Predicting global usages of resources endowed with local policies

Chiara Bodei, Viet D. Dinh, and Gian L. Ferrari

University of Pisa

Sept 10, 2011
Content

- Motivations
- Syntax and Semantics
- An example
- Control Flow Analysis
- Conclusion and Future Work
Motivations
Motivations

Applications

Resources

Predicting global usages of resources endowed with local policies
Motivations

- A *formal model* for modern distributed systems

Applications

- \texttt{req r}_3
- P_1, P_2, P_3

Resources

- r_1, r_2, r_3, r_4
- Q_1
Motivations

- A formal model for modern distributed systems
  - Resource awareness: mechanisms to express and enforce policies governing resource usages
Motivations

- A formal model for modern distributed systems
  - Resource awareness: mechanisms to express and enforce policies governing resource usages
  - Reasoning techniques: static and dynamic checks of the program behavior

Predicting global usages of resources endowed with local policies

University of Pisa
A Process Calculus Approach

G-Local $\pi$-Calculus:

- processes as loosely couple entities with
  - primitives based on $\pi$-calculus
  - primitives to acquire and discharge resources
- resources as stateful entities with
  - local state + global properties
Representing resources

A triple of

- $r$: a resource name
- $\varphi$: a policy
  - $\varphi$ is specified by resource-aware logics
- $\eta$: a trace of actions
Resource management

Two primitives for managing resource boundaries:

- a resource joint point: \((r, \varphi, \eta)\{Q\}\)
- an action prefix: \(\alpha(r).Q\)
- a resource request point: \(\text{req}(r)\{Q\}\)
Semantics - Samples

- Available resources: $(r, \varphi, \epsilon)\{0\}$
- A structural rule of resource movement:
  $$(r_1, \varphi_1, \eta_1)\{(r_2, \varphi_2, \eta_2)\{0\} \parallel Q\} \equiv (r_2, \varphi_2, \eta_2)\{0\} \parallel (r_1, \varphi_1, \eta_1)\{Q\}$$
- this rule simulates reconfigurations of resources
Semantics - Samples

Resource Usages:

\[
Q \xrightarrow{\alpha?r} Q' \quad \eta.\alpha \models \varphi \\
(r, \varphi, \eta)\{Q\} \xrightarrow{\alpha(r)} (r, \varphi, \eta.\alpha)\{Q'\}
\]

- $\alpha?r$: an open action
- $\alpha(r)$: a closed action
- $\eta' = \eta.\alpha$: an updated state
- $\eta.\alpha \models \varphi$: $\varphi$ is validated on $\eta.\alpha$
Resource Usages (cont.):

\[ Q \xrightarrow{\alpha?r} Q' \quad \eta.\alpha \not\models \varphi \]

\[ (r, \varphi, \eta)\{Q\} \xrightarrow{\alpha(r)} (r, \varphi, \eta)\{0\} \parallel Q' \]

- \( \alpha(r) \): a faulty action
- \( \eta \): is not changed
- \( \eta.\alpha \not\models \varphi \): \( \varphi \) is not validated on \( \eta.\alpha \)
Resource acquisition:

\[ \text{req}(r)\{Q\} \parallel (r, \varphi, \eta)\{0\} \xrightarrow{\tau} (r, \varphi, \eta)\{Q\} \]
Semantics - Samples

Resource release:

- a special action for releasing resources: \texttt{rel}

\[
\frac{Q \xrightarrow{\text{rel}\,?\,r} Q'}{(r, \varphi, \eta)\{Q\} \xrightarrow{\tau} (r, \varphi, \eta.\texttt{rel})\{0\} \parallel Q'}
\]
Semantics - Samples

- Abstract behavior of the resource manager:
  - resource disappearance:
    
    \[ (r, \varphi, \eta) \{0\} \xrightarrow{\tau} 0 \]
Semantics - Samples

- Abstract behavior of the resource manager (cont.):

  - resource appearance:

    $$P \xrightarrow{\tau} P \parallel (r, \varphi, \eta)\{0\}$$
An example - a robot scenario
An example - cont.

