Logics for Traces

Christian Kissig, Alexander Kurz

Sierra Nevada, January 22, 2008

(Non-Deterministic) Labelled Transition Systems

Definition (Labelled Transition Systems)

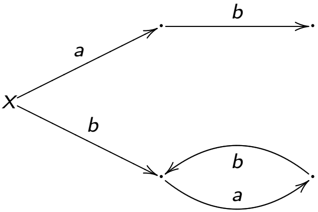
A **Labelled Transition System [LTS]** is a tuple

$$\langle N, \Sigma, R \subseteq N \times \Sigma \times N \rangle$$

consisting of

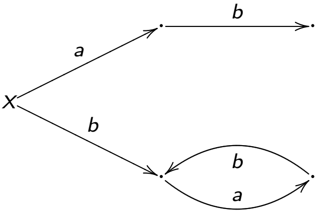
- a set N of **nodes**
- a set Σ of **labels**
- a **relation** $R \subseteq N \times \Sigma \times N$

Traces for LTS



[R.V.Glabbeek, “The Linear Time-Branching-Spectrum”]

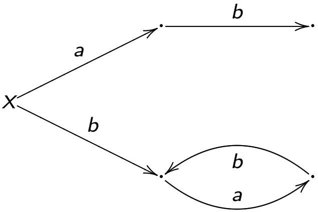
Traces for LTS



- **finite traces** from x: $\{ab\}$

[R.V.Glabbeek, "The Linear Time-Branching-Spectrum"]

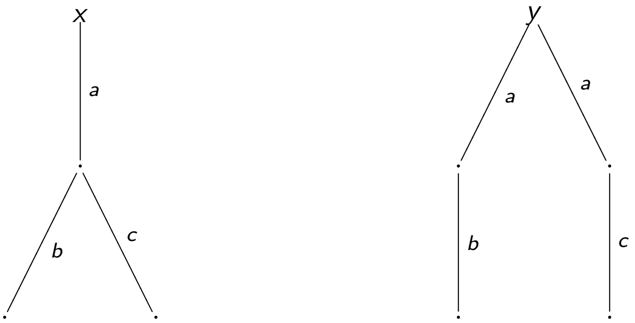
Traces for LTS



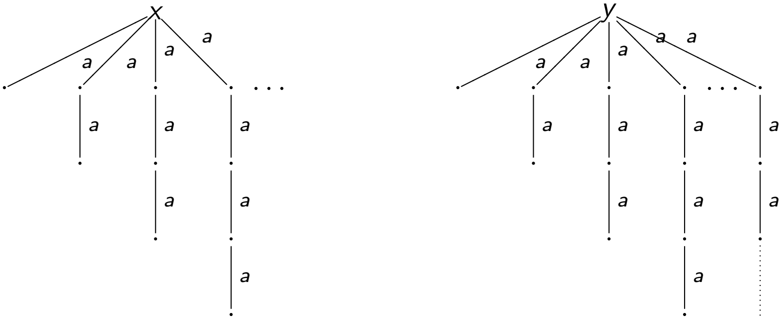
- **finite traces** from x: $\{ab\}$
- **infinite traces** from x: $\{bababab\cdots\}$

[R.V.Glabbeek, "The Linear Time-Branching-Spectrum"]

Bisimulation vs Trace Equivalence for LTS (1)



Bisimulation vs Trace Equivalence for LTS (2)



Coalgebras

Let $\mathcal{T}$ be a **monad** and $\mathcal{F}$ an **endofunctor** on $\mathcal{S}et$ and let X be a set.

Definition (Coalgebras)

A $\mathcal{T}\mathcal{F}$-coalgebra c with carrier X is a map $c : X \to \mathcal{T}\mathcal{F}X$.

Coalgebras

Let $\mathcal{T}$ be a **monad** and $\mathcal{F}$ an **endofunctor** on $\mathcal{S}et$ and let X be a set.

Definition (Coalgebras)

A $\mathcal{T}\mathcal{F}$-coalgebra c with carrier X is a map $c : X \to \mathcal{T}\mathcal{F}X$.

We understand

- $\mathcal{T}$ as the **branching** type
- $\mathcal{F}$ as the **transition** type

[B. Jacobs, "Introduction to Coalgebra. Towards mathematics of states and observations."]

