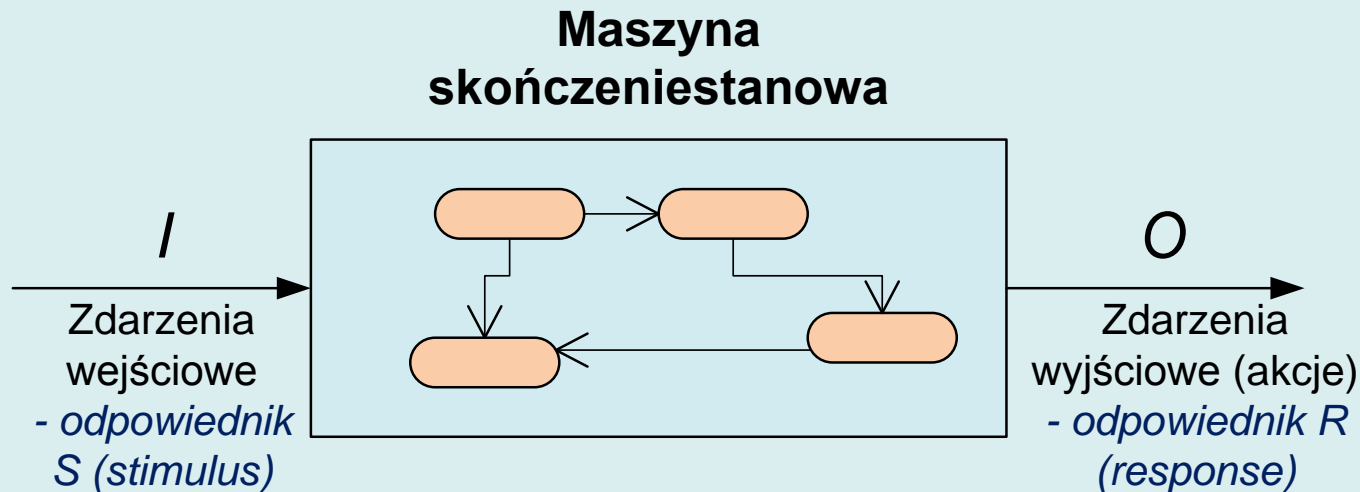


# Modelowanie ograniczeń czasowych

Popularnym modelem ograniczeń czasowych są rozszerzenia maszyn skończeniostanowych (ang. *FSM – Finite State Machine*).



