FSPL:Functional synchronous programming language

Michael Wittig   Marco Ziener

Freie Universität Berlin

25. Februar 2013

Zeit

Definition

Sei $\mathbb{T}$ ein Modell für die Zeit.

$$\textbf{time} : (\mathbb{T} \mapsto \mathbb{R}_+)$$

$$\textbf{step} := (\mathbb{R}_+)$$

$$\textbf{tick} : (\mathbb{T} \mapsto \mathbb{N})$$

$$\textbf{toTime} : (\mathbb{N} \mapsto \mathbb{T})$$

$$\textbf{time}\ t = \textbf{step} * (\textbf{tick}\ t)$$

$$\textbf{toTime}(\textbf{tick}\ t) = t$$

$$\textbf{tick}(\textbf{toTime}\ n) = n$$

Definition

Sei $t_1, t_2 \in \mathbb{T}$ und $n \in \mathbb{N}$:

$$t_1 + t_2 := \textbf{toTime}(\textbf{tick}\ t_1 + \textbf{tick}\ t_2)$$

$$t_1 - t_2 := \textbf{toTime}(max(0, \textbf{tick}\ t_1 - \textbf{tick}\ t_2))$$

$$n * t_1 := \textbf{toTime}(n * \textbf{tick}\ t)$$

$$t\ \ n := \textbf{toTime}((\textbf{tick}\ t)\ div\ n)$$

Vergleich von Prozessen

Definition

$$p_1 = p_2 \Leftrightarrow \mathbf{state}_{\mathbb{Z}}\ p_1\ t = \mathbf{state}_{\mathbb{Z}}\ p_2\ t$$

$$p_1 \overset{t_0}{|=} p_2 \Leftrightarrow \exists t_o \in \mathbb{T} : (t \geq t_0 \Rightarrow \mathbf{state}_{\mathbb{Z}}\ p_1\ t = \mathbf{state}_{\mathbb{Z}}\ p_2\ t)$$

$$p_1 \overset{t_0}{=|} p_2 \Leftrightarrow \exists t_o \in \mathbb{T} : (t < t_0 \Rightarrow \mathbf{state}_{\mathbb{Z}}\ p_1\ t = \mathbf{state}_{\mathbb{Z}}\ p_2\ t)$$

$$p_1 \overset{t_0}{|}= \overset{t_1}{|} p_2 \Leftrightarrow \exists t_o, t_1 \in \mathbb{T} : (t_0 \leq t < t_1 \Rightarrow \mathbf{state}_{\mathbb{Z}}\ p_1\ t = \mathbf{state}_{\mathbb{Z}}\ p_2\ t)$$

Ereignisse

Definition

Das Auftreten eines Ereignisses nach einem gegebenen Zeitpunkt t_0 wird beschrieben durch:

$$\textbf{happened} : \mathbb{E} \mapsto \mathbb{T} \mapsto \mathbb{T} \mapsto \mathbb{B}$$

$$t \leq t_0 \Rightarrow \neg(\textbf{happened}\ e\ t_0\ t)$$

$$(\textbf{happened}\ e\ t_0\ t) \wedge (t \leq t') \Rightarrow \textbf{happened}\ e\ t_0\ t'$$

Vergleich von Ereignissen

Definition

$$\forall e_1, e_2 \in \mathbb{E},\ \forall t_0, t \in \mathbb{T},\ \textbf{never} \in \mathbb{E}:$$

$$e_1 = e_2 \Leftrightarrow \textbf{happened}\ e_1\ t_0\ t = \textbf{happened}\ e_2\ t_0\ t$$

$$e_1 \leq e_2 \Leftrightarrow \textbf{happened}\ e_2\ t_0\ t \Rightarrow \textbf{happened}\ e_1\ t_0\ t$$

$$\textbf{happened never}\ t\ t_0 = \mathit{false}$$

Timeout und Takt

Definition (Timeout)

Für eine Funktion **after** : $\mathbb{T} \to \mathbb{E}$:

$$\textbf{happened}\ (\textbf{after}\ \Delta t)\ t_0\ t = t \geq (t_0 + \Delta t)$$

Definition (Takt)

Ein Takt ist definiert durch eine Periodendauer ΔT und eine Phasenverschiebung t_1.
Für eine Funktion **clock** : $\mathbb{T} \to \mathbb{T}_+ \to \mathbb{E}$:

$$\textbf{happened}\ (\textbf{clock}\ t_1\ \Delta t) t_0\ t = \begin{cases} t \geq t_1\ , & \text{falls } t_1 > t \\ \textbf{happened}(\textbf{clock}(t_1 + \Delta t)\ \Delta t) t_0\ t\ , & \text{sonst} \end{cases}$$

(**clock** t_1 Δt) wirft nach $t_1 + \Delta t$ ein Event.

