A Semiring-based Trace Semantics for Processes

David Clark

26 May 2010

Some Context

- Motivated by a desire to find a syntax directed leakage analysis for process languages
- Joint work with Michele Boreale (Florence) and Daniele Gorla (Rome)

Scenario

Consider a discrete-time, non-deterministic system

High events

- updates of high variables under the control of a secret scheduler
- not directly observable from outside

Scenario

Low events

- other events are observable
- e,g, certain variables, input/output actions, file accesses
- not under control of the secret scheduler
- may have prescribed or known non-deterministic or probabilistic behaviour

Scenario

Attacker

- An attacker has constraints and abilities
- able to observe only at prescribed times, e.g. at termination
- can repeatedly execute the system and make more observations
- high scheduler remains the same between executions

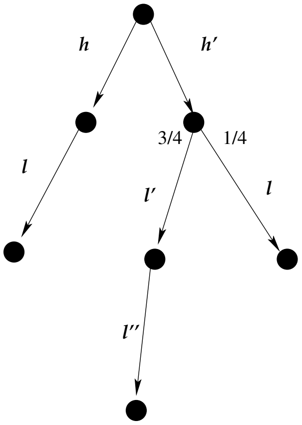
Combining Observations

- Combine a sequence of consecutive observations
  - e.g. o_1, o_2, o_3, results into a combined observation, $o = o_1 \star o_2 \star o_3$
  - only o may be available to the attacker
- Combine observations arising from repeated executions of the system into a global observation
  - e.g. $o_1 \star o_2$ and $o_3 \star o_4$, into a global observation $(o_1 \star o_2) + (o_3 \star o_4)$

Attacker's Deductions

- sequence of high events, π, corresponds to a global observation o
- we call this mapping $\mathcal{L}(P)$
- $\mathcal{L}(P)$ can be deduced from P's specification
- from o and $\mathcal{L}(P)$ attacker can deduce information about π
- e.g. does $\pi \in (\mathcal{L}(P))^{-1}(o)$?

An example of $\mathcal{L}(P)$



$\mathcal{L}(P)$ as a mapping

$\mathcal{L}(P)(h) = [l \mapsto 1]$ and
$\mathcal{L}(P)(h') = [l'l'' \mapsto \frac{3}{4},\ l \mapsto \frac{1}{4}]$

Designer's Calculations

- designer constructs system specification, assesses the security
- central object of interest is $\mathcal{L}(P)$
- system is secure if $\mathcal{L}(P)$ is a constant function
- related to non-deducibility on strategies [Wittbold and Johnson 1990]
- at least minimise the number of partitions of high sequences induced by $(\mathcal{L}(P))^{-1}$
- may want to perform quantitative measures relating to tolerable flow quantity thresholds
- designer must be able to generate $\mathcal{L}(P)$ and reason about it, preferably in a compositional, syntax-driven way

Formalising the Scenario

- observable events are elements of a semiring $\mathbb{S}$, whose product and sum correspond to the $\star$ and $+$ operations
- a set of unobservable, high-events H is assumed
- the security significant behaviour of the system, $\mathcal{L}(P)$, is then a mapping from H^* to $\mathbb{S}$
- this is called a formal power series ($\textsc{fps}$) on H and $\mathbb{S}$

Formalising the Scenario

- provide a simple process calculus to specify systems
- give the language a semantics in terms of Moore Automata
- characterise the semantic mapping $\mathcal{L}(\cdot)$ in terms of the unique homomorphism from this calculus into the set of formal power series seen as a final coalgebra [c.f. Rutten]
- provide a compositional semantics of the calculus in terms of rational operators on FPS's, defined via behavioural differential equations (BDE's) [Rutten again]
- show that the final and the compositional semantics coincide

Benefits of the Two Semantics

- the final semantics allows for reasoning – proving equivalences – on systems by co-induction
- the compositional semantics, the BDE's, can be used for step-wise, syntax-driven generation of the behaviours $\mathcal{L}(P)$, for any P

Semirings

Definition of a Semiring

a *semiring* $\mathbb{S}$ is a tuple $(S, +, \times, 0, 1)$ such that $(S, +, 0)$ is a commutative monoid, $(S, \times, 1)$ is a monoid, $\times$ distributes over $+$ both on the left and on the right, and 0 annihilates both on the left and on the right (i.e., $0 \times o = o \times 0 = 0$ for each $o \in S$)

