Software Model Checking

Algebra of Communicating Shared Resources (ACSR)

Moonzoo Kim

CS Dept. KAIST

Outline

- Real-time systems
- Resource-bound computation
- ACSR
  - Algebra, analysis techniques, examples

Real-time systems

- Correctness and reliability of real-time systems depends on
  - Functional correctness
  - Temporal correctness
  - Failures
- Factors that affect behavior:
  - Synchronization and communication
  - Availability of resources and scheduling
    - Computational systems are always constrained in their behavior
    - Resources capture physical constraints
    - Resources should be used as a primitive notion in modeling and analysis
    - Resource-bound computation is a general framework of wide applicability

Design Issues

- Time -- discrete time, dense time
- Concurrency Semantics -- interleaving, synchronous lock step, true parallelism
- Operators -- prefix, choice, parallel, restriction, recursion
- Timed Operators -- delay, timeout, bound (deadline)
- Communication -- two-synchronous, n-way
- Abstraction -- hiding, restriction
- Resource -- implicit, explicit, unlimited, bounded
- Priorities -- static, dynamic

ACSR (Algebra of Communicating Shared Resources) has two types of actions:

1. Timed Actions -- represent the passage of time and resource consumption (e.g., CPUs)
2. Instantaneous Events -- provide a synchronization between processes.

A labelled transition system:

$$P_0 \xrightarrow{\varnothing} P_1 \xrightarrow{NC} P_2 \xrightarrow{\{gate, train\}} P_3 \xrightarrow{\{gate, train\}} P_4 \xrightarrow{IC} \ldots$$

Timed Actions

- A finite set of serially reusable resources, denoted by $\mathcal{R}$
- The domain, $\mathcal{D}_R = \mathbb{P}(\mathcal{R} \times \mathbb{N})$ with the restriction that each resource be represented at most once, e.g., $\{(r,p)\}$, $\{(r_1,p_1),(r_2,p_2)\}$, $\emptyset$
- $\rho(A)$ denotes the set of resources used by the action *A* e.g. $\rho(\{(r_1,p_1),(r_2,p_2)\}) = \{r_1,r_2\}$
- $\pi_r(A)$ denotes the priority level of the action *A* in the resource *r; e.g.,* $\pi_{r1}(\{(r_1,p_1),(r_2,p_2)\}) = p_1$
  If *r is not in* $\rho(A)$ *, then* $\pi_r(A) = 0$
- *A, B,* and *C* range over $\mathcal{D}_R$

Instantaneous Events

- An event is denoted by a pair *(a, p)*, where *a* is the *label* of the event, and *p* is its *priority*
- Labels are drawn from the set $\mathcal{A} \cup \bar{\mathcal{A}} \cup \{\tau\}$;
- The special label, τ, arises when two events with ($\bar{\bar{a}} = a$) inverse labels (e.g., $a, \bar{a}$) are executed in parallel.
- $\mathcal{D}_E$ denotes the domain of events. $l(e)$ and $\pi(e)$ to represent the label and priority
- *e*, *f* and *g* range over $\mathcal{D}_E$
- The entire domain of actions is $\mathcal{D} = \mathcal{D}_R \cup \mathcal{D}_E$, and we let α and β range over $\mathcal{D}$

Resources

- Resources capture constraints on executions
- Resources can be
  - Serially reusable:
    - processors, memory, communication channels
  - Consumable
    - power
- Resource capacities
  - Single-capacity resources
  - Multiple-capacity resources
  - Time-sliced, etc.