- A simple model:

\[
R_1 = \text{req(item)}\{ \text{East(item)}.0 \} \\
R_2 = \text{req(item)}\{ \text{East(item)}.0 \} \\
R'_2 = \text{req(item)}\{ \text{North(item)}.0 \} \\
R_3 = \text{req(item)}\{ \text{North(item)}.0 \}
\]

\[
S = (\text{item}, \varphi, \epsilon)\{ 0 \} \parallel R_1 \parallel R_2 \parallel R'_2 \parallel R_3
\]

- the policy \( \varphi \) constrains movement of the item on the grid.
Control Flow Analysis (CFA)

for a sub-calculus:

- given a fixed amount of resources
- not consider dynamic reconfiguration of resources
- process using resources are sequential
Control Flow Analysis (CFA)

- CFA represents an abstraction of the actual executions
- Concretisation cannot be precise

**Static**
- Trace $\eta$ is included
- Trace $\eta$ is not included

**Dynamic**
- Trace $\eta$ can happen
- Trace $\eta$ never happens
Control Flow Analysis - cont.

What are abstractions of behaviour?

- $\rho$ - name bindings:
  - $a \in \rho(x)$: $a$ can be bound to $x$

- $\kappa$ - values sent on each channel:
  - $a \in \kappa(x)$: $a$ can be sent on $x$

- $\Gamma$ - resource behaviour:
  - $(\varphi, \eta) \in \Gamma(r)$ then $\eta$ - a possible trace over $r$, respecting $\varphi$

- $\psi$ - a set of possible locked resources:
  - given $(r, \varphi, \eta)\{\alpha(r').P\}$, where $\alpha$ does not reach $r'$, then $r$ is locked
Control Flow Analysis - cont.

- **Judgment of analysis**: \( (\rho, \kappa, \Gamma, \psi) \models^\delta P \)

- **Subject Reduction**: \( (\rho, \kappa, \Gamma, \psi) \models^\delta P \) and \( P \xrightarrow{\tau}^* P' \), then \( (\rho, \kappa, \Gamma, \psi) \models^\delta P' \).

- **Existence of solutions** \( (\rho, \kappa, \Gamma, \psi) \): there always exists a least solution
CFA with the robot scenario

- with annotations:

\[
\begin{align*}
R_1 &= \text{req}(item)\{ \text{East}(item).0 \}^{\chi_1} \\
R_2 &= \text{req}(item)\{ \text{East}(item).0 \}^{\chi_2} \\
R'_2 &= \text{req}(item)\{ \text{North}(item).0 \}^{\chi_2'} \\
R_3 &= \text{req}(item)\{ \text{North}(item).0 \}^{\chi_3} \\
S &= (item, \varphi, \epsilon)\{ 0 \}^{\chi_0} \parallel R_1 \parallel R_2 \parallel R'_2 \parallel R_3
\end{align*}
\]

- CFA result on traces of item:
  - \((\text{East}(item)\text{.out\_err}(\chi_2'), \chi_0 \cdot \chi_1 \cdot \chi_2')\)
    - this is a faulty trace!
  - \((\text{East}(item)\text{.East}(item)\text{.North}(item), \chi_0 \cdot \chi_1 \cdot \chi_2 \cdot \chi_3)\)
    - this is a trace leading to the required location
Conclusion

We propose a process calculus with resource-awareness:

- specifying properties of resources
- mechanisms to control resource usages
- CFA for a sub-calculus

Future work:

- investigate logics for specifying policies
  - focus on decidability
- polyadic request primitives
- behavioral types for the full calculus
  - model checking of behavioral types
Q & A

Thank you!

and questions?
Future Works

- more effective views of resources:
  - resource coordination: restrict resource movement

\[
(r_1, \varphi_1, \eta_1) \{ (r_2, \varphi_2, \eta_2) \{ 0 \} \parallel Q \} \\
\equiv (r_2, \varphi_2, \eta_2) \{ 0 \} \parallel (r_1, \varphi_1, \eta_1) \{ Q \}
\]
Future Works

- more effective views of resources:
  - resource coordination: restrict resource movement

\[
(r_1, \varphi_1, \eta_1) \{ (r_2, \varphi_2, \eta_2) \{ 0 \} \parallel Q \}
\equiv (r_2, \varphi_2, \eta_2) \{ 0 \} \parallel (r_1, \varphi_1, \eta_1) \{ Q \}
\]