Przykład: ABP

Nieformalny opis:

Sender dostawszy komunikat do przesłania, wysyła go wraz z bitem `b`; następnie może on:

- dostać `timeout`; wtedy powtarza wysłanie komunikatu z bitem `b`.
- dostać potwierdzenie `b`; wtedy oczekuje na następny komunikat (który zostanie przesłany z bitem $\hat{\mathtt{b}}$.
- dostać potwierdzenie $\hat{\mathtt{b}}$, które ignoruje (*kiedy jest to możliwe?*).

Receiver wysławszy komunikat odpowiada potwierdzeniem `b`; następnie może on:

- dostać `timeout`; wtedy powtarza wysłanie potwierdzenia `b`.
- dostać nowy komunikat z bitem $\hat{\mathtt{b}}$; wtedy wysyla nowy komunikat (i wysyła potwierdzenie $\hat{\mathtt{b}}$.
- dostać komunikat z bitem `b`, który ignoruje (*kiedy jest to możliwe?*).

(Wariant: *Receiver* bez *Timer*'a, potwierdza dokładnie raz każdy komunikat.)

Formalny opis:

$$\begin{aligned}
Send(\mathtt{b}) &\stackrel{\text{def}}{=} \overline{\mathtt{send}_{\mathtt{b}}}.\overline{\mathtt{time}}.Sending(\mathtt{b})\\
Sending(\mathtt{b}) &\stackrel{\text{def}}{=} \mathtt{timeout}.Send(\mathtt{b}) \;+\; \mathtt{ack}_{\mathtt{b}}.\mathtt{timeout}.Accept(\hat{\mathtt{b}}) \;+\; \mathtt{ack}_{\hat{\mathtt{b}}}.Sending(\mathtt{b})\\
Accept(\mathtt{b}) &\stackrel{\text{def}}{=} \mathtt{accept}.Send(\mathtt{b})
\end{aligned}$$

$$\begin{aligned}
Reply(\mathtt{b}) &\stackrel{\text{def}}{=} \overline{\mathtt{reply}_{\mathtt{b}}}.\overline{\mathtt{time}}.Replying(\mathtt{b})\\
Replying(\mathtt{b}) &\stackrel{\text{def}}{=} \mathtt{timeout}.Reply(\mathtt{b}) \;+\; \mathtt{trans}_{\hat{\mathtt{b}}}.\mathtt{timeout}.Deliver(\hat{\mathtt{b}}) \;+\; \mathtt{trans}_{\mathtt{b}}.Replying(\mathtt{b})\\
Deliver(\mathtt{b}) &\stackrel{\text{def}}{=} \overline{\mathtt{deliver}}.Reply(\mathtt{b})
\end{aligned}$$

$$Timer \stackrel{\text{def}}{=} \mathtt{time}.\overline{\mathtt{timeout}}.Timer$$

$$
\begin{array}{llll}
Trans(s\mathtt{b}) \xrightarrow{\overline{\mathtt{trans}_\mathtt{b}}} Trans(s) & \qquad & Ack(\mathtt{b}s) \xrightarrow{\overline{\mathtt{ack}_\mathtt{b}}} Ack(s) \\
Trans(s) \xrightarrow{\mathtt{send}_\mathtt{b}} Trans(\mathtt{b}s) & & Ack(s) \xrightarrow{\mathtt{reply}_\mathtt{b}} Ack(s\mathtt{b}) \\
Trans(s\mathtt{b}t) \xrightarrow{\tau} Trans(st) & & Ack(t\mathtt{b}s) \xrightarrow{\tau} Ack(ts) \\
Trans(s\mathtt{b}t) \xrightarrow{\tau} Trans(s\mathtt{bb}t) & & Ack(t\mathtt{b}s) \xrightarrow{\tau} Ack(t\mathtt{bb}s)
\end{array}
$$

Eliminujemy *Timer*

$$
Send'(\mathtt{b}) \quad \stackrel{\text{def}}{=} \quad (Send(\mathtt{b}) \mid Timer) \setminus \{\mathtt{time}, \mathtt{timeout}\}, \quad \text{itd.} \ldots
$$

$$
\begin{array}{rcl}
Send(\mathtt{b}) & \stackrel{\text{def}}{=} & \overline{\mathtt{send}_\mathtt{b}}.Sending(\mathtt{b}) \\
Sending(\mathtt{b}) & \stackrel{\text{def}}{=} & \tau.Send(\mathtt{b}) \ + \ \mathtt{ack}_\mathtt{b}.Accept(\hat{\mathtt{b}}) \ + \ \mathtt{ack}_{\hat{\mathtt{b}}}.Sending(\mathtt{b}) \\
Accept(\mathtt{b}) & \stackrel{\text{def}}{=} & \mathtt{accept}.Send(\mathtt{b})
\end{array}
$$

$$
\begin{array}{rcl}
Reply(\mathtt{b}) & \stackrel{\text{def}}{=} & \overline{\mathtt{reply}_\mathtt{b}}.Replying(\mathtt{b}) \\
Replying(\mathtt{b}) & \stackrel{\text{def}}{=} & \tau.Reply(\mathtt{b}) \ + \ \mathtt{trans}_{\hat{\mathtt{b}}}.Deliver(\hat{\mathtt{b}}) \ + \ \mathtt{trans}_\mathtt{b}.Replying(\mathtt{b}) \\
Deliver(\mathtt{b}) & \stackrel{\text{def}}{=} & \overline{\mathtt{deliver}}.Reply(\mathtt{b})
\end{array}
$$

$$
AB \quad \stackrel{\text{def}}{=} \quad Accept(\hat{\mathtt{b}}) \parallel Trans(\varepsilon) \parallel Ack(\varepsilon) \parallel Reply(\mathtt{b})
$$

Specyfikacja:

$$
\begin{array}{rcl}
Buff & \stackrel{\text{def}}{=} & \mathtt{accept}.Buff' \\
Buff' & \stackrel{\text{def}}{=} & \overline{\mathtt{deliver}}.Buff
\end{array}
$$

$$
\text{czy} \quad AB \quad \approx \quad Buff \ ?
$$

Dowód - słaba bisymulacja z dokł. do $\approx$:

stany AB:	stany $Buff$:
$Accept(\hat{\mathsf{b}}) \parallel Trans(\mathsf{b}^n) \parallel Ack(\mathsf{b}^p) \parallel \begin{matrix} Reply(\mathsf{b}) \\ Replying(\mathsf{b}) \end{matrix}$	$Buff$
$\begin{matrix} Send(\hat{\mathsf{b}}) \\ Sending(\hat{\mathsf{b}}) \end{matrix} \parallel Trans(\hat{\mathsf{b}}^m \mathsf{b}^n) \parallel Ack(\mathsf{b}^p) \parallel \begin{matrix} Reply(\mathsf{b}) \\ Replying(\mathsf{b}) \end{matrix}$	$Buff'$
$\begin{matrix} Send(\hat{\mathsf{b}}) \\ Sending(\hat{\mathsf{b}}) \end{matrix} \parallel Trans(\hat{\mathsf{b}}^m) \parallel Ack(\mathsf{b}^p) \parallel Deliver(\hat{\mathsf{b}})$	$Buff'$
$\begin{matrix} Send(\hat{\mathsf{b}}) \\ Sending(\hat{\mathsf{b}}) \end{matrix} \parallel Trans(\hat{\mathsf{b}}^m) \parallel Ack(\mathsf{b}^p \hat{\mathsf{b}}^q) \parallel \begin{matrix} Reply(\hat{\mathsf{b}}) \\ Replying(\hat{\mathsf{b}}) \end{matrix}$	$Buff$