Events

- Events represent communication
  - events are instantaneous
  - point-to-point communication across channels
  - prioritized access to channels
  - input and output events

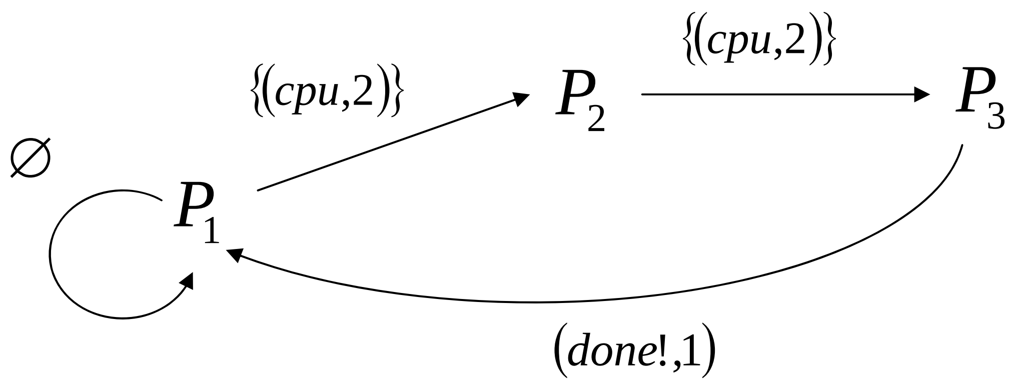
$$(e?, p_1) \qquad\qquad (e!, p_2)$$

- Actions represent computation
  - actions take time
  - require access to resources
  - each resource has priority of access

    $$A = \{(r_1, p_1), (r_2, p_2)\}$$
  - each resource can be used at most once
  - resources of action A: $\rho(A)$
  - idling action: $\varnothing$

Computation model

- A specification is composed of processes
- Processes evolve by performing events and actions



Syntax for ACSR processes

- Process terms

$$
\begin{array}{lll}
P & ::= & NIL \\
 & | & A : P \\
 & | & (a, n).P \\
 & | & P + P \\
 & | & P \parallel P \\
 & | & P \Delta_t^a (Q, R, S) \\
 & | & [P]_I \\
 & | & P \setminus F \\
 & | & b \rightarrow P \\
 & | & C
\end{array}
$$

- Process names

$$C \overset{def}{=} P$$

ACSR semantics

- Two-level semantics:
  - A collection of inference rules gives unprioritized transition relation

$$P \xrightarrow{\alpha} P'$$

  - A preemption relation on actions and events disables some of the transitions, giving a prioritized transition relation

$$P \xrightarrow{\alpha}_{\pi} P'$$

Unprioritized transition relation

- Prefix operators

ActT $\dfrac{-}{A:P \xrightarrow{A} P}$ **ActI** $\dfrac{-}{(a,p):P \xrightarrow{(a,p)} P}$

- Choice

ChoiceL $\dfrac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$

- Parallel

ParIL $\dfrac{P \xrightarrow{(a,p)} P'}{P \,||\, Q \xrightarrow{(a,p)} P' \,||\, Q}$

Unprioritized transition relation (II)

- Resource-constrained execution

ParT $$\frac{P \xrightarrow{A_1} P' \quad Q \xrightarrow{A_2} Q'}{P \| Q \xrightarrow{A_1 \cup A_2} P' \| Q'} \quad \rho(A_1) \cap \rho(A_2) = \varnothing$$

- Priority-based communication

ParCom $$\frac{P \xrightarrow{(a?,p_1)} P' \quad Q \xrightarrow{(a!,p_2)} Q'}{P \| Q \xrightarrow{(\tau, p_1 + p_2)} P' \| Q'}$$

- Resource reservation

CloseT $$\frac{P \xrightarrow{A_1} P'}{[P]_I \xrightarrow{A_1 \cup A_2} [P']_I} \quad A_2 = \{(r,0) \mid r \in I - A_1\}$$

$$\textbf{ActT} \quad \frac{-}{A : P \xrightarrow{A} P}$$

$$\textbf{ActI} \quad \frac{-}{(a, n).P \xrightarrow{(a,n)} P}$$

- E.g., The process $\{(r_{1,}\, p_1), (r_{2,}\, p_2)\}{:}P$ simultaneously uses resources r_1 and r_2 for one time unit, and then executes *P.*
- The process $(a,p).P$ executes the event "(a, p)" and proceeds to *P*

Choice

$$\textbf{ChoiceL} \quad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

$$\textbf{ChoiceR} \quad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

- E.g., $(a,7).P + \{(r_1, 3),(r_2, 7)\}:Q$ may choose between executing the event $(a, 7)$ (from **ActI**) or the time-consuming action $\{(r_1, 3),(r_2, 7)\}$ (from **ActT**).

