Algorithms for Model Checking (2IW55)

Lecture 9

Data Abstraction

Chapter 13

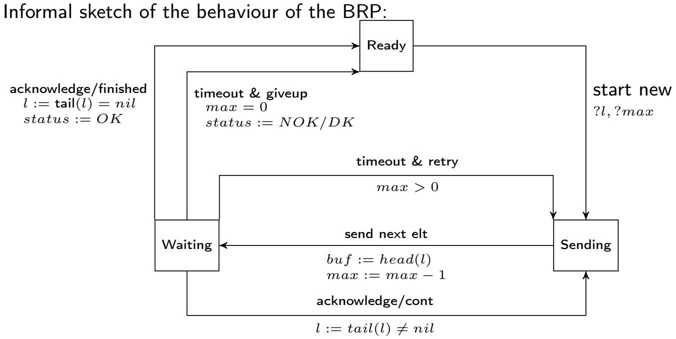
Tim Willemse
(timw@win.tue.nl)
http://www.win.tue.nl/∼timw
HG 6.81

Bounded Retransmission Protocol

The Bounded Retransmission Protocol:

- It sends elements from a list l one by one over an unreliable channel
- On time-out, it will resend elements, at most *max* times
- The status indicates if the list arrived correctly

Informal sketch of the behaviour of the BRP:



Bounded Retransmission Protocol

We have the following variables (v_i) with their domains (D_i)

- l : List[Data]
- max : Nat
- $state$: State $\quad (= \{\text{Ready, Sending, Waiting}\})$
- $status$: Status $\quad (= \{OK, NOK, DK\})$
- buf : Data

Example states (for convenience, take Data = Nat):

- $(l \mapsto [3, 4], max \mapsto 5, state \mapsto \text{Sending}, status \mapsto OK, buf \mapsto 3)$
- $(l \mapsto [4], max \mapsto 0, state \mapsto \text{Waiting}, status \mapsto NOK, buf \mapsto 4)$

Bounded Retransmission Protocol

Specification of the Bounded Retransmission Protocol:

- Initial states: $\mathcal{S}_0 := (state = \text{Ready})$
- Transitions:
  1. $Start\_new\_transmission :=$
     $state = \text{Ready} \wedge state' = \text{Sending}$
  2. $Send\_next\_element :=$
     $state = \text{Sending} \wedge state' = \text{Waiting} \wedge max = max' + 1 \wedge buf' = head(l) \wedge l = l'$
  3. $Get\_acknowledgement :=$
     $state = \text{Waiting} \wedge l' = tail(l) \wedge ((l' = [] \wedge state' = \text{Ready} \wedge status' = OK) \vee (l' \neq [] \wedge state' = \text{Sending} \wedge max' = max))$
  4. $Timeout\_and\_retry :=$
     $state = \text{Waiting} \wedge max > 0 \wedge state' = \text{Sending} \wedge l = l' \wedge max = max'$
  5. $Timeout\_and\_give\_up :=$
     $state = \text{Waiting} \wedge max = 0 \wedge state' = \text{Ready} \wedge ((status' = DK \wedge l = []) \vee (status' = NOK \wedge l \neq []))$
- The full transition relation $\mathcal{R}$ is defined as:

$$\begin{array}{ll} & Start\_new\_transmission \vee Send\_next\_element \\ \vee & Get\_acknowledgement \vee Timeout\_and\_retry \\ \vee & Timeout\_and\_give\_up \end{array}$$

Bounded Retransmission Protocol

- The Kripke Structure underlying the BRP specification is infinite
- The control-aspects of the system can be studied by model checking by abstracting from the data and the counter (a finite abstraction is needed)
- Abstract domains:
  - $A_{\mathsf{List}} := \{empty, non\_empty\}$
  - $A_{\mathsf{Nat}} := \{\cdot\}$
  - $A_{\mathsf{Data}} := \{\cdot\}$
  - $A_{\mathsf{State}} :=$ State
  - $A_{\mathsf{Status}} :=$ Status
- Abstraction mapping:
  - $h(n{:}\mathsf{Nat}) = h(d{:}\mathsf{Data}) = \cdot$
  - $h([]) = empty,\ h(x \vdash l) = non\_empty$
  - $h(s{:}\mathsf{State}) = s,\ h(s{:}\mathsf{Status}) = s$

Technische Universiteit
Eindhoven
University of Technology

- Examples of abstract labels $\widehat{AP}$:
  $\widehat{l} = empty, \widehat{l} = non\_empty, \widehat{max} = \cdot, \widehat{state} = waiting$, etcetera.
- Labels in L':

$$\begin{aligned} & L'((l \mapsto [3,4], max \mapsto 5, state \mapsto \text{Sending}, status \mapsto OK, buf \mapsto 3)) \\ = \\ & (\widehat{l} \mapsto non\_empty, \widehat{max} = \cdot, \widehat{state} = \text{Sending}, \widehat{status} = OK, \widehat{buf} = \cdot) \end{aligned}$$

- So, we can still express properties like:

$$\text{A G } (\widehat{status} = OK \longrightarrow \widehat{l} = empty)$$

Technische Universiteit
Eindhoven
University of Technology

Bounded Retransmission Protocol

Abstract specification of the Bounded Retransmission Protocol:

- Initial states: $\mathcal{S}_0 := (\widehat{state} = \text{Ready})$
- Transitions:
  1. $Start\_new\_transmission :=$
     $\widehat{state} = \text{Ready} \wedge \widehat{state'} = \text{Sending}$
  2. $Send\_next\_element :=$
     $\widehat{state} = \text{Sending} \wedge \widehat{state'} = \text{Waiting} \wedge \widehat{l'} = \widehat{l}$
  3. $Get\_acknowledgement :=$
     $\widehat{state} = \text{Waiting} \wedge ((\widehat{l'} = empty \wedge \widehat{state'} = \text{Ready} \wedge \widehat{status'} = OK) \vee (\widehat{l'} = non\_empty \wedge \widehat{state'} = \text{Sending}))$
  4. $Timeout\_and\_retry :=$
     $\widehat{state} = \text{Waiting} \wedge \widehat{state'} = \text{Sending} \wedge \widehat{l} = \widehat{l'}$
  5. $Timeout\_and\_give\_up :=$
     $\widehat{state} = \text{Waiting} \wedge \widehat{state'} = \text{Ready} \wedge ((\widehat{status'} = DK \wedge \widehat{l} = empty) \vee (\widehat{status'} = NOK \wedge l = non\_empty))$
- The full transition relation $\mathcal{R}$ is defined as:

$$\begin{aligned} & Start\_new\_transmission \vee Send\_next\_element \\ \vee\ & Get\_acknowledgement \vee Timeout\_and\_retry \\ \vee\ & Timeout\_and\_give\_up \end{aligned}$$

Technische Universiteit
Eindhoven
University of Technology

Bounded Retransmission Protocol

Informal sketch of the abstract behaviour of the BRP: