Temporale Spezifikation objekt-orientierter Systeme

Jens Brandt

Juli 2003

Inhaltsübersicht

- Daten-/Klassensignaturen
- Ereignisstrukturen
- Temporallogische Formeln
- Beispiel Cocopeli
- Zusammenfassung

Datensignaturen

- $\Sigma_D = (S_D, \mathbf{\Omega}_D)$ heißt Datensignatur:
  - S_D (Datensorten)
  - $\mathbf{\Omega_D} = \{\Omega_{x,s}\}_{x \in S_D^*, s \in S_D}$ (Operationssymbole)
- Terme:
  - $\mathbf{X} = \{X_s\}_{s \in S_D}$ (Variablensymbole)
  - $\mathbf{T}_{\Sigma_D}(\mathbf{X})$ (Terme über $\mathbf{X}$)

Interpretation einer Datensignatur

- Σ_D-Algebra $\mathbf{U}$
  - s_U (Trägermengen für jede Datensorte $s \in S_D$)
  - $\omega_U : x_U \to s_U$
    (Operationen für jedes $\omega : x \to s \in \mathbf{\Omega}$)
- Interpretation eines Terms:
  - t_U^θ (Interpretation in $\mathbf{U}$ mit Belegung θ)

Klassensignaturen

- $\Sigma_C = (S_O, I, A)$ heißt Klassensignatur über der Datensignatur Σ_D:
  - S_O (Objektsorten)
  - $I = \{I_{x,b}\}_{x \in S_{DO}^*, b \in S_O}$ (Instanzoperatoren)
  - $A = \{A_{x,b}\}_{x \in S_{DO}^*, b \in S_O}$ (Aktionsoperatoren)

Klassensignaturen: Beispiel

class User

uses nat, string, boolean;
attributes mtknr:nat, name,login,pwd:string;
actions new(nat,string,string), delete,
setPassword(string);

axioms *var* m:nat; n,l,p:string;

$\odot$ new(m,n,l,p)
$\Rightarrow (\neg \triangleright \text{new} \wedge \triangleright \text{setPassword} \wedge \triangleright \text{delete})\ U^{\circ}\ \odot \text{delete}$
$\wedge$ (mtknr=m $\wedge$ name=n $\wedge$ login=l $\wedge$ pwd=p),
$\odot$ delete
$\Rightarrow \neg \triangleright \text{new} \wedge \neg \triangleright \text{setPassword} \wedge \neg \triangleright \text{delete}$,
$\odot$ setPassword(p)
$\Rightarrow \text{pwd=p}\ \ W^{\circ}\ \odot \text{setPassword}$

Instanzsignatur

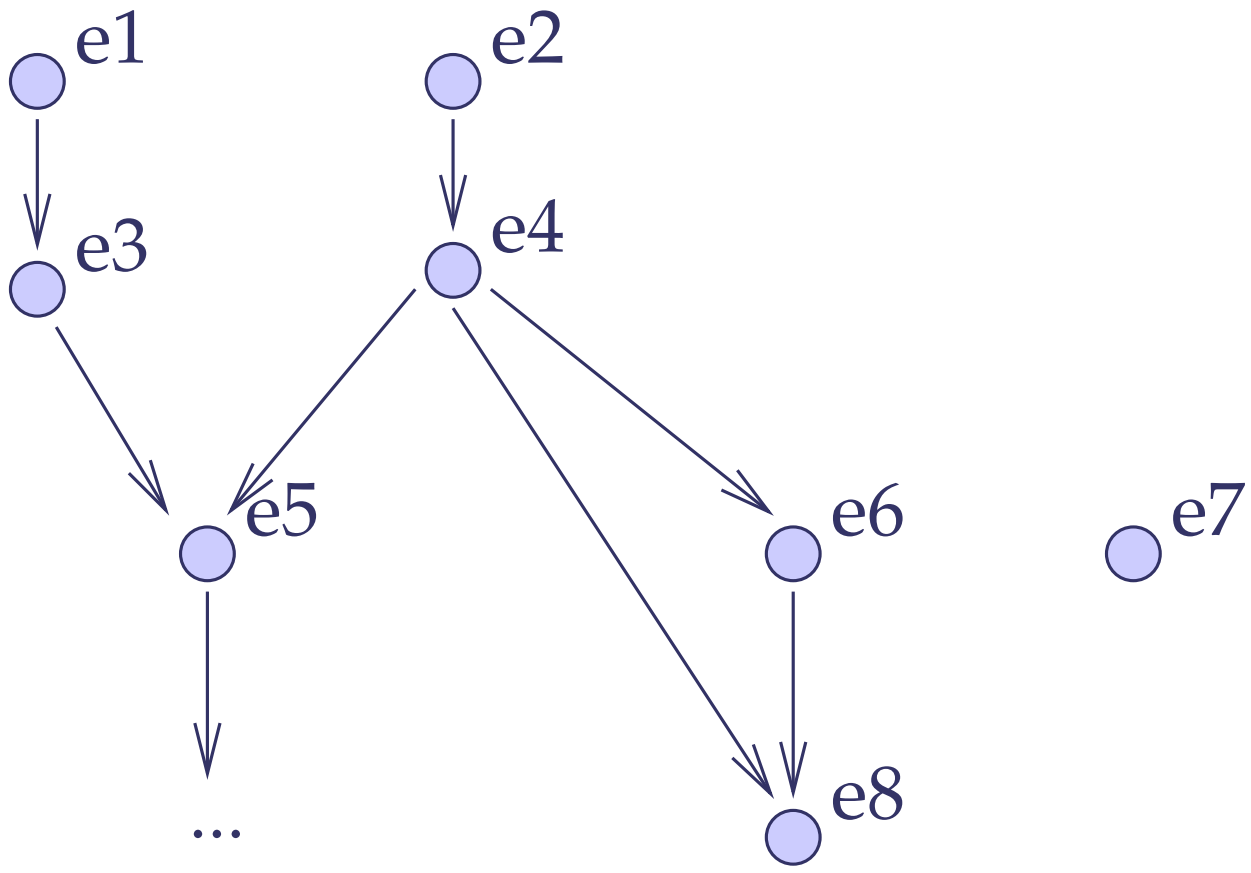
$\Sigma_I = (Id, \mathbf{Ac})$ heißt Instanzsignatur:

- Id (Identitäten)
- $\mathbf{Ac} = \{Ac_i\}_{i \in Id}$ (Aktionsalphabete)

Ereignisstrukturen

- Tripel $E = (Ev, \rightarrow^*, \#)$ heißt Ereignisstruktur:
  - Ev (Ereignisse)
  - $\rightarrow^* \subseteq Ev \times Ev$ (Kausalität: partielle Ordung auf Ev)
  - $\# \subseteq Ev \times Ev$ (Konflikt: symm. und irreflex. Relation)
- Zwei Ereignisse sind
  - kausal abhängig $e_1 \sim e_2 \Leftrightarrow (e_1 \rightarrow^* e_2 \vee e_2 \rightarrow^* e_1)$
  - nebenläufig $e_1 co e_2 \Leftrightarrow \neg(e1 \sim e2 \vee e1 \# e2)$

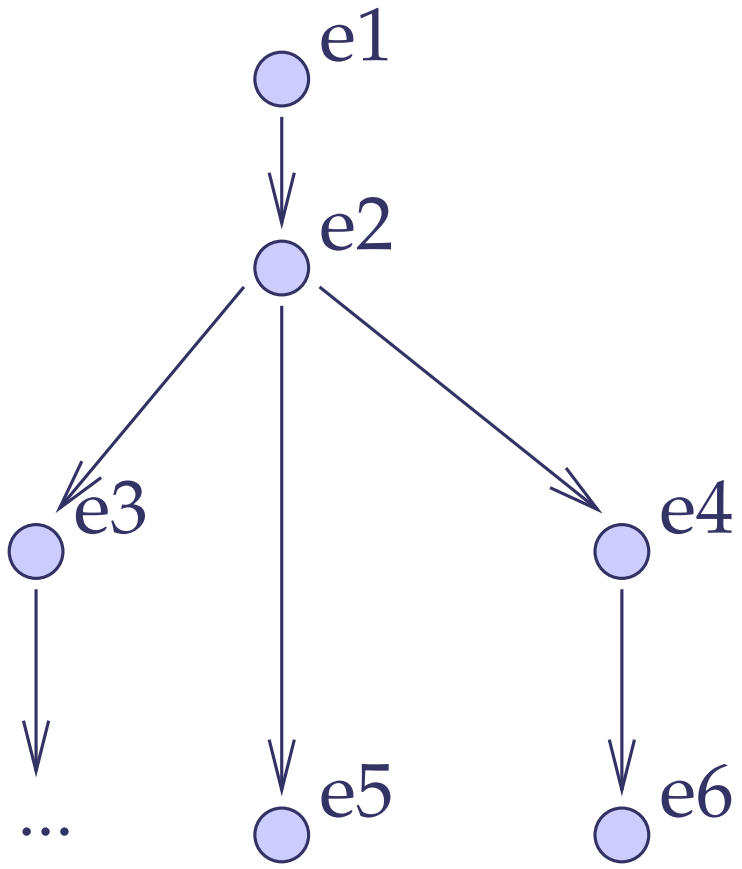
Ereignisstrukturen: Beispiel



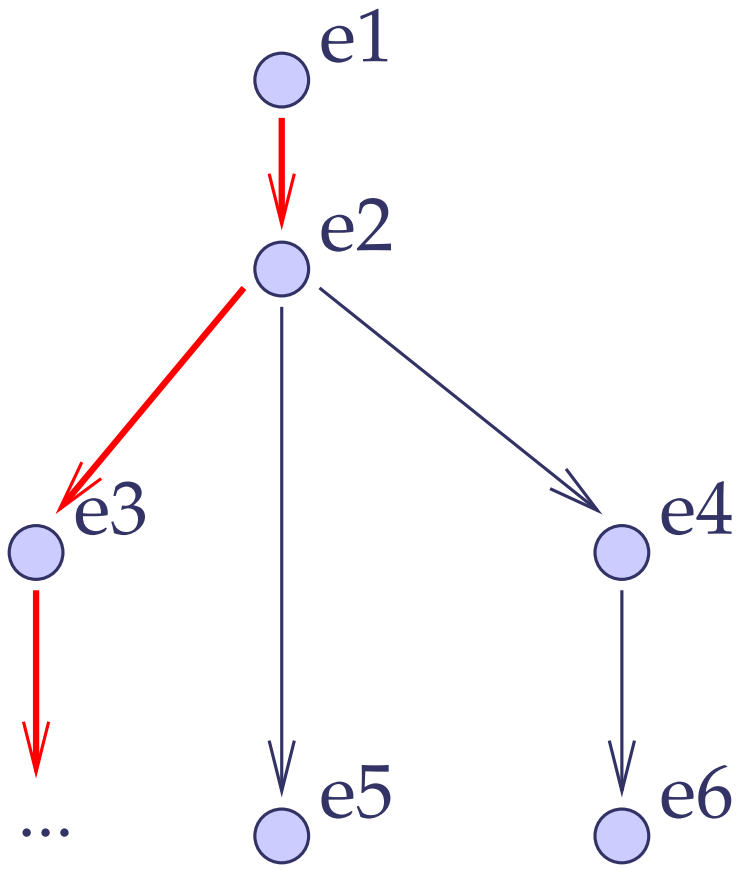
Ereignisstrukturen: Weitere Definitionen

- Lokale Konfiguration: $\downarrow e = \{e' | e' \rightarrow^* e\}$.
- Lebenszyklus: maximale Konfiguration in E
- Sequentielle Ereignisstruktur:
  - eindeutiges, minimales Element $\epsilon \in Ev$
  - lokale Konfiguration $\downarrow e$ vollständig geordnet
    $\Rightarrow e \# f \Leftrightarrow \neg(e \, co \, f)$

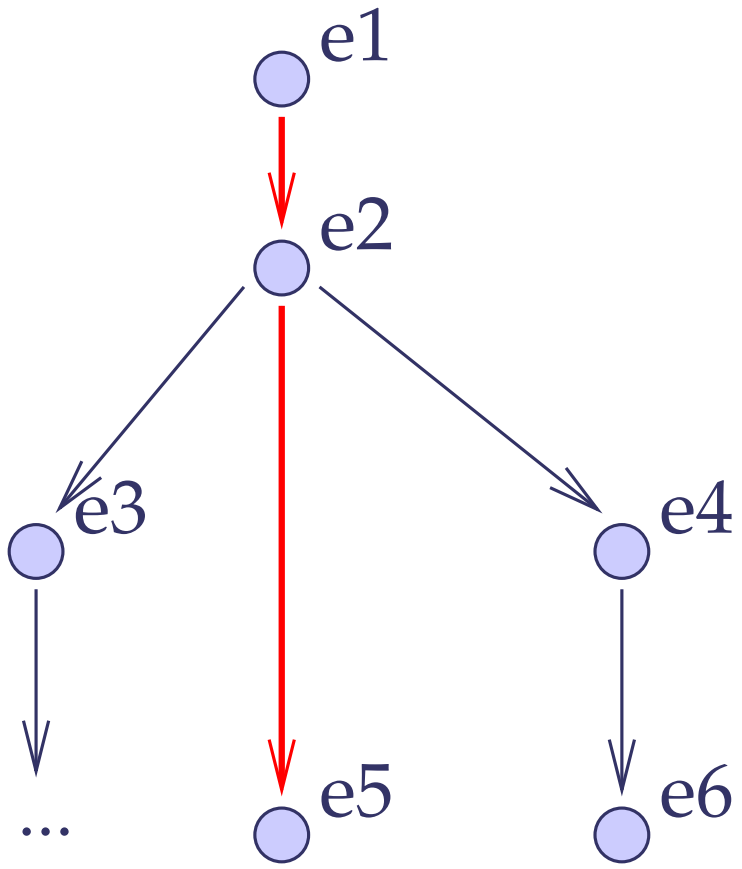
Sequentielle Ereignisstrukturen: Beispiel



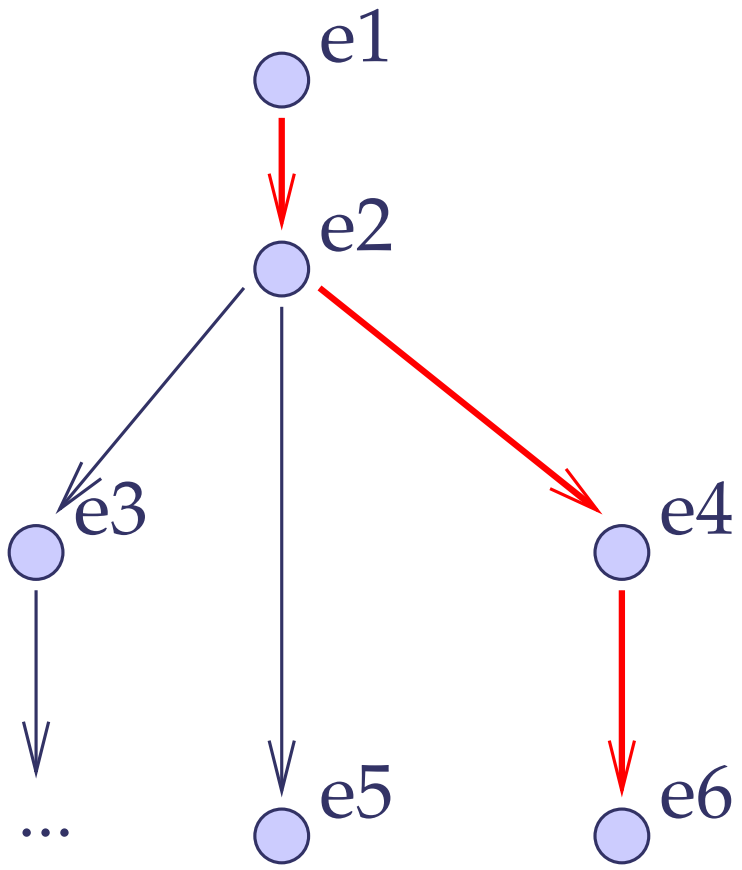
Sequentielle Ereignisstrukturen: Beispiel



Sequentielle Ereignisstrukturen: Beispiel



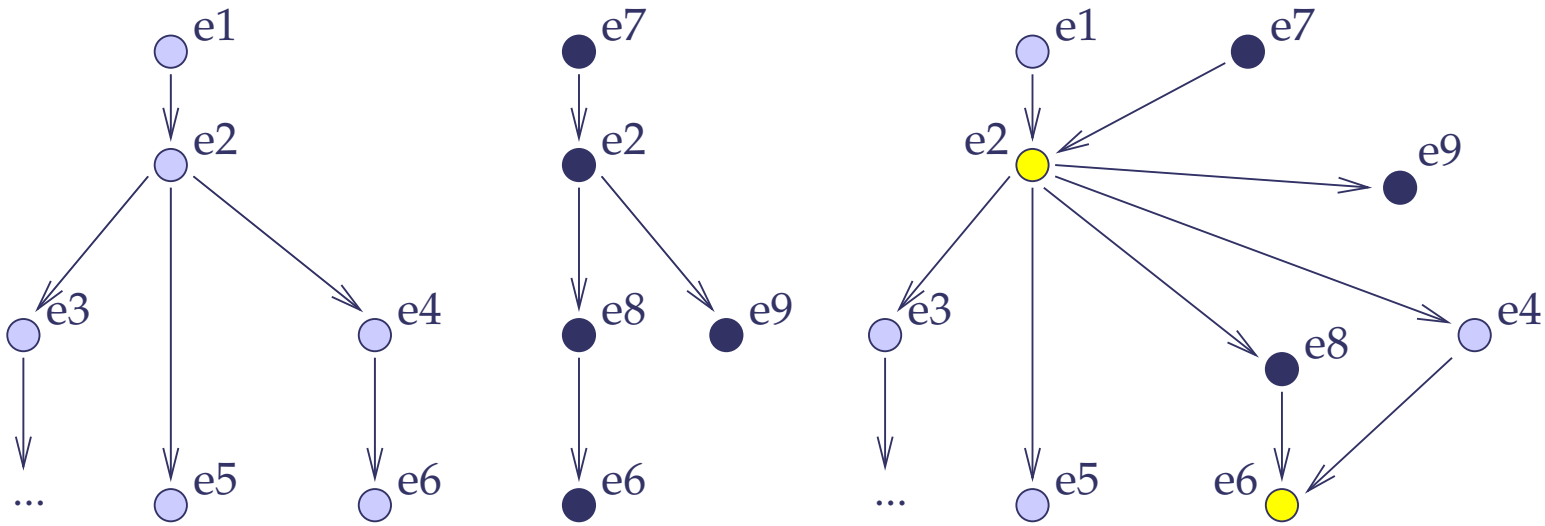
Sequentielle Ereignisstrukturen: Beispiel



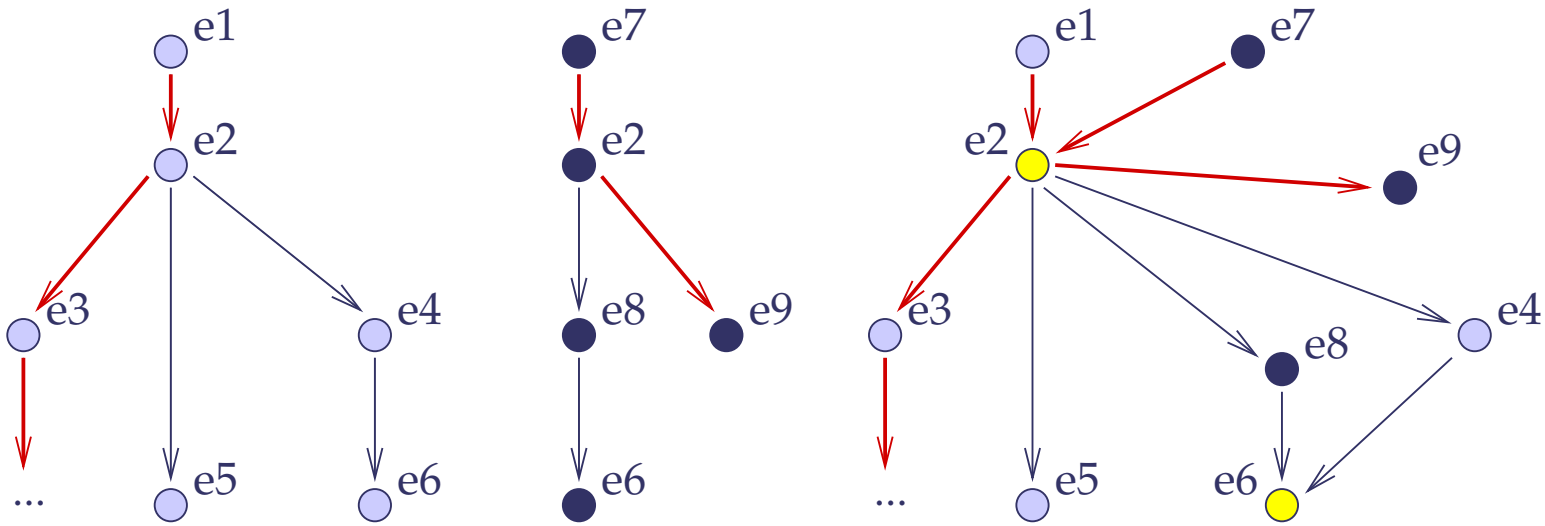
Σ_I-Ereignisstrukturen

- $E = \bigcup \mathbf{E}$ heißt Σ_I-Ereignisstruktur:
  - $\Sigma_I = (Id, \mathbf{Ac})$ (Instanzsignatur)
  - $\mathbf{E} = \{E_i\}_{i \in Id}$ (Vereinigung aller Ereignisstrukturen)
- $L = (Lc, \rightarrow) = \bigcup \mathbf{L}$ heißt verteilter Lebenszyklus:
  - $L \subseteq E$
  - $\mathbf{L} = \{L_i\}_{i \in Id} \subseteq \mathbf{E}$ mit $L_i \subseteq E_i$ für jedes i $\in Id$