Parallel

$$\textbf{ParT}\quad \frac{P \xrightarrow{A_1} P',\ Q \xrightarrow{A_2} Q'}{P||Q \xrightarrow{A_1 \cup A_2} P'||Q'} \qquad (\rho(A_1) \cap \rho(A_2) = \emptyset)$$

- The condition $\rho(A_1) \cap \rho(A_2) = \emptyset$ ensures that at most one process uses a single resource during any time step.

$$\textbf{ParIL}\quad \frac{P \xrightarrow{(a,n)} P'}{P||Q \xrightarrow{(a,n)} P'||Q}$$

$$\textbf{ParIR}\quad \frac{Q \xrightarrow{(a,n)} Q'}{P||Q \xrightarrow{(a,n)} P||Q'}$$

$$\textbf{ParCom}\quad \frac{P \xrightarrow{(a,n)} P',\ Q \xrightarrow{(\bar{a},m)} Q'}{P||Q \xrightarrow{(\tau,n+m)} P'||Q'}$$

Example 1

$$P \stackrel{\text{def}}{=} ((a,3).P_1) + (\{(r_3,8)\} : P_2)$$
$$Q \stackrel{\text{def}}{=} ((\bar{a},5).Q_1) + (\{(r_1,7)\} : P_2)$$

$P||Q$ admits the following four transitions:

$$P||Q \xrightarrow{(a,3)} P_1||Q \qquad \text{[by } \textbf{ParIL}\text{]}$$
$$P||Q \xrightarrow{(\bar{a},5)} P||Q_1 \qquad \text{[by } \textbf{ParIR}\text{]}$$
$$P||Q \xrightarrow{(\tau,8)} P_1||Q_1 \qquad \text{[by } \textbf{ParCom}\text{]}$$
$$P||Q \xrightarrow{\{(r_1,7),(r_3,8)\}} P_2||Q_2 \qquad \text{[by } \textbf{ParT}\text{]}$$

Example 2(Why add priorities in ParCom?)

$$P \stackrel{\text{def}}{=} (a,2).P_1 + (a,3).P_2$$
$$Q \stackrel{\text{def}}{=} (\bar{a},5).Q_1 + (\bar{a},3).Q_2$$

In P the second choice is preferred, while in Q the first choice is preferred.

$P||Q$ can:

$$P||Q \xrightarrow{(a,2)} P_1||Q \qquad P||Q \xrightarrow{(a,3)} P_2||Q$$
$$P||Q \xrightarrow{(\bar{a},5)} P||Q_1 \qquad P||Q \xrightarrow{(\bar{a},3)} P||Q_2$$
$$P||Q \xrightarrow{(\tau,7)} P_1||Q_1 \qquad P||Q \xrightarrow{(\tau,5)} P_1||Q_2$$
$$P||Q \xrightarrow{(\tau,8)} P_2||Q_1 \qquad P||Q \xrightarrow{(\tau,6)} P_2||Q_2$$

Note that the τ-transition with the highest priority is that associated with the derivative $P_2||Q_1$.

These transitions had the highest priorities in their original constituent processes. □

Scope(sequential composition, timeout and interrupt)

- for “continue”

$$\textbf{ScopeCT} \quad \frac{P \xrightarrow{A} P'}{P \,\triangle_t^b\, (Q,R,S) \xrightarrow{A} P' \,\triangle_{t-1}^b\, (Q,R,S)} \quad (t>0)$$

$$\textbf{ScopeCI} \quad \frac{P \xrightarrow{(a,n)} P'}{P \,\triangle_t^b\, (Q,R,S) \xrightarrow{(a,n)} P' \,\triangle_t^b\, (Q,R,S)} \quad (t > 0)$$

