First steps towards probabilistic higher-order model-checking

Charles Grellois   Ugo dal Lago

FOCUS Team – INRIA & University of Bologna

GDRI-LL Meeting on Intersection Types
June 14, 2016

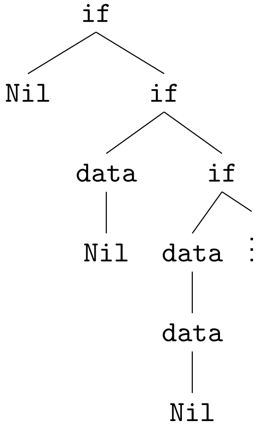
Roadmap

1. A reminder of higher-order model-checking (HOMC) and intersection types for HOMC
2. Non-idempotent intersection types and HOMC for almost-sure MSO properties
3. Automata for probabilistic properties (weaker than MSO), intersection types, tensorial logic with effects

Higher-order model-checking

Model-checking

$\mathcal{T} \quad = $



ϕ a logical property on trees, e.g. "all executions are finite".

Model-checking: does $\mathcal{T} \vDash \phi$?

Higher-order model-checking

Infinite trees with a finite representation: a λY-term or a higher-order recursion scheme (HORS).

$$\mathcal{G} \;=\; \begin{cases} \texttt{S} & = \quad \texttt{L Nil} \\ \texttt{L}\ x & = \quad \texttt{if}\ x\ (\texttt{L}\ (\texttt{data}\ x\ )\ ) \end{cases}$$

$\longrightarrow$ model-checking on λ-terms or HORS.

Alternating tree automata

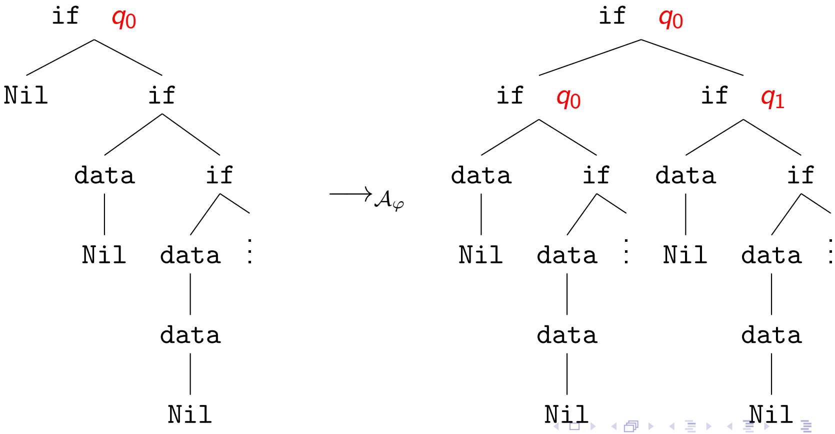
ATA: non-deterministic tree automata whose transitions may duplicate or drop a subtree.

Typically: $\delta(q_0, \texttt{if}) = (2, q_0) \wedge (2, q_1)$.

Alternating tree automata

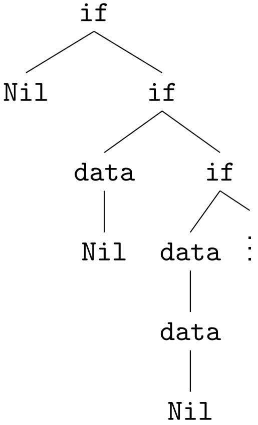
ATA: non-deterministic tree automata whose transitions may duplicate or drop a subtree.

Typically: $\delta(q_0, \texttt{if}) = (2, q_0) \wedge (2, q_1)$.



Alternating parity tree automata

Express reachability with ATA: does every branch ends by Nil?



Problem: ATA execute coinductively.

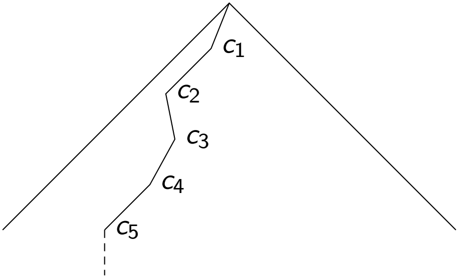
Solution: parity condition.

Alternating parity tree automata

Each state of an APT is attributed a color

$$\Omega(q) \in Col \subseteq \mathbb{N}$$

An infinite branch of a run-tree is winning iff the maximal color among the ones occuring infinitely often along it is even.



Alternating parity tree automata

Each state of an APT is attributed a color

$$\Omega(q) \in Col \subseteq \mathbb{N}$$

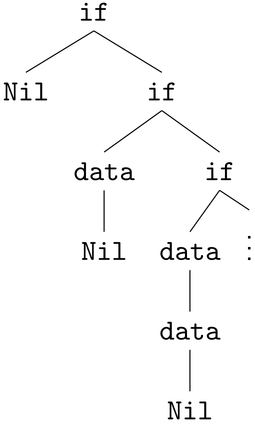
An infinite branch of a run-tree is winning iff the maximal color among the ones occuring infinitely often along it is even.

A run-tree is winning iff all its infinite branches are.

For a MSO formula φ:

$\mathcal{A}_\varphi$ has a winning run-tree over $\langle \mathcal{G} \rangle$ iff $\langle \mathcal{G} \rangle \vDash \phi$.

Alternating parity tree automata



$$Q = \{q\}$$

$$\Omega(q) = 1$$

$$\delta(\texttt{if}, q) = (1, q) \wedge (2, q)$$

$$\delta(\texttt{data}, q) = (1, q)$$

$$\delta(\texttt{Nil}, q) = \top$$

HOMC and intersection types

Alternating tree automata and intersection types

A key remark (Kobayashi 2009):

$$\delta(q_0, \texttt{if}) \;=\; (2, q_0) \wedge (2, q_1)$$

can be seen as the intersection typing

$$\texttt{if} \;:\; \emptyset \rightarrow (q_0 \wedge q_1) \rightarrow q_0$$

refining the simple typing

$$\texttt{if} \;:\; o \rightarrow o \rightarrow o$$

(this talk is NOT about filter models!)

Alternating tree automata and intersection types

A run-tree over $\texttt{if}\ \ T_1\ \ T_2$ is a derivation of $\emptyset\ \vdash\ \texttt{if}\ \ T_1\ \ T_2$:

$$
\dfrac{\dfrac{\dfrac{}{\emptyset \vdash \texttt{if} : \emptyset \to (q_0 \wedge q_1) \to q_0}\;\delta \qquad \color{red}{\emptyset}}{\emptyset \vdash \texttt{if}\ T_1 : (q_0 \wedge q_1) \to q_0}\;\text{App} \qquad \dfrac{\vdots}{\color{red}{\emptyset \vdash T_2 : q_0}} \qquad \dfrac{\vdots}{\color{red}{\emptyset \vdash T_2 : q_1}}}{\emptyset \vdash \texttt{if}\ T_1\ T_2 : q_0}\;\text{App}
$$

Intersection types naturally lift to higher-order – and thus to $\mathcal{G}$, which finitely represents $\langle \mathcal{G} \rangle$.

Theorem (Kobayashi)

$S : q_0 \vdash S : q_0$ *iff* *the ATA $\mathcal{A}_\varphi$ has a run-tree over $\langle \mathcal{G} \rangle$.*

Here: variant with non-idempotent types.

Under connection Rel/non-idempotent types, we obtain a similar denotational theorem.

