:

- \((p, [m]_+)\),
  - \(p\) is a place of the net,
  - \([m]_+\) a multi-set of algebraic terms
  - The multi-set union on the pairs \((p, [m]_+)\) will be denoted by \(\otimes\) w.r.t. the ACI (Associativity Commutativity Identity) axioms for \(\otimes\)
Semantic : Rewriting Rules

- **IC(p,t)** is of the form \([m]_\oplus\)

- **case1** \([IC(p,t)]_\oplus = [DT(p,t)]_\oplus\)
  - The form of the rule is then given by:
    \[
    t : (p, [IC(p,t)]_\oplus) \rightarrow (p', [CT(t,p')]_\oplus)
    \]

- **case2** \([IC(p,t)]_\oplus \cap [DT(p,t)]_\oplus = \phi_M\)
  - The form of the rule is given by:
    \[
    t : (p, [IC(p,t)]_\oplus) \otimes (p, [DT(p,t)]_\oplus) \cap [M(p)]_\oplus
    \rightarrow (p, [IC(p,t)]_\oplus) \otimes (p', [CT(t,p')]_\oplus)
    \]

- **Case3** \([IC(p,t)] \cap [DT(p,t)] \neq \phi_M\)
  - that it could be brought to the two already treated cases
ECATNets

- **IC(p,t) is of the form \( \sim[m] \)**
  - The form of the rule is given by:
    
    \[
    t : (p, \{D(p,t) \land M(p)\}) \rightarrow (p',\{C(t,p')\})
    \]
    
    if \( ([IC(p,t) \land ([IC(p,t) \lor [M(p)]) = f \rightarrow \text{false}]

- **IC(p,t) = empty**
  - The form of the rule is given by:
    
    \[
    t: (p, \{D(p,t) \land [M(p)]\}) \rightarrow (p',[C(t,p')] )\]
    
    if \( ([M(p) \rightarrow f \rightarrow \phi_M]

- When the place capacity \( C(p) \) is finite, the conditional part of the rewrite rule will include the following component:
  - \( ([C(p,t) \land [M(p)] \rightarrow [C(p)] \times (\text{Cap})] \)
Motivations

- Maude Model Checking requires finite state space
- Lack of any tool for Reachability Analysis of ECATNets
- Easy implementation of Dynamic Analysis tool under Maude system
  - Reflectivity of rewriting logic: The power of this logic to interpret itself allows the modeling of an ECATNet and then act on it
Motivations

Main Aim

- Enrichment of ECATNets’ possibilities verification
  - Static proprieties of infinite system or finite system
    - Liveness properties, Deadlock detection, …
Study of Unbounded Places Case

- **Unbounded place**
  - The number of algebraic terms in this place increases infinitely

- **We exclude from our study the cases:**
  - \((IC(p,t)\) is of the form \(\sim[m]_\oplus\), and \(IC(p,t) = \text{empty}\)
Study of Unbounded Places Case

- **ECATNet's places with only infinite capacity**
  - (Absence / Presence) of transitions conditions
    - Monotony respected

- **ECATNet's places with finite capacity & infinite capacity**
  - (Absence / Presence) of transitions conditions
    - Monotony is not always respected: The following proved proposition allows detecting infinite increase of terms in places with infinite capacity
Proposition. If $M \xrightarrow{s} M'$ and $M \subseteq M'$ and the first transition in $S$ become not enabled since $M'$. If it exists $S'$ such that $M' \xrightarrow{s} M''$. $S'$ stops when $S$ become enabled. If we have the following case:

$M \xrightarrow{s} M' \xrightarrow{s} M'' \xrightarrow{s} M'''$ and if $M' \subseteq M'''$ and $M'''(p) \subseteq M(p)$ for each place $p$ with bounded capacity yielding $S$ disabled at $M'$, then every infinite place $p'$ such $M'(p') \subset M'''(p')$ is an unbounded place.
For every multi-set \( m \) whose elements are of type \( T \), we create \( \omega_T \) which covers the increase of number of elements in this multi-set:

- \( \forall m : T \)
- \( m \cup \omega_T = \omega_T \)
- \( m \subseteq \omega_T \)
- \( \omega_T \setminus m = \omega_T \)
- \( \omega_T \subseteq \omega_T \)
- \( \omega_T \cup \omega_T = \omega_T \)
In rewriting logic

- \((p, \omega_T) \otimes (p, \omega_T) = (p, \omega_T)\)
- \((p, e) \otimes (p, \omega_T) = (p, \omega_T)\) where \(e\) is an algebraic term of type \(T\). By recurrence:
- \((p, e_1) \otimes \ldots \otimes (p, e_n) \otimes (p, \omega_T) = (p, \omega_T)\) where \(e_1, \ldots, e_n\) are of type \(T\) and
- If we have a rewriting rule of the form \((p, e_1) \otimes \ldots \otimes (p, e_n) \otimes m \rightarrow m'\) if \(C\) (\(m\) doesn't concern \(p\)), then we add rewriting rule:
  \((p, \omega_T) \otimes m \rightarrow (p, \omega_T) \otimes m'\) if \(C\).
The idea of the algorithm is simple:

We check the path from root (initial marking) until the node of current marking to be calculated:

- If conditions of unbounded places are true
  - We calculate this marking by covering these places
- Else
  - We calculate the marking in usual way
Example: Production Cell