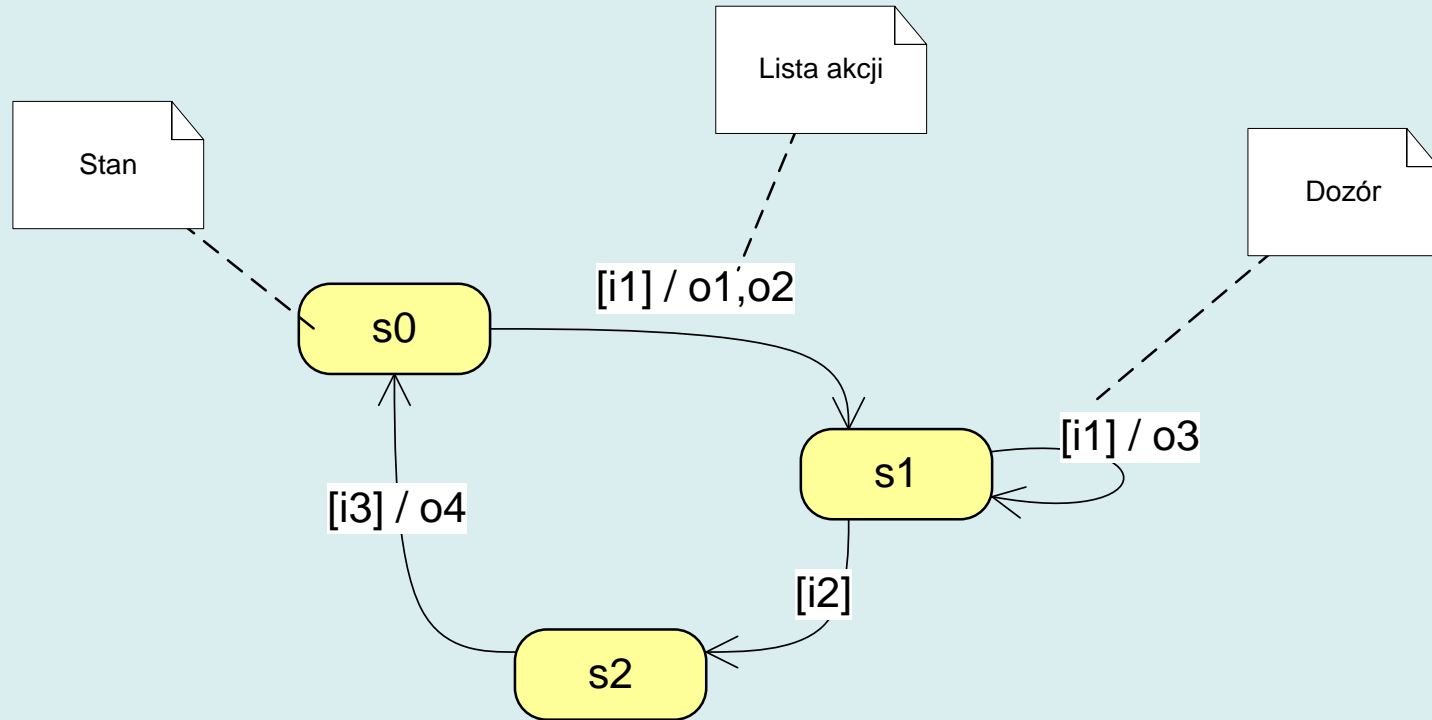
Maszyna skończeniostanowa:

- Otrzymuje na wejściu sygnały (bodźce)
- W odpowiedzi na sygnał zmienia stan
- Przy okazji zmiany stanu mogą być wykonywane akcje (pojawiają się zdarzenia wyjściowe, odpowiedzi)

# FSM: model formalny

- Maszyna skończenie stanowa jest zdefiniowana jako  $M=(S, I, O, F_S, F_O, s_0)$  , gdzie
- $S$  – jest skończonym zbiorem stanów
- $I$  – jest skończonym zbiorem zdarzeń ( dozorów, sygnałów wejściowych, warunków)
- $O$  – jest zbiorem sygnałów wyjściowych (akcji wyjściowych)
- $F_S \subset S \times I \times S$  – jest relacją przejścia pomiędzy stanami. Element relacji  $(s_k, i_m, s_{k+1})$  określa następny stan  $s_{k+1}$ , do którego nastąpi przejście, jeżeli maszyna skończeni stanowa jest w stanie  $s_k$  i otrzymała sygnał wejściowy  $i_m$ .
- $F_O \subset S \times I \times O$  – jest analogiczną relacją dla akcji wyjściowych. Element relacji  $(s_k, i_m, o_k)$  specyfikuje, że jeżeli maszyna skończeni stanowa jest w stanie  $s_k$  i otrzymała sygnał wejściowy  $i_m$  , wówczas zostanie wysłany sygnał (wykonana akcja wyjściowa  $o_k$ .
- $s_0$  – jest stanem początkowym