Mehr Ereignisse

Definition (Trigger)

Die trigger-Funktion erzeugt ein Event, wenn der Zustand eines booleschen Prozesses von false auf true wechselt [1]. Für eine Funktion **trigger** : $\mathbb{P} \to \mathbb{E}$:

$$\textbf{happened}\ (\textbf{trigger}\ p) t_0\ t = \begin{cases} \mathit{false}, & \text{falls}\ \ t \leq t_0 \\ \mathit{true}, & \text{falls}\ \ \textbf{happened}\ (\textbf{trigger}\ p)\ t_0\ (t-1) \\ (\textbf{state}_{\mathbb{B}}\ p\ t) \wedge \neg(\textbf{state}_{\mathbb{B}}\ p\ (t-1)), & \text{sonst} \end{cases}$$

[1] eine steigende Flanke

Sequenz von Ereignissen

Definition (Ereignissequenz)

$e_0 + e_1$ bezeichnet das Ereignis e_1, das auf das Ereignis e_0 folgt. Für eine Funktion $+ : \mathbb{E} \times \mathbb{E} \rightarrow \mathbb{E}$ in infix-Notation gilt:

$$\textbf{happened } (e_0 + e_1) t_0 \ t = \begin{cases} \mathit{false}, & \text{falls } \neg(\textbf{happened } e_0 \ t_0 \ t) \\ \textbf{happened } e_1 \ t_1{}^2 \ t, & \text{sonst} \end{cases}$$

[2] $t_1 = \mathit{min}\{t \in \mathbb{T} | \textbf{happened } e_0 \ t_0 \ t\}$

Ereignisausschluss und bewachte Ereignisse

Definition (Ereignisausschluss)

Für eine Funktion $-$: $\mathbb{E} \mapsto \mathbb{E} \mapsto \mathbb{E}$ in Infix-Notation gilt:

$$\textbf{happened}\ (e_0 - e_1)\ t_0\ t \Leftrightarrow$$

$$\textbf{happened}\ e_0\ t_0\ t \wedge \neg(\textbf{happened}\ e_1\ t_0\ t)$$

Definition (bewachtes Ereignis)

Für eine Funktion **guard** : $\mathbb{P} \mapsto \mathbb{E} \mapsto \mathbb{E}$ gilt:

$$\textbf{happened}\ (\textbf{guard}\ p\ e)\ t_0\ t = \begin{cases} \mathit{false}, & \text{falls}\ \ \neg(\textbf{happened}\ e_0\ t_0\ t) \\ \mathit{true}, & \text{falls}\ \ \textbf{state}_{\mathbb{B}}\ p\ t_1 = \mathit{true} \\ \textbf{happened}\ (\textbf{guard}\ p\ e)\ t_1{}^{3}\ t, & \text{sonst} \end{cases}$$

[3] wobei $t_1 = \mathit{min}\{t \in \mathbb{T} | \textbf{happened}\ e\ t_0\ t\}$

Definition (bedingtes Ereignis)

Für eine Funktion **ifElseEvent** : $\mathbb{P} \mapsto \mathbb{E} \mapsto \mathbb{E} \mapsto \mathbb{E}$ gilt:

$$\textbf{happened}\ (\textbf{ifElseEvent}\ p\ e_0\ e_1)\ t_0 = \begin{cases} \textbf{happened}\ e_0\ \ t_0, \text{falls}\ \ \textbf{state}\ p\ \ t_0 = tr \\ \textbf{happened}\ e_1\ \ t_0,\ \ \text{sonst} \end{cases}$$

Definition (Watchdog-Timeout)

Für eine Funktion **watchdog** : $\mathbb{E} \mapsto \mathbb{T}_{+} \mapsto \mathbb{E}$ gilt:

$$\textbf{happened}\ (\textbf{watchdog}\ e\ \Delta t)\ t_0\ t = \begin{cases} \mathit{false},\ \ \text{falls}\ \ t < (t_0 + \Delta t) \\ \mathit{true},\ \ \text{falls}\ \ \neg(\textbf{happened}\ e\ \ t_0\ \ (t_0 + \Delta t \\ \textbf{happened}\ (\textbf{watchdog}\ e\ \ \Delta t)\ \ t_1\ \ t,\ \ \text{son} \end{cases}$$

$$, \mathit{wobei}\ \ t_1 = \mathit{min}\{t \in \mathbb{T} \mid \textbf{happened}\ e\ \ t_0\ \ t\}$$

Signalflusssysteme

Definition (Konstante)

Für eine Funktion $\mathbf{const}_X : X \mapsto \mathbb{P}_X$ zu jeder Zustandsmenge X gilt:

$$\mathbf{state}_X\ (\mathbf{const}_X k)t = k$$

Definition (Zeit als Signal)

Für ein Objekt $\mathbf{timeProcess} : \mathbb{P}_\mathbb{T}$ gilt:

$$\mathbf{state}_\mathbb{T}\ \mathbf{timeProcess}\ t = t$$

$$,\ \ mit\ \mathbf{timeProcess} = id_\mathbb{T}$$

Statische Systeme

Definition (Statisches System)

Ein statisches System hat daher die Systemfunktion

$$f : \mathbb{P}_X \mapsto \mathbb{P}_Y$$

und ist charakterisiert durch eine Funktion

$$f' : X \mapsto Y$$

$$, mit\ \mathbf{state}_Y(f\ \ p)\ \ t = f'(\mathbf{state}_X\ \ p\ \ t),\ \forall p \in \mathbb{P}_X\ und\ \forall t \in \mathbb{T}$$

Definition (Lifting)

Für eine Funktion $\mathbf{apply}_{X \mapsto Y} : (X \mapsto Y) \mapsto \mathbb{P}_X \mapsto \mathbb{P}_Y$ gilt:

$$\mathbf{state}_Y(\mathbf{apply}_{X \mapsto Y}\ f\ \ p)\ \ t = f\ \ (\mathbf{state}_X\ \ p\ \ t)$$

Dynamische Systeme

Definition (Zeitverschiebung)

Für eine Funktion $\mathbf{delay}_X : X \mapsto \mathbb{P}_X \mapsto \mathbb{P}_X$ gilt:

$$\mathbf{state}_X(\mathbf{delay}_X\ x_0\ p)0 = x_0$$

$$\mathbf{state}_X(\mathbf{delay}_X\ x_0\ p)(t+1) = \mathbf{state}_X\ p\ t$$

Reaktive Prozesse

Definition (Phase)

$$\mathbf{phase}_X : \mathbb{P}_X \mapsto \mathit{Phase}_X$$

$$\mathbf{valueInPhase}_{Y,X} : \mathbb{P}_Y \mapsto (Y \mapsto \mathit{Phase}_X) \mapsto \mathit{Phase}_X$$

$$\mathbf{switch}_X : \mathit{Phase}_X \mapsto \mathit{Transition}_X \mapsto \mathit{Phase}_X$$

$$\mathbf{start}_X : \mathit{Phase}_X \mapsto \mathit{Time} \mapsto \mathbb{P}_X$$

$$\mathbf{goto}_X : \mathit{Phase}_X \mapsto \mathit{Continuation}_X$$

$$\mathbf{wait}_X \in \mathit{Continuation}_X$$

$$\mathbf{when}_X : \mathbb{E} \mapsto \mathit{Continuation}_X \mapsto \mathit{Transition}_X$$

$$\mathbf{start}_X\ (\mathbf{phase}_X\ p)\ t_0\ \Big|\overset{t_0}{=} p$$

Definition

$$\textbf{start}_X\ (\textbf{valueInPhase}_{Y,X}\ p\ \varphi_p)\ t_0 \mid \overset{t_0}{=} \textbf{start}_X(\varphi_p(\textbf{state}_Y\ p\ t_0))\ t_0$$

$$t \geq t_0 \Rightarrow \textbf{state}_X(\textbf{start}_X(\textbf{switch}_X\ \varphi\ [\textbf{when}_X\ e_0\ \psi_0, \ldots, \textbf{when}_X\ e_n\ \psi_n])\ t_0)\ t =$$

$$\begin{cases} \textbf{state}_X(\textbf{start}_X\ \varphi\ t_0)t & , \mathit{falls}\ \neg(\textbf{happened}(e_0 \vee \cdots \vee e_n)t_0\ t) \\ \textbf{state}_X(\textbf{start}_X\ \varphi_k\ t_1)\ t & , \mathit{sonst} \end{cases}$$

, wobei

$$t_1 = min\{t \in \mathbb{T} \mid \textbf{happened}\ (e_0 \vee \cdots \vee e_n)t_0\ t\}$$

$$x_1 = \textbf{state}_X(\textbf{start}_X\ \varphi\ t_0)\ t_1$$

$$k = min\{i \in 0 \ldots n \mid \textbf{happened}\ e_i\ t_0\ t_1\}$$

$$\varphi_i = \begin{cases} \phi_i & , \mathit{falls}\ \psi_i = \textbf{goto}_X\ \phi_i \\ \textbf{phase}_X(\textbf{const}_X\ x_1)\ t & , \mathit{falls}\ \psi_i = \textbf{wait}_X \end{cases}$$

Mehr Phasen

Definition ($\textbf{ifElsePhase}_X$)

Für eine Funktion $\textbf{ifElsePhase}_X : \mathbb{P}_{\mathbb{B}} \mapsto \mathit{Phase}_X \mapsto \mathit{Phase}_X$ und für jede Zustandsmange X gilt:

$$\textbf{ifElsePhase}_X\ c\ \varphi_0\ \varphi_1 =$$

$$\textbf{valueInPhase}_X\ c\ \left(\lambda c_0 \in \mathbb{B}. \begin{cases} \varphi_0 & , \mathit{falls}\ c_0 = \mathit{true} \\ \varphi_1 & , \mathit{sonst} \end{cases}\right)$$

Definition (bedingter Phasenwechsel)

$$\textbf{switch}_X\ \varphi[\textbf{when}_X\ e\ (\textbf{goto}_X(\textbf{ifElsePhase}_X\ c\ \varphi_0\ \varphi_1))]$$

Nebenläufigkeit

Definition (Phasenparallelisierung)

Sei $\lambda i \in I.X_i$ eine Mengenabbildung die jedem Index i aus einer Indexmenge I eine Zustandsmenge X_i zuordnet und

$$\mathbf{orthogonalize}_{\Pi(\lambda i \in I.X_i)} : \Pi(\lambda i \in I.Phase_{X_i}) \mapsto Phase_{\Pi(\lambda i \in I.X_i)}$$

$$\mathbf{start}_X(\mathbf{orthogonalize}_{\Pi(\lambda i \in I.X_i)}(\lambda i \in I.\varphi_i))\ t_0 \mid \stackrel{t_0}{=}$$

$$\mathbf{zip}_{\Pi(\lambda i \in I.X_i)}(\lambda i \in I.\mathbf{start}_X\ \varphi_i\ t_0)$$

Lokalität

Definition (lokale Variablen)

Für eine Funktion
$\mathbf{variableInPhase}_{Y,X} : Phase_Y \mapsto (\mathbb{P}_Y \mapsto Phase_X) Phase_X$ gilt:

$$\mathbf{start}_X(\mathbf{variableInPhase}_{Y,X}\ p\ \varphi_p)\ t_0 \mid \stackrel{t_0}{=} \mathbf{start}_X\ (\varphi_p\ (start_Y\ p\ t_0))t_0$$

Definition (Ereignislokale Auswertung)

Für eine Funktion $\mathbf{valueInEvent}_X : \mathbb{P}_X \mapsto (X \mapsto \mathbb{E}) \mapsto \mathbb{E}$ und $Phase_X \mapsto (\mathbb{P}_X \mapsto \mathbb{E}) \mapsto \mathbb{E}$ gilt:

$$\mathbf{happened}\ (\mathbf{valueInEvent}_X\ p\ e_p)t_0\ t = \mathbf{happened}(e_p(\mathbf{state}_X p\ t_0))\ t_0\ t$$

$$\mathbf{happened}\ (\mathbf{variableInEvent}_X\ \varphi\ e_\varphi)t_0\ t = \mathbf{happened}(e_\varphi(\mathbf{start}_X \varphi\ t_0))\ t_0\ t$$

Sequentielle Prozesse

Definition

$$\textbf{behaviour}_X : \mathbb{A}_X \mapsto \mathit{Phase}_X$$

$$\textbf{termination}_X : \mathbb{A}_X \mapsto \mathbb{E}$$

$$\textbf{until}_X : \mathbb{E} \mapsto \mathit{Phase}_X \mapsto \mathbb{A}_X$$

$$\textbf{continue}_X : \mathbb{A}_X \mapsto \mathit{Continuation}_X \mapsto \mathit{Phase}_X$$

$$\textbf{loop}_X : \mathbb{A}_X \mapsto \mathit{Phase}_X$$

$$\textbf{sequence}_X : \mathbb{A}_X \mapsto \mathbb{A}_X \mapsto \mathbb{A}_X$$

$$\textbf{ifElse}_X : \mathbb{P}_\mathbb{B} \mapsto \mathbb{A}_X \mapsto \mathbb{A}_X \mapsto \mathbb{A}_X$$

$$\textbf{repeatUntil}_X : \mathbb{A}_X \mapsto \mathbb{P}_\mathbb{B} \mapsto \mathbb{A}_X$$

$$\textbf{valueInAction}_X : \mathbb{P}_Y \mapsto (Y \mapsto \mathbb{A}_X) \mapsto \mathbb{A}_X$$

$$\textbf{variableInAction} : \mathit{Phase}_Y \mapsto (\mathbb{P}_Y \mapsto \mathbb{A}_X) \mapsto \mathbb{A}_X$$