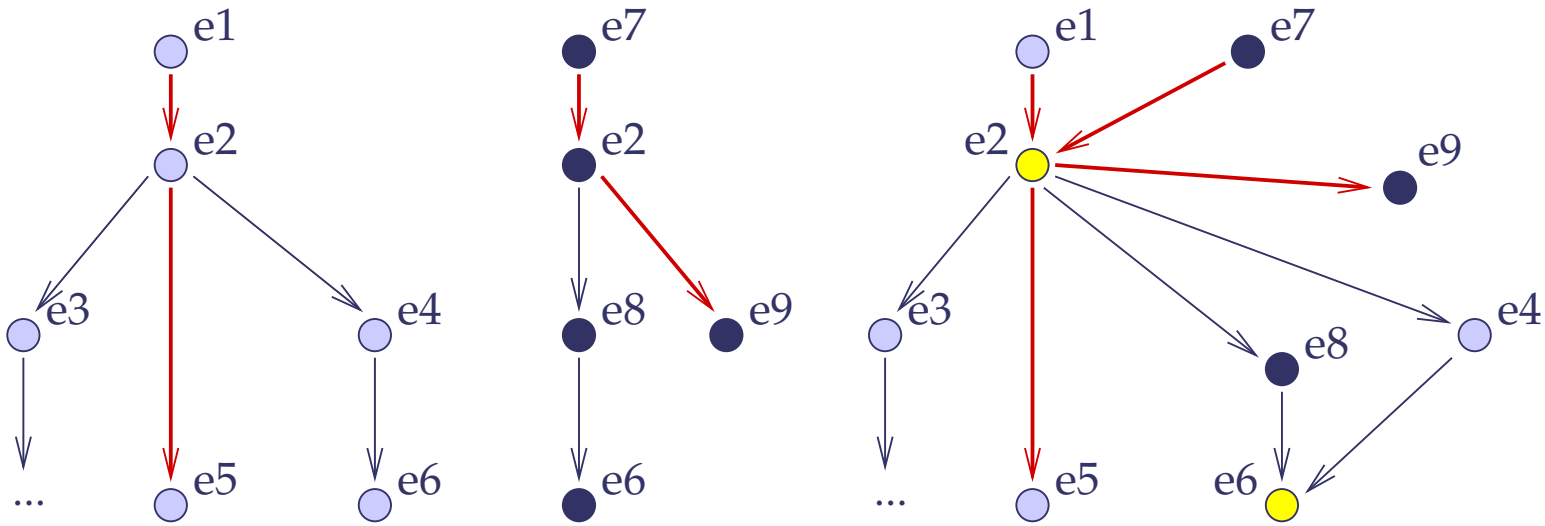
Σ_I-Ereignisstrukturen: Beispiel



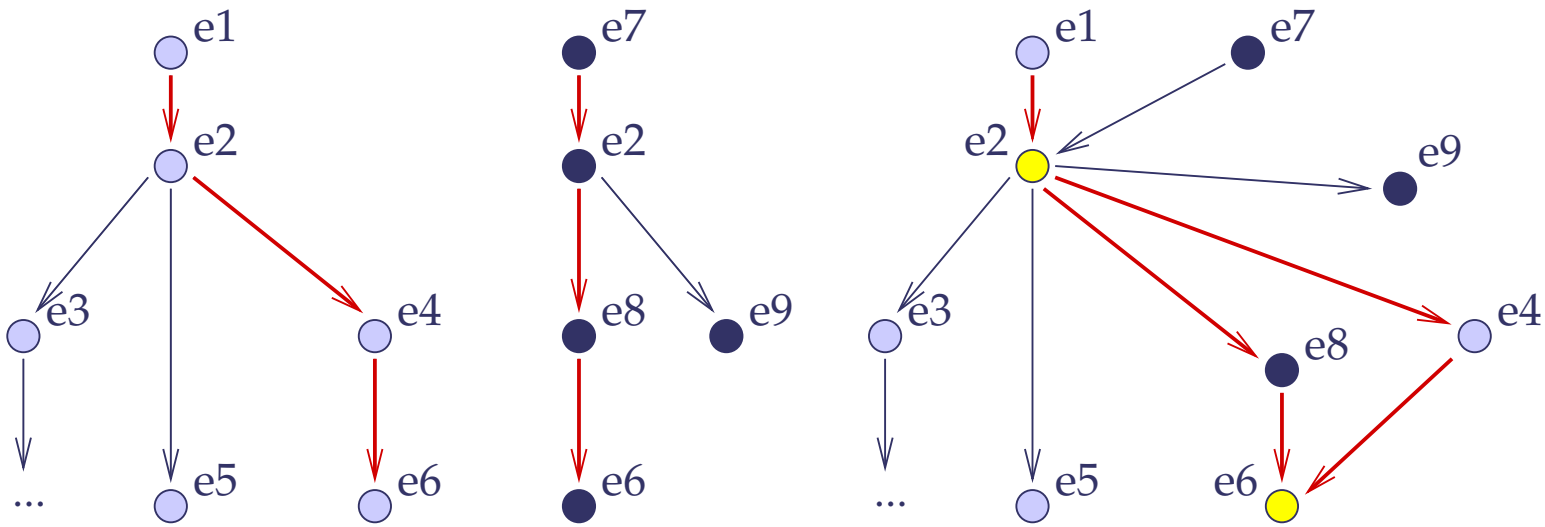
Σ_I-Ereignisstrukturen: Beispiel



Σ_I-Ereignisstrukturen: Beispiel



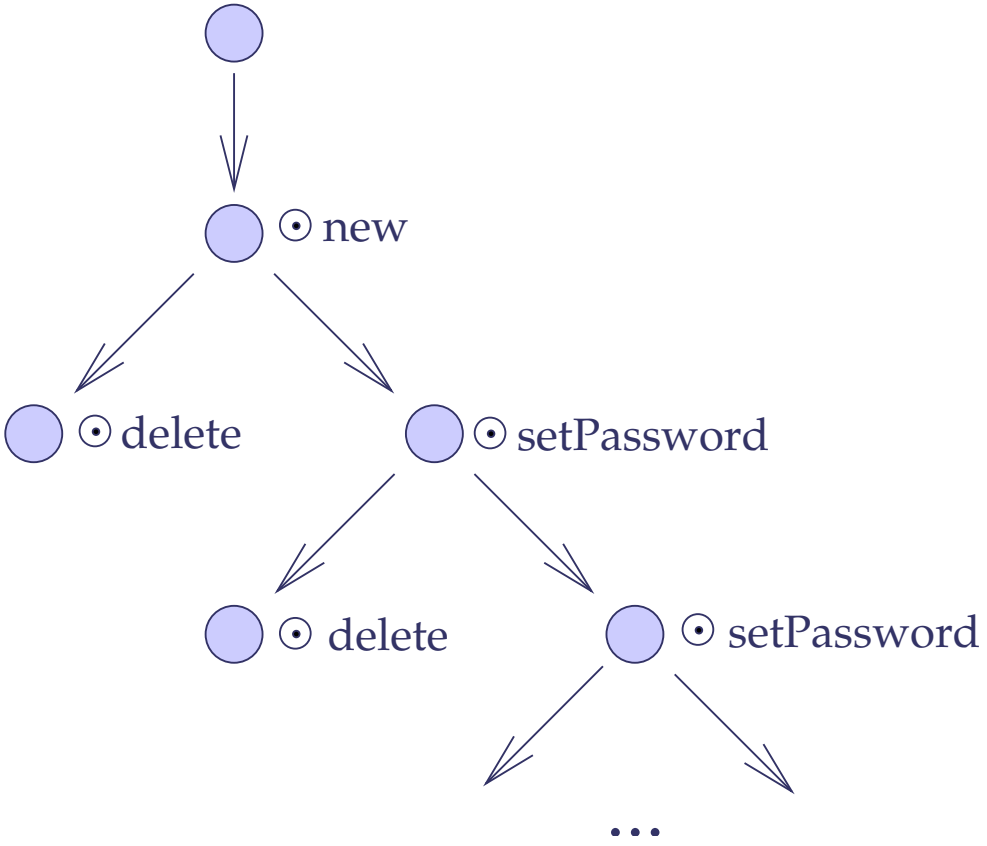
Σ_I-Ereignisstrukturen: Beispiel



Beschriftungen von Ereignisstrukturen

- Beschriftung einer Σ_I-Ereignisstruktur $E = (Ev, \rightarrow)$
  - $\bar{\alpha} : Ev_+ \rightarrow \bigcup \mathbf{Ac}$
  - $\bar{\alpha} = \bigcup \alpha$, $\alpha = \{\alpha_i : Ev_{i+} \rightarrow Ac_i\}_{i \in Id}$
  - Für alle Ereignisse $e_1, e_2 \in Ev$ gilt:
    $\bar{\alpha}(e_1) \neq \bar{\alpha}(e_2)$, wenn $e \in Ev$ mit $e \rightarrow e_1$ und $e \rightarrow e_2$