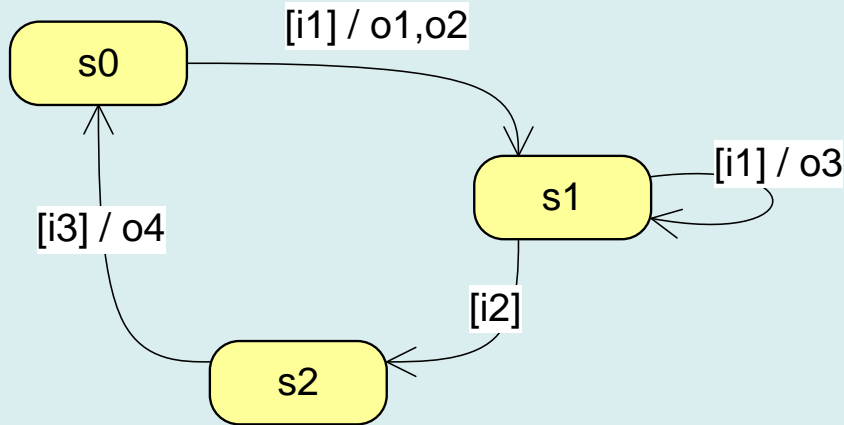
# FSM: postać graficzna



Elementami diagramu są:

- **stany**
- **przejścia** opisane:
  - Dozorem [sygnał wejściowy] (wymagane)
  - Listą akcji, sygnałów wyjściowych (opcjonalnie)

# FSM: postać tabelaryczna



$S = \{s0, s1, s2\}$

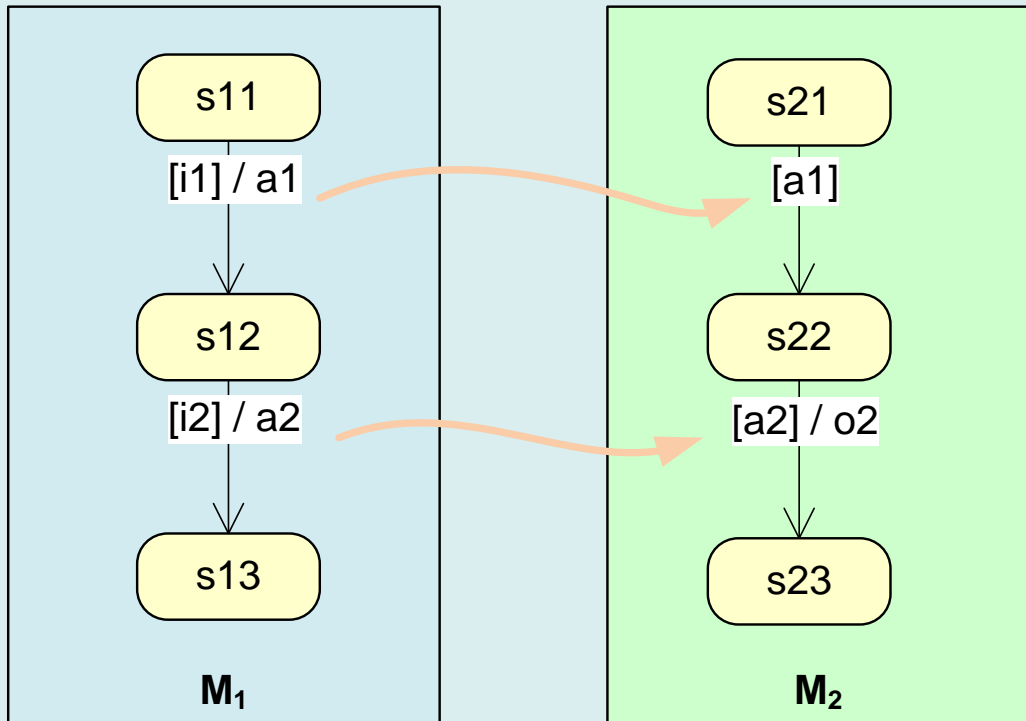
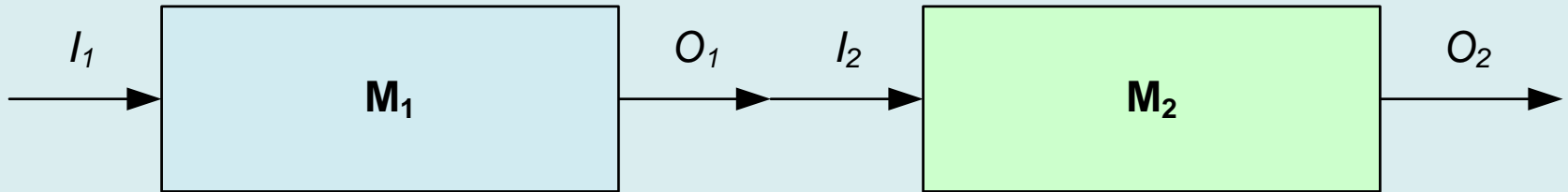
$I = \{i1, i2, i3\}$