Mehr Sequentielle Prozesse

Definition

$$\mathbf{behaviour}_X(\mathbf{until}_X\ e\ \varphi) = \varphi$$

$$\mathbf{termination}_X(\mathbf{until}_X\ e\ \varphi) = e$$

$$\mathbf{continue}_X\ a\ \psi = \mathbf{switch}_X(\mathbf{behaviour}_X a)[\mathbf{when}_X(\mathbf{termination}_X\ a)\psi]$$

$$\mathbf{loop}_X\ a = \mathbf{continue}_X\ a(\mathbf{goto}_X(\mathbf{loop}_X\ a))$$

$$\mathbf{behaviour}_X(\mathbf{sequence}_X\ a\ b) = (\mathbf{termination}_X\ a) + (\mathbf{termination}_X\ b)$$

$$\mathbf{behaviour}_X(\mathbf{ifElse}_X\ c\ a\ b) = \mathbf{ifElsePhase}_X\ c\ (\mathbf{behaviour}_X\ a)(\mathbf{behaviour}_X\ b$$

$$\mathbf{termination}_X(\mathbf{ifElse}_X\ c\ a\ b) = \mathbf{ifElseEvent}\ c(\mathbf{termination}_X\ a)(\mathbf{termination}$$

$$\mathbf{behaviour}_X(\mathbf{repeatUntil}_X\ a\ c) = \mathbf{loop}_X\ a$$

$$\mathbf{termination}_X(\mathbf{repeatUntil}_X\ a\ c) = \mathbf{guard}\ c(\mathbf{termination}_X\ a)$$

Noch mehr Sequentielle Prozesse

Definition

$$\mathbf{start}_X(\mathbf{behaviour}_X(\mathbf{valueInAction}_{Y,X}\ y\ a_y))\ t_0 \mid \overset{t_0}{=}$$

$$\mathbf{start}_X(\mathbf{behaviour}_X(a_y(\mathbf{state}_Y\ y\ t_0)))t_0$$

$$\mathbf{happened}_X(\mathbf{termination}_X(\mathbf{valueInAction}_{Y,X}\ y\ a_y))\ t_0\ t =$$

$$\mathbf{happened}_X(\mathbf{termination}_X(a_y(\mathbf{state}_Y\ y\ t_0)))t_0\ t$$

$$\mathbf{start}_X(\mathbf{behaviour}_X(\mathbf{variableInAction}_{Y,X}\ \varphi\ a_\varphi))\ t_0 \mid \overset{t_0}{=}$$

$$\mathbf{start}_X(\mathbf{behaviour}_X(a_\varphi(start_Y\ \varphi y\ t_0)))t_0$$

$$\mathbf{happened}(\mathbf{termination}_X(\mathbf{variableInAction}_{Y,X}\ \varphi\ a_\varphi))\ t_0\ t =$$

$$\mathbf{happened}(\mathbf{termination}_X(a_\varphi(start_Y\ \varphi\ t_0)))t_0\ t$$

Ein paar sequentielle Details

Definition (Vergleich von Aktionen)

$$a_1 = a_2 \Leftrightarrow (\textbf{behaviour}_X\ a_1) = (\textbf{behaviour}_X\ a_2)$$

Definition (Aktionswiederholung)

$$1 * a = a$$

$$(n+1) * a = \textbf{sequence}\ (n * a)\ a$$

Definition (Ausnahmebehandlung)

Für die Funktion $\textbf{continueWithExceptions}_X : \mathbb{A}_X \mapsto \mathit{Continuation}_X \mapsto \mathit{Transition}^*_X \mapsto \mathit{Phase}_X$ gilt:

$$\textbf{continueWithExceptions}_X\ a\ \psi[\tau_0, \ldots, \tau_{n-1}] =$$

$$\textbf{switch}_X(\textbf{behaviour}_X\ a)[\textbf{when}_X(\textbf{termination}_X\ a)\psi, \tau_0, \ldots, \tau_{n-1}]$$