- Cell of production that manufactures forged pieces of metal with the help of a press
- The cell is composed of a table A that serves to feed the cell by raw pieces, of a robot of handling, a press and a table B that serves to the storage of forged pieces
- The robot includes two arms, disposed at right angles on one same horizontal plan, interdependent of one same axis of rotation and without vertical mobility possibility
Example: ECATNet Modeling Production Cell

•ECATNet Model of the cell
Example : Places

- ECATNet Places.
- Ta : table A ; set, possibly empty, of raw pieces.
- Tb : table B ; set, possibly empty, of raw pieces.
- Ar1 : arm 1 of robot ; at most a raw piece.
- Ar2 : arm 2 of robot ; at most a forge piece.
- Pr : press ; at most a raw piece or a forge piece.
- Pos-I : initial spatial position of robot ; it is marked "ok" if it is the current position of robot.
- Pos-S : secondary spatial position of robot ; it is marked "ok" if it is the current position of robot.
- EA : this place is added for testing if the tow arms of robot are empty.
Example : Transitions

- **ECATNet Transitions.**
- T1 : Taking of a raw piece by the arm 1 of the robot.
- T2 : Taking of a forge piece by the arm 2 of the robot.
- D1 : deposit of a raw piece in the press.
- D2 : deposit of a forge piece on the table B.
- TS1, TS2 : rotation of the robot from its initial position towards its secondary position.
- TI : rotation of the robot from its secondary position towards its initial position.
- F : forge of the raw piece introduced in the press.
- E : deposit of a raw piece on the table A.
- R : removing forge pieces from the table A.
Example: Rewriting Rules

Rewriting Rules.

[T1] : (Ta, raw) → (Ar1, raw) \quad \text{if } (M(\text{Pos-I}) \rightarrow \text{ok})
\quad \text{and } ((Ar1, \text{raw}) \otimes M(Ar1) \cap C(Ar1)) \rightarrow (Ar1, \text{raw}) \otimes M(Ar1)

[T2] : (Pr, forge) → (Ar2, forge) \quad \text{if } (M(\text{Pos-I}) \rightarrow \text{ok})
\quad \text{and } ((Ar2, \text{raw}) \otimes M(Ar2) \cap C(Ar2)) \rightarrow (Ar2, \text{raw}) \otimes M(Ar2)

[D1] : (Ar1, raw) → (Pr, raw) \otimes (Ea, Ear1) \quad \text{if } (M(\text{Pos-S}) \rightarrow \text{ok})

[D2] : (Ar2, forge) → (Tb, forge) \otimes (Ea, Ear2) \quad \text{if } (M(\text{Pos-S}) \rightarrow \text{ok})

[TS1] : (Pos-I, ok) → (Pos-S, ok) \quad \text{if } (M(Ar1) \rightarrow \text{raw})

[TS2] : (Pos-I, ok) → (Pos-S, ok) \quad \text{if } (M(Ar2) \rightarrow \text{forge})

[TI] : (Pos-S, ok) \otimes (Ea, Ear1) \otimes (Ea, Ear2) → (Pos-I, ok) \quad \text{if } (M(\text{Pos-S}) \rightarrow \text{ok})

[F] : (Pr, raw) → (Pr, forge)

[E] : \phi → (Ta, raw)

[R] : (Tb, forge) → \phi
1. \((\text{Pos-I, ok}) \otimes (\text{Ea, Ear2}) \xrightarrow{E} \)

2. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-I, ok}) \otimes (\text{Ea, Ear2}) \xrightarrow{T_1} \)

3. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-I, ok}) \otimes (\text{Ar1, r}) \otimes (\text{Ea, Ear2}) \xrightarrow{T_{S1}} \)

4. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-S, ok}) \otimes (\text{Ar1, r}) \otimes (\text{Ea, Ear2}) \xrightarrow{D_1} \)

5. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-S, ok}) \otimes (\text{Pr, r}) \otimes (\text{Ea, Ear1}) \otimes (\text{Ea, Ear2}) \xrightarrow{F} \)

6. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-S, ok}) \otimes (\text{Ea, Ear1}) \otimes (\text{Ea, Ear2}) \otimes (\text{Pr, f}) \xrightarrow{T_1} \)

7. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-I, ok}) \otimes (\text{Pr, f}) \xrightarrow{T_1} \)

8. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-I, ok}) \otimes (\text{Pr, f}) \otimes (\text{Ar1, r}) \xrightarrow{T_2} \)

9. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-I, ok}) \otimes (\text{Ar2, f}) \otimes (\text{Ar1, r}) \xrightarrow{T_{S1}, T_{S2}} \)

10. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-S, ok}) \otimes (\text{Ar2, f}) \otimes (\text{Ar1, r}) \xrightarrow{D_1} \)

11. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-S, ok}) \otimes (\text{Ar2, f}) \otimes (\text{Pr, r}) \otimes (\text{Ea, Ear1}) \xrightarrow{D_2} \)

12. \((\text{Ta, } \omega_{\text{Raw}}) \otimes (\text{Pos-S, ok}) \otimes (\text{Pr, r}) \otimes (\text{Ea, Ear1}) \otimes (\text{Ea, Ear2}) \otimes (\text{Tb, } \omega_{\text{Forge}}) \)
Rewriting logic based-tool for the implementation of such algorithm is under development

- Not Mature
  - Space Reduction
- Extending this algorithm for ECATNets with arcs inhibitor