$O = \{o1, o2, o3, o4\}$

Stan	Zdarzenie wejściowe	Następny stan	Lista akcji (zdarzeń wyjściowych)
s0	i1	s1	o1,o2
s1	i1	s1	o3
s1	i2	s2	
s2	i3	s0	o4

# Złożenie FSM

Dwie maszyny  $M_1$  i  $M_2$ , takie że  $O_1 \cap I_2 \neq \emptyset$

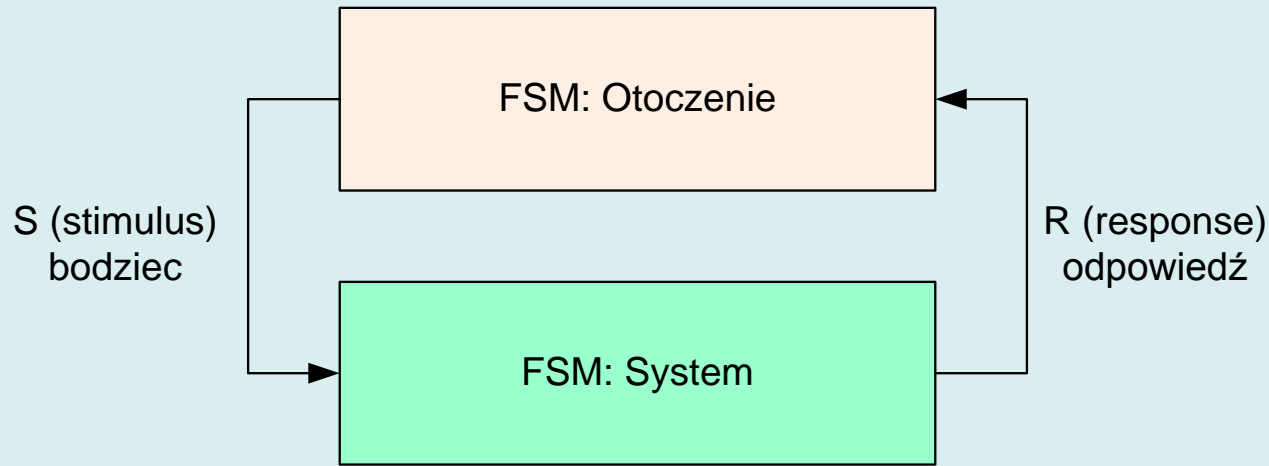


Pod wpływem zdarzeń wejściowych ze zbioru  $I_1$ , następują przejścia w  $M_1$ .

Akcje  $M_1$  są zdarzeniami wejściowymi  $M_2$ .

Przejścia  
 $s_{11} \rightarrow s_{12}$  i  $s_{21} \rightarrow s_{22}$   
 $s_{12} \rightarrow s_{13}$  i  $s_{22} \rightarrow s_{23}$   
następują synchronicznie.

# FSM: System i Otoczenie



Złożone dwóch maszyn skończeniostanowych:

- $O_{\text{Otoczenie}} = I_{\text{System}} = S$
- $O_{\text{System}} = I_{\text{Otoczenie}} = R$