Examples for Coalgebras

- Non-Deterministic LTS: $\mathcal{T} = \mathcal{P}$, $\mathcal{F} = 1 + \Sigma \times (-)$
- Probabilistic LTS: $\mathcal{T} = \mathcal{D}$, $\mathcal{F} = 1 + \Sigma \times (-)$
- LTS with Termination and Deadlock: $\mathcal{T} = 1 + (-)$, $\mathcal{F} = 1 + \Sigma \times (-)$
- Context-free Grammars: $\mathcal{T} = \mathcal{P}$, $\mathcal{F} = ((-) + \Sigma)^*$

The Kleisli-Category, syntactically

Given a monad $\langle \mathcal{T}, \mu, \eta \rangle$, we define the Kleisli-Category $Kl(\mathcal{T})$ over $\mathcal{S}et$:

- objects X in $Kl(\mathcal{T})$ are objects X over $\mathcal{S}et$
- arrows $X \xrightarrow{f} Y$ in $Kl(\mathcal{T})$ are arrows $X \xrightarrow{f} \mathcal{T}(Y)$ in $\mathcal{S}et$
- the composition $X \xrightarrow{f} Y \xrightarrow{g} Z$ in $Kl(\mathcal{T})$
  is the composition $X \xrightarrow{f} \mathcal{T}(Y) \xrightarrow{\mathcal{T}g} \mathcal{T}\mathcal{T}(Z) \xrightarrow{\mu_Z} \mathcal{T}Z$ in $\mathcal{S}et$

Distributive Laws

Definition (Distributive Laws)

A distributive law π for a monad $\langle \mathcal{T}, \mu, \eta \rangle$ and a functor $\mathcal{F}$ is a natural transformation $\pi : \mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$ which is compatible with the monad structure :

- $\pi \circ \mathcal{F}\eta = \eta_{\mathcal{F}}$
- $\pi \circ \mathcal{F}\mu = \mu_{\mathcal{F}} \circ \mathcal{T}\pi \circ \pi_{\mathcal{T}}$

Distributive Laws

Definition (Distributive Laws)

A distributive law π for a monad $\langle \mathcal{T}, \mu, \eta \rangle$ and a functor $\mathcal{F}$ is a natural transformation $\pi : \mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$ which is compatible with the monad structure :

- $\pi \circ \mathcal{F}\eta = \eta_{\mathcal{F}}$
- $\pi \circ \mathcal{F}\mu = \mu_{\mathcal{F}} \circ \mathcal{T}\pi \circ \pi_{\mathcal{T}}$

Provided a **distributive law** $\pi : \mathcal{F}\mathcal{T} \Rightarrow \mathcal{T}\mathcal{F}$, there is a **lifting** $\overline{\mathcal{F}}$ of the functor $\mathcal{F}$ to $Kl(\mathcal{T})$

- $\overline{\mathcal{F}} : X \mapsto \mathcal{F}X$
- $\overline{\mathcal{F}} : X \xrightarrow{f} Y \mapsto \overline{\mathcal{F}}X \xrightarrow{\pi_Y \circ \mathcal{F}f} \overline{\mathcal{F}}Y$

Trace Semantics in $Kl(\mathcal{T})$

- if $\mathcal{F}$ preserves ω-colimits, the initial sequence in $\mathcal{S}et$ yields the **initial** $\mathcal{F}$**-algebra** in $\mathcal{S}et$.

[Hasuo, Jacobs, Sokolova, "Generic Trace Theory"]

Trace Semantics in $Kl(\mathcal{T})$

- if $\mathcal{F}$ preserves ω-colimits, the initial sequence in $\mathcal{S}et$ yields the **initial $\mathcal{F}$-algebra** in $\mathcal{S}et$.
- provided a distributive law, the initial sequence can be lifted to $Kl(\mathcal{T})$ and yields the **initial $\overline{\mathcal{F}}$-algebra** $\alpha : \overline{\mathcal{F}}A \xrightarrow{\cong} A$ in $Kl(\mathcal{T})$.

[Hasuo, Jacobs, Sokolova, "Generic Trace Theory"]

Trace Semantics in $Kl(\mathcal{T})$

- if $\mathcal{F}$ preserves ω-colimits, the initial sequence in $\mathcal{S}et$ yields the **initial $\mathcal{F}$-algebra** in $\mathcal{S}et$.
- provided a distributive law, the initial sequence can be lifted to $Kl(\mathcal{T})$ and yields the **initial $\overline{\mathcal{F}}$-algebra** $\alpha : \overline{\mathcal{F}}A \xrightarrow{\cong} A$ in $Kl(\mathcal{T})$.
- (Smyth, Plotkin) if
  - the Kleisli-category is $DCPO_\perp$-enriched with composition left-strict and
  - $\overline{\mathcal{F}}$ is locally monotone

  the initial $\overline{\mathcal{F}}$-algebra coincides with the **final $\overline{\mathcal{F}}$-coalgebra** $\zeta : \overline{\mathcal{F}}Z \xrightarrow{\cong} Z$