- A semiring is a ring without additive inverses
- examples include natural numbers, $\mathbb{N}$, the nonnegative reals $\mathbb{R}^+$
- simplest possible semiring is $\mathbb{B}$, obtained by taking $S = \{0, 1\}$ and $+$ and $\times$ to be the sum and product of booleans, that is *or* and *and*

The Semiring of Weighted (Low-)Traces

Elements of this semiring are low observations in our process algebra

Definition of $\mathbb{WL}$

- fix a finite, non-empty alphabet L, ranged over by $l, l', \ldots$
- let $\lambda, \lambda', \ldots$ range over L^*
- elements of $\mathbb{WL}$ are functions $o : L^* \to \mathbb{R}^+$
- $(o_1 + o_2)(\lambda) = o_1(\lambda) + o_2(\lambda)$
- $(o_1 \times o_2)(\lambda) = \sum_{\lambda', \lambda'' : \lambda' \lambda'' = \lambda} o_1(\lambda') \times o_2(\lambda'')$

$\mathbb{WL}$ and Probability Distributions

$\mathbb{WL}$ includes

- all functions $o : L^* \to [0,1]$ such that $\sum_{\lambda \in L^*} o(\lambda) = 1$, that is, all probability distributions on low traces
- all functions o such that $\sum_{\lambda \in L^*} o(\lambda) \leq 1$, that is, all probability sub-distributions on low traces
- neither of these form a semiring

A Process Calculus

- fix a finite, non-empty alphabet H, ranged over by $h, h', \ldots$
- let $\pi, \pi', \ldots$ range over H^*
- fix a semiring $\mathbb{S}$

The set of all processes

$P ::= o \mid h \mid P + P \mid P; P \mid P\langle f \rangle \mid P^*$

where

- $o \in \mathbb{S}$, $h \in H$ and $f : \mathbb{S} \to \mathbb{S}$ is a semiring morphism
- $+$, ; and $*$ denote nondeterministic choice, sequential composition and iteration respectively

BDE's

Initial condition	Condition on derivatives
$o(\epsilon) \triangleq o$	$(o)_h \triangleq 0_{\mathbb{F}}$
$h(\epsilon) \triangleq 0$	$(h)_{h'} \triangleq \begin{cases} 1_{\mathbb{F}} & \text{if } h = h' \\ 0_{\mathbb{F}} & \text{otherwise} \end{cases}$
$(\sigma + \sigma')(\epsilon) \triangleq \sigma(\epsilon) + \sigma'(\epsilon)$	$(\sigma + \sigma')_h \triangleq \sigma_h + \sigma'_h$
$(\sigma ; \sigma')(\epsilon) \triangleq \sigma(\epsilon) \times \sigma'(\epsilon)$	$(\sigma ; \sigma')_h \triangleq \sigma_h ; \sigma' + \sigma(\epsilon) \times \sigma'_h$
$(\sigma\langle f \rangle)(\epsilon) \triangleq f(\sigma(\epsilon))$	$(\sigma\langle f \rangle)_h \triangleq (\sigma_h)\langle f \rangle$
$(\sigma^*)(\epsilon) \triangleq \begin{cases} 1 & \text{if } \sigma(\epsilon) = 0 \\ 0 & \text{otherwise} \end{cases}$	$(\sigma^*)_h \triangleq \sigma_h ; \sigma^*$

Example: Dependency Matrix

Result of BDE analysis

$$M(o_1.h.(o_2.h' + o_3.h')) \triangleq \begin{array}{c|cccccc} & l_1 & l_2 & l_1 l_3 & l_1 l_4 & l_2 l_3 & l_2 l_4 \\ \hline \epsilon & 1 & 2 & 0 & 0 & 0 & 0 \\ h & 0 & 0 & 4 & 4 & 8 & 8 \\ hh' & 0 & 0 & 4 & 4 & 8 & 8 \end{array}$$

where

$$o_1 \triangleq \begin{array}{l} l_1 \mapsto 1 \\ l_2 \mapsto 2 \end{array}$$

$$o_2 \triangleq \begin{array}{l} l_3 \mapsto 3 \\ l_4 \mapsto 4 \end{array}$$

$$o_3 \triangleq l_3 \mapsto 1$$

Conclusions

- Compositional construction of dependency matrices
- excludes parallel operator at present (difficult but not impossible)
- iterator and filter and sequential composition allow imperative update modelling as well as protocols
- underlying theory works for any semiring – so not limited to $\mathbb{WL}$
- future work: expand language, expand instantiations of semiring