- for "end"

$$\textbf{ScopeE}\quad \frac{P \xrightarrow{(b,n)} P'}{P \,\triangle_t^b\, (Q,R,S) \xrightarrow{(\tau,n)} Q} \qquad (t > 0)$$

- for "timeout"

$$\textbf{ScopeT}\quad \frac{R \xrightarrow{\alpha} R'}{P \,\triangle_t^b\, (Q,R,S) \xrightarrow{\alpha} R'} \qquad (t = 0)$$

- for "interrupt"

$$\textbf{ScopeI}\quad \frac{S \xrightarrow{\alpha} S'}{P \,\triangle_t^b\, (Q,R,S) \xrightarrow{\alpha} S'} \qquad (t > 0)$$

Preemption relation

- α is preempted by β : $\alpha \prec \beta$
- action preempts action $\{(r_1,3),(r_2,5)\} \prec \{(r_1,7),(r_2,5)\}$
  - no lower priorities: $\forall r \in \rho(\alpha), \pi_r(\alpha) \le \pi_r(\beta)$
  - some higher priorities: $\exists r \in \rho(\beta), \pi_r(\alpha) < \pi_r(\beta)$
  - $\rho(\beta) \subseteq \rho(\alpha)$
- event preempts event $(a!,1) \prec (a!,3)$
  - same label, higher priority
- event preempts action $(\tau,1) \prec \{(r,4)\}$
  - τ with non-zero priority preempts all actions

Prioritized transition relation

- We define

$$P \xrightarrow{\alpha}_{\pi} P'$$

when

- there is an unprioritized transition

$$P \xrightarrow{\alpha} P'$$

- there is no $P \xrightarrow{\beta} P''$ such that $\alpha \prec \beta$

Example

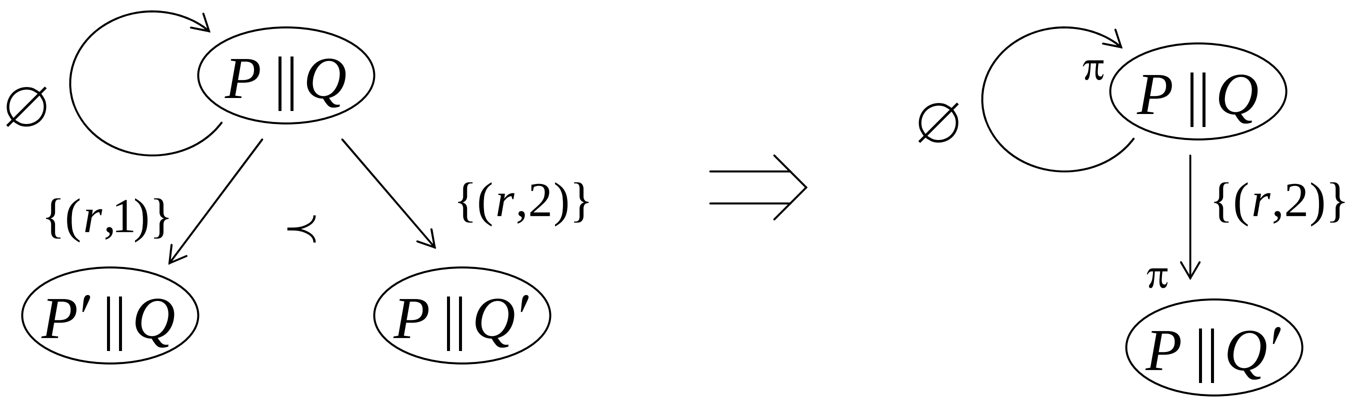
- Resource conflict:

$$P = \{(r,1)\} : P' \qquad Q = \{(r,2)\} : Q' \qquad P \,\|\, Q \sim NIL$$

- Processes must provide for preemption

$$P = \{(r,1)\} : P' + \varnothing : P \qquad Q = \{(r,2)\} : Q' + \varnothing : Q$$

- Unprioritized and prioritized transitions:



Example (cont.)

- Resource reservation enforces progress