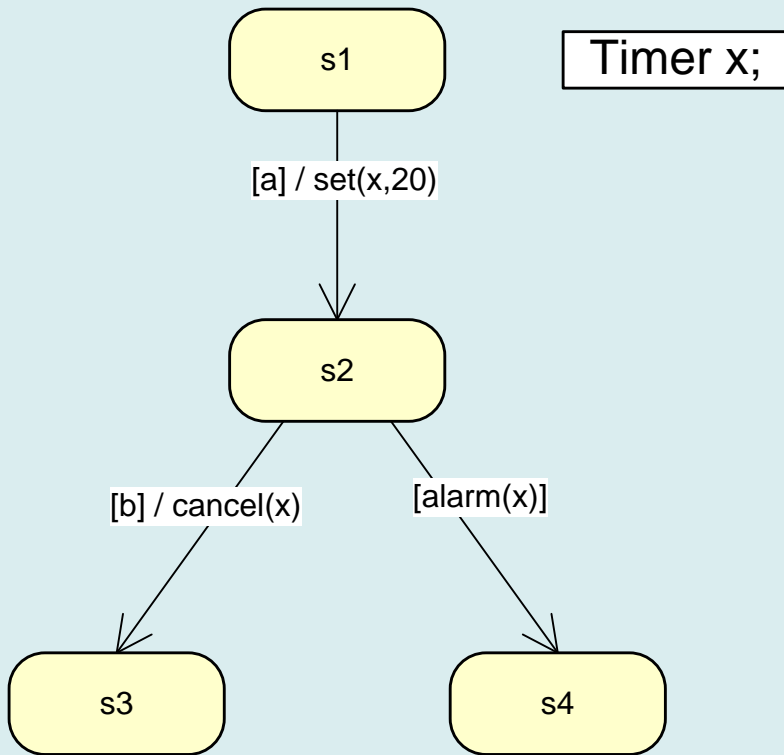
Ograniczenia czasowe

- behawioralne (SS, RS) dotyczą tego, jak ma zachowywać się otoczenie i jak system ma zareagować na zachowanie niezgodne z oczekiwanym
- Wydajnościowe (SR, RR) – jak zachowuje się system

# Rozszerzenie czasowe FSM 1

- Dodatkowym elementem modelu są budziki (zmienne typu timer)
- Dla budzika można ustawić zadany czas alarmu za pomocą operacji `set()`
- Po upływie zadanego czasu budzik wygeneruje alarm; alarm będzie dozorem przejścia w FSM.
- W niektórych wersjach można odwoływać alarmy za pomocą operacji `cancel()`. W innych przyjmuje się, że alarmy, na które nie oczekuje się są po prostu ignorowane.
- Mechanizm budzika jest powszechny w systemach operacyjnych czasu rzeczywistego (ang. *watchdog timer*)