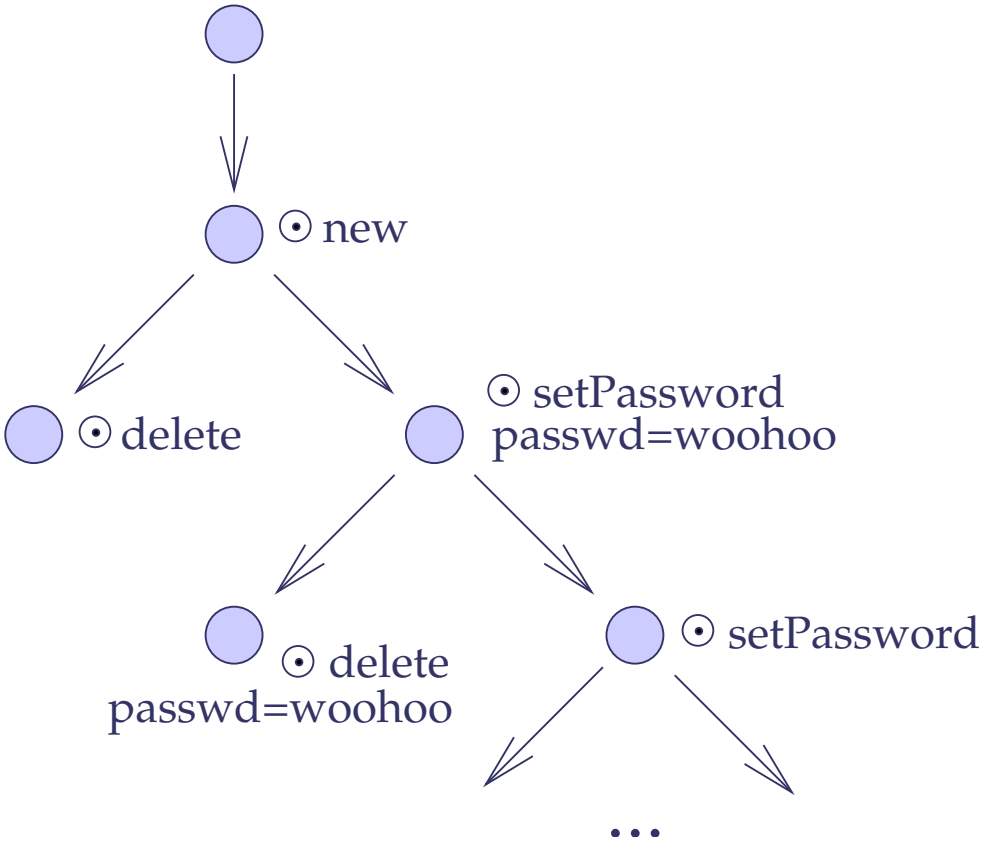
Beschriftungen: Beispiel



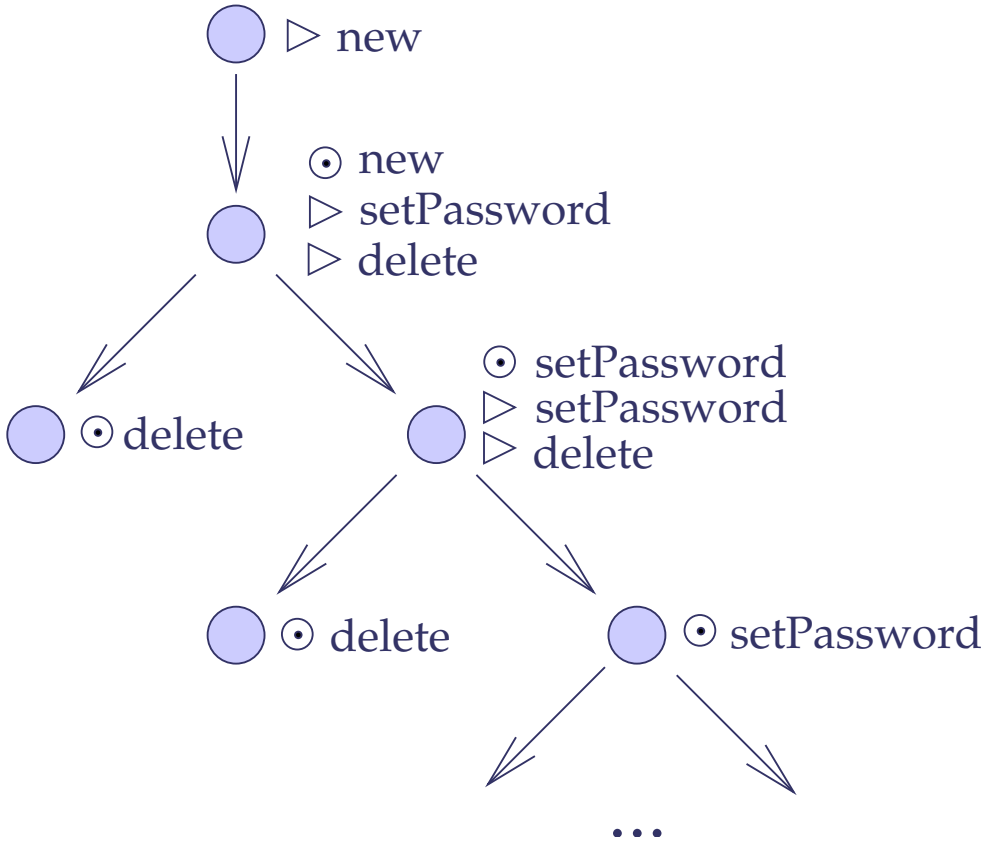
Temporallogische $L_\Sigma(\mathbf{X})$-Ausdrücke

- Syntaktisch korrekte $L_\Sigma(\mathbf{X})$-Ausdrücke:
  - $(t_1 =_{ss} t_2) \in L_\Sigma(\mathbf{X})$ für $t1, t2 \in T_\Sigma(\mathbf{X})_s$
  - $\odot\alpha(i), \triangleright\, \alpha(i) \in L_\Sigma(\mathbf{X})$
    für $\alpha(i) \in T_\sigma(\mathbf{X}_{ac})$ und $i \in T_\Sigma(\mathbf{X})_{id}$ Identität von α
  - $(\neg\phi) \in L_\Sigma(\mathbf{X})$ für $\phi \in L_\Sigma(\mathbf{X})$
  - $(\phi_1 \Rightarrow \phi_2) \in L_\Sigma(\mathbf{X})$ für $\phi_1, \phi_2 \in L_\Sigma(\mathbf{X})$
  - $(\mathsf{X}_{\,i}\phi), (\mathsf{F}_{\,i}\phi), (\mathsf{Y}_{\,i}\phi), (\mathsf{P}_{\,i}\phi) \in L_\Sigma(\mathbf{X})$
    für $i \in T_\Sigma(\mathbf{X})_{id}$ und $\phi \in L_\Sigma(\mathbf{X})$
- Lokale Ausdrücke: $\{i : \phi | i \in T_\Sigma(\mathbf{X}_{id}), \phi \in L_\Sigma(\mathbf{X})\}$

Temporallogische Ausdrücke: Beispiel



Temporallogische Ausdrücke: Beispiel



Semantik von $L_\Sigma(\mathbf{X})$-Ausdrücken (1/2)