[Hasuo, Jacobs, Sokolova, "Generic Trace Theory"]

Trace Semantics in $Kl(\mathcal{T})$

- if $\mathcal{F}$ preserves ω-colimits, the initial sequence in $\mathcal{S}et$ yields the **initial $\mathcal{F}$-algebra** in $\mathcal{S}et$.
- provided a distributive law, the initial sequence can be lifted to $Kl(\mathcal{T})$ and yields the **initial $\overline{\mathcal{F}}$-algebra** $\alpha : \overline{\mathcal{F}}A \xrightarrow{\cong} A$ in $Kl(\mathcal{T})$.
- (Smyth, Plotkin) if
  - the Kleisli-category is $DCPO_{\perp}$-enriched with composition left-strict and
  - $\overline{\mathcal{F}}$ is locally monotone

  the initial $\overline{\mathcal{F}}$-algebra coincides with the **final $\overline{\mathcal{F}}$-coalgebra** $\zeta : \overline{\mathcal{F}}Z \xrightarrow{\cong} Z$
- we obtain for any $\overline{\mathcal{F}}$-coalgebra c in $Kl(\mathcal{T})$ the **trace map** tr_c into ζ

[Hasuo, Jacobs, Sokolova, "Generic Trace Theory"]

Logic for Traces: Syntax

From now on let $\mathcal{T} = \mathcal{P}$.

Logic for Traces: Syntax

From now on let $\mathcal{T} = \mathcal{P}$.

Definition (Syntax)

- $Fma_0 = \emptyset$
- $Fma_{n+1} = [\mathcal{F} Fma_n]_{\perp,\vee}$
- $Fma = \bigcup_{n<\omega} Fma_n$

Logic for Traces: Syntax

From now on let $\mathcal{T} = \mathcal{P}$.

Definition (Syntax)

- $Fma_0 = \emptyset$
- $Fma_{n+1} = [\mathcal{F} Fma_n]_{\bot,\vee}$
- $Fma = \bigcup_{n<\omega} Fma_n$

Concretely, for $\mathcal{F} = 1 + \Sigma \times (-)$ where $1 = \{\checkmark\}$ and $a, b \in \Sigma$,

- $Fma_0 = \emptyset$
- $Fma_1 = \{\bot, \checkmark, \checkmark \vee \checkmark, \cdots\}$
- $Fma_2 = \{\bot, \checkmark, \checkmark \vee \checkmark, \cdots, (a, \checkmark), \cdots, (b, \checkmark), \cdots, (a, \checkmark) \vee (b, \checkmark), \cdots\}$
- $\ldots$

Logic for Traces: Semantics

Definition (Modality for $\mathcal{P}$)

Define a modality $\lambda : 2^{(-)} \Rightarrow 2^{\mathcal{P}(-)}$ by
$\lambda_Y(Y' \subseteq Y) := \{Y'' \subseteq Y \mid Y' \cap Y'' \neq \emptyset\}$ for all sets Y.

Logic for Traces: Semantics

Definition (Modality for $\mathcal{P}$)

Define a modality $\lambda : 2^{(-)} \Rightarrow 2^{\mathcal{P}(-)}$ by
$\lambda_Y(Y' \subseteq Y) := \{Y'' \subseteq Y \mid Y' \cap Y'' \neq \emptyset\}$ for all sets Y.

Definition (Semantics)

Let c be a $\overline{\mathcal{F}}$-coalgebra with carrier X in $Kl(\mathcal{T})$, then for all $x \in X$,

- $x \nVdash_c \bot$
- $x \Vdash_c \phi \vee \psi$ iff $x \Vdash_c \phi$ or $x \Vdash_c \psi$
- $x \Vdash_c \nabla\phi$ iff $\exists x' \in c(x).x'(\overline{\mathcal{F}} \Vdash_c)\phi$,
  i.e. iff $x \in c^{-} \circ \lambda_{\mathcal{F}X}((\overline{\mathcal{F}} \Vdash_c)[\phi])$

Expressivity

Theorem

The proposed logic does not distinguish between trace-equivalent states.

For all $\mathcal{PF}$-coalgebras $c : X \to \mathcal{PF}X$ and points $x, y \in X$, if $tr_c(x) = tr_c(y)$ then $x \Vdash_c \phi$ iff $y \Vdash_c \phi$ for all $\phi \in Fma$.

Conclusions

Work in Progress :

- more boolean operators
- more monads $\mathcal{T}$
- more expressive modalities for $\mathcal{T}$

Interesting (for me) :

- axiomatisation of trace logics
- infinite traces

Conclusions

¡muchas gracias!