Distributed Resource Allocation Protocol Verification in Event-B:

NII Model Notes

Paulius Stankaitis

Distributed Resource Allocation Model Description

Request. An agent A_i which intends to lock a set of resources $res \subseteq R$ generates a request to request pools associated with resources $\mathbf{r}$. Such requests are sent and received in no particular order and contain only agent name A_i. We define such message as $\mathsf{request(A_i)}$ and write $\mathsf{request(A_i)} \rightarrow \mathsf{r_k}$ to state it is addressed to resource $r_k \in \mathbf{r}$.

Reply. Once a request pool r_k receives request $\mathsf{request(A_i)}$ it replies with a message $\mathsf{reply(r_k, pp(r_k))} \rightarrow A_i$ and then increments $\mathsf{pp}(r_k)$.

ConfirmWR. After sending all $\mathsf{request(A_i)}$ messages an agent A_i awaits for all replies to arrive which carry values $\mathsf{pp}(r_k)$. Depending on these values following actions should be taken:

Write. If all reply values on reception are equal then A_i should write at index n to request pools $\mathsf{write(A_i, n)} \rightarrow \mathsf{r_k}$.

sRequest. If all values on reception are not equal then the agent must renegotiate a new index. This time an agent sends new (special) requests to a subset of resources. We define such message as $\mathsf{srequest(A_i, max)} \rightarrow \mathsf{r_k}$ where r_k is $\mathsf{r_k} \subset \mathbf{r}$ and must satisfy $\forall \mathsf{r} \cdot \mathsf{r} \in \mathsf{r_k} \Rightarrow \mathsf{reply(r_k)} < \mathsf{max(replies(A_i))}$.

sReply. Once a request pool r_k receives a special request it replies with the following message $\mathsf{reply(r_k, max)} \rightarrow A_i$ where max the maximum value of $\mathsf{pp(r_k)}$ and received $\mathsf{srequest(A_i)}$.

pReady. A pre-ready message is sent by a resource to inform an agent that it is available for consumption and we define such message $\mathsf{pready(r_k)} \rightarrow A_i$.

pReady (wr).

pReady (rl).

Lock. An agent waits for all pre-ready messages to arrive and once it receives them it sends a lock message to resources as follows $\mathsf{lock(A_i)} \rightarrow r_k$.

Respond. A request pool r_k which receives a lock message will respond with a message $\mathsf{respond(r_k, response)} \rightarrow A_i$ where $\mathsf{response} \in \{\mathsf{confirm}, \mathsf{deny}\}$.

Decide. An agent waits for all respond messages to arrive and depending on these messages following actions will be taken.

Unlock. If one of the messages is a deny message, an agent $\mathsf{A_i}$ will send an unlock messages to all resources which replied with confirm message.

Consumption. If all messages were confirm messages, an agent $\mathsf{A_i}$ can proceed with resource consumption.

Release. An agent A_i will eventually release a resource by sending a message to to resource.

Distributed Resource Allocation Model Verification

Model Refinement M2.

- Deadlock Freedom (DF).
  - DF: all lock messages must be delete when agent status is consume.
  - DF: all response messages must be delete when agent status is release.
- Distributed Lane Forming (DLF).