- Semantik von $L_\Sigma(\mathbf{X})$-Ausdrücken
  - $\bar{L}, \downarrow e, \theta \models i : (t_1 =_{ss} t_2) \Leftrightarrow {t_1}_U^\theta = {t_2}_U^\theta$
  - $\bar{L}, \downarrow e, \theta \models i : \odot\alpha(j) \Leftrightarrow e \in \mathsf{Lc}_{i_U^\theta}$ und $\bar{\alpha}(e) = \alpha(j)$
  - $\bar{L}, \downarrow e, \theta \models i :\rhd \alpha(j) \Leftrightarrow e \in \mathsf{Lc}_{i_U^\theta}$
    und für ein $e_2 \in \mathsf{Ev}_{i_U^\theta}, e \rightarrow_{i_U^\theta} e_2$ und $\bar{\alpha}(e_2) = \alpha(j)$
  - $\bar{L}, \downarrow e, \theta \models i : (\neg\phi) \Leftrightarrow e \in \mathsf{Lc}_{i_U^\theta}$
    und nicht $\bar{L}, \downarrow e, \theta \models i : \phi$
  - $\bar{L}, \downarrow e, \theta \models i : (\phi_1 \Rightarrow \phi_2) \Leftrightarrow e \in \mathsf{Lc}_{i_U^\theta}$
    und ($\bar{L}, \downarrow e, \theta \models i : \phi_2$ oder nicht $\bar{L}, \downarrow e, \theta \models i : \phi_1$)

Semantik von $L_\Sigma(\mathbf{X})$-Ausdrücken (2/2)

- Semantik von $L_\Sigma(\mathbf{X})$-Ausdrücken
  - $\bar{L}, \downarrow e_1, \theta \vDash i : (\mathsf{X}_j \phi) \Leftrightarrow e_1 \in \mathit{Lc}_{j_U^\theta}$
    und $\bar{L}, \downarrow e_2, \theta \vDash j : \phi$ für ein Ereignis $e_2 \in \mathit{Lc}_{j_U^\theta}$
    so dass $e_1 \rightarrow_{i_U^\theta} e_2$ oder $e_1 \rightarrow_{j_U^\theta} e_2$
  - $\bar{L}, \downarrow e_1, \theta \vDash i : (\mathsf{F}_j \phi) \Leftrightarrow e_1 \in \mathit{Lc}_{j_U^\theta}$
    und $\bar{L}, \downarrow e_2, \theta \vDash j : \phi$ für ein Ereignis $e_2 \in \mathit{Lc}_{j_U^\theta}$
    so dass $e_1 \rightarrow^* e_2$
  - $\bar{L}, \downarrow e_1, \theta \vDash i : (\mathsf{Y}_j \phi)$ und $\bar{L}, \downarrow e_1, \theta \vDash i : (\mathsf{P}_j \phi)$ analog

Beispiel: Cocopeli (1/5)

class User

uses nat, string, boolean;

attributes mtknr:nat, name,login,pwd:string, verified:boolean;

actions new(nat,string,string), delete, setPassword(string);

axioms *var* m:nat; n,l,p:string;

$\odot$ new(m,n,l,p)
$\Rightarrow (\neg \rhd \text{new} \wedge \rhd \text{setPassword} \wedge \rhd \text{delete})\ U^{\circ}\ \odot \text{delete}$
$\wedge$ (mtknr=m $\wedge$ name=n $\wedge$ login=l $\wedge$ pwd=p $\wedge$ verified=false),
$\odot$ delete
$\Rightarrow \neg \rhd \text{new} \wedge \neg \rhd \text{setPassword} \wedge \neg \rhd \text{delete}$,
$\odot$ setPassword(p)
$\Rightarrow \text{pwd=p}\ \ W^{\circ}\ \odot \text{setPassword}$

Beispiel: Cocopeli (2/5)

class User

uses nat, string, boolean;

attributes mtknr:nat, name,login,pwd:string, verified:boolean;

actions *new(nat,string,string), +delete, setPassword(string);

axioms *var* m:nat; n,l,p:string;

$\odot$ new(m,n,l,p)
$\Rightarrow$ (mtknr=m $\wedge$ name=n $\wedge$ login=l $\wedge$ pwd=p $\wedge$ verified=false),
$\odot$ setPassword(p)
$\Rightarrow$ pwd=p W° $\odot$ setPassword

Beispiel: Cocopeli (3/5)

class Database

uses User;

actions *new, +delete,
changePassword, checkPassword ;

axioms *var* u:user; old,new:string;

$\odot$ changePassword(user,old,new) $\wedge$ user.verified
$\Rightarrow \neg \triangleright$ changePassword U° $\odot$ user.setPassword(new),
$\odot$ checkPassword(user,pwd) $\wedge$ user.pwd=pwd
$\Rightarrow$ user.verified
W° ($\odot$ checkPassword $\vee \odot$user.setPassword),
$\odot$ checkPassword(user,pwd) $\wedge$ user.pwd$\neq$pwd
$\Rightarrow \neg$ user.verified
W° ($\odot$ checkPassword; $\vee \odot$user.setPassword)

Beispiel: Cocopeli (4/5)

class Negotiator

uses User, Database;

attributes user:User, db:Database;

actions *new(User), +delete,
clickChangePassword, submitChangePassword;

axioms *var* u:user; old,new:string;

$\odot$ create(u)
$\Rightarrow$ user=u
$\odot$ clickChangePassword
$\Rightarrow \triangleright$ submitChangePassword
$\odot$ submitChangePassword(old,new)
$\Rightarrow \odot$ db.changePassword(user,new)

Beispiel: Cocopeli (5/5)

Mögliches Anwendungsszenario:

- Nebenläufige Komposition zweier Negotiator-Objekte
- Ein Benutzer zweimal angemeldet
- Passwortänderung konsistent?

Objektorientierte Erweiterungen

- eingebettet durch Morphismen
- definiert für
  - Spezialisierung
  - Generalisierung
  - Überschreiben
  - Verstecken
  - ...

Zusammenfassung

- Ansatz: Algebraische Spezifikation
- Modell: Ereignisstrukturen
- Beschreibung: Temporale Logik
- Erfolg versprechend?
  - viele ungelöste Details
  - kein Beweiser
  - keine Veröffentlichungen in letzter Zeit