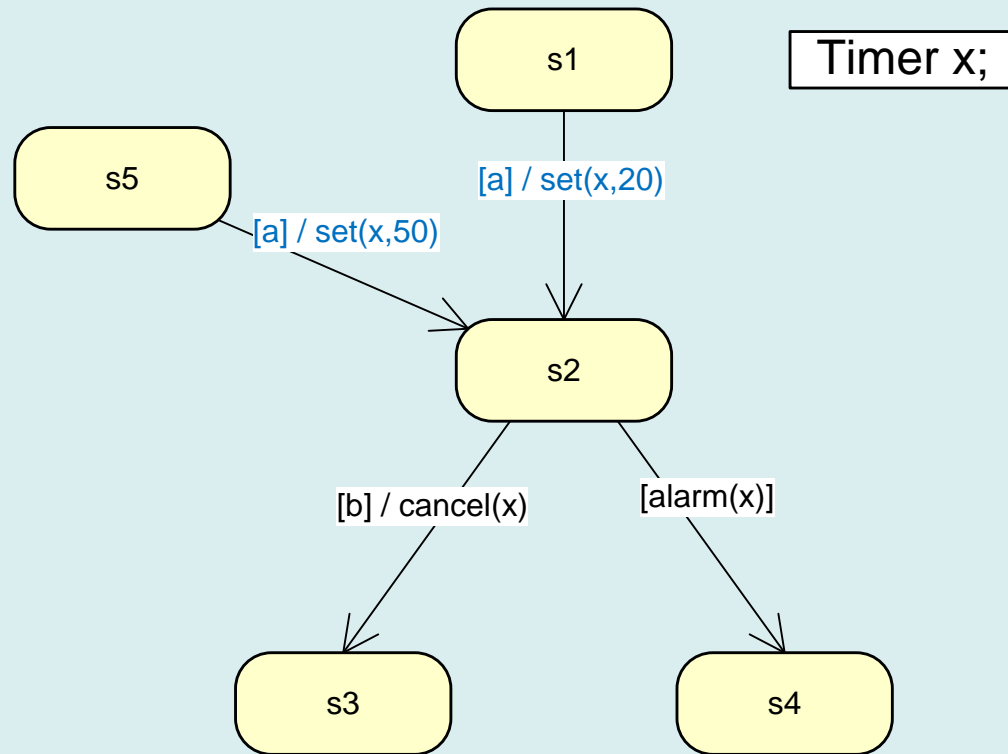
# Rozszerzenie czasowe FSM 2



- Przy przejściu ze stanu s1 do s2 budzik jest ustawiany na 20 jednostek czasu: `set(x, 20)`.
- Jeżeli czas ten upłynie i sygnał b nie pojawi się, wygenerowany zostanie sygnał `alarm(x)` i nastąpi przejście z s2 do s4
- Jeżeli sygnał b pojawi się przed alarmem, budzik zostanie wyłączony (`cancel(x)`) i nastąpi przejście do stanu s3.



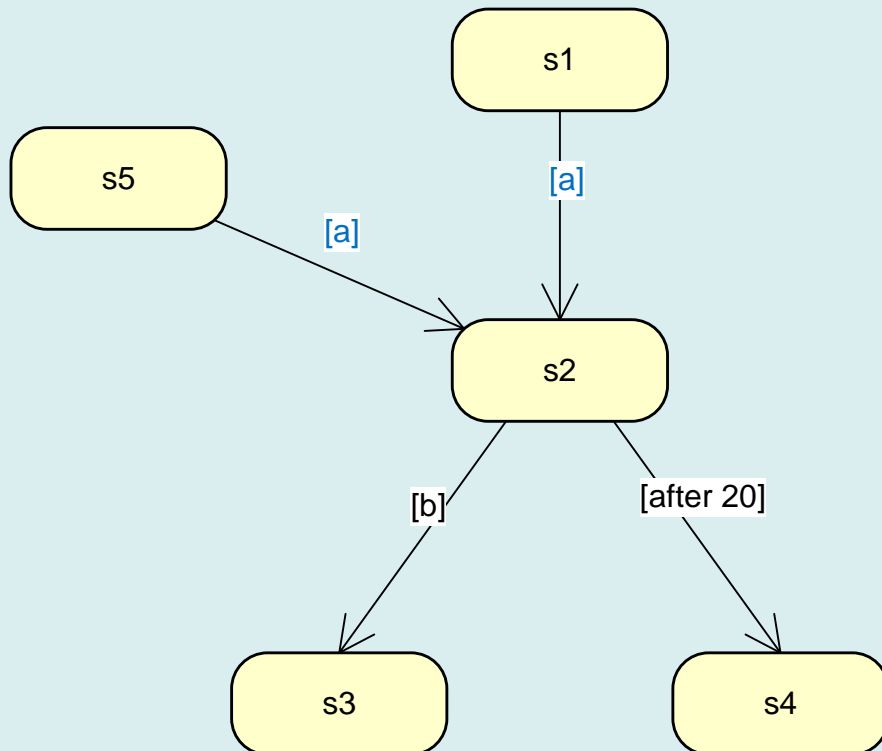
# Rozszerzenie czasowe FSM 3



- Zaletą specyfikacji jest możliwość implementacji za pomocą powszechnie stosowanych mechanizmów systemów operacyjnych czasu rzeczywistego.
- Wadą: zależność stanu budzika od ścieżki dojścia do stanu oczekiwania (s2)

# Rozszerzenie czasowe FSM 4

- Prostsza w specyfikacji jest konstrukcja UML **after t**
  - Po upływie zdanego czasu t od osiągnięcia stanu s2 nastąpi przejście do stanu s4.



- Jeżeli sygnał b pojawi się wcześniej, nastąpi przejście do stanu s3.