A type-system for verification: without parity conditions

Axiom $$\frac{}{x \,:\, \bigwedge_{\{i\}} \;\theta_i \;::\; \kappa \;\vdash\; x \,:\, \theta_i \;::\; \kappa}$$

δ $$\frac{\{\,(i, q_{ij}) \mid 1 \leq i \leq n, 1 \leq j \leq k_i\,\} \quad \text{satisfies} \quad \delta_A(q, a)}{\emptyset \vdash a \,:\, \bigwedge_{j=1}^{k_1} \; q_{1j} \;\rightarrow\; \ldots \;\rightarrow\; \bigwedge_{j=1}^{k_n} \; q_{nj} \rightarrow q \;::\; o \rightarrow \cdots \rightarrow o}$$

App $$\frac{\Delta \vdash t : (\;\theta_1 \;\wedge \cdots \wedge\; \theta_k) \rightarrow \theta \;::\; \kappa \rightarrow \kappa' \qquad \Delta_i \vdash u \,:\, \theta_i \;::\; \kappa}{\Delta \,+\, \Delta_1 \,+\, \ldots \,+\, \Delta_k \;\vdash\; t\, u \,:\, \theta :: \kappa'}$$

λ $$\frac{\Delta\,,\, x \,:\, \bigwedge_{i \in I} \;\theta_i \;::\; \kappa \;\vdash\; t \,:\, \theta \;::\; \kappa'}{\Delta \;\vdash\; \lambda x\,.\,t \,:\, \left(\bigwedge_{i \in I} \;\theta_i\right) \rightarrow \theta :: \kappa \rightarrow \kappa'}$$

fix $$\frac{\Gamma \vdash \mathcal{R}(F) \,:\, \theta \;::\; \kappa}{\overline{F \,:\; \theta \;::\; \kappa \;\vdash\; F \,:\, \theta \;::\; \kappa}}$$

Idea of the proof

Theorem

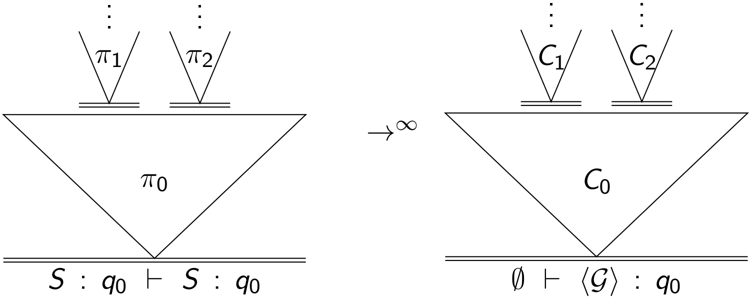
$S : q_0 \vdash S : q_0$ iff the ATA $\mathcal{A}_\phi$ has a run-tree over $\langle \mathcal{G} \rangle$.

$$\dfrac{\begin{array}{c}\pi \\ \vdots\end{array}}{S : q_0 \vdash S : q_0} \quad \longleftrightarrow \quad \dfrac{\begin{array}{c}\pi' \\ \vdots\end{array}}{\emptyset \vdash \langle \mathcal{G} \rangle : q_0} \quad \Longleftrightarrow$$

$\langle \mathcal{G} \rangle$ is accepted by $\mathcal{A}$.

- Soundness: infinitary (in fact, coinductive) subject reduction
- Completeness: build a derivation for $\mathcal{G}$ (similar to subject expansion)

Soundness



where the C_i are the tree contexts obtained by normalizing each π_i.

$C_0[C_1[], C_2[]]$ is a prefix of a run-tree of $\mathcal{A}$ over $\langle \mathcal{G} \rangle$.

Colored intersection types

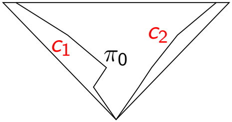
A type-system for verification

(G.-Melliès 2014, from Kobayashi-Ong 2009)

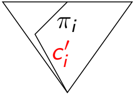
App
$$\frac{\Delta \vdash t : (\Box_{c_1}\ \theta_1\ \wedge \cdots \wedge \Box_{c_k}\ \theta_k) \rightarrow \theta\ ::\ \kappa \rightarrow \kappa' \qquad \Delta_i \vdash u\ :\ \theta_i\ ::\ \kappa}{\Delta + \Box_{c_1}\Delta_1 + \ldots + \Box_{c_k}\Delta_k\ \vdash\ t\,u\ :\ \theta :: \kappa'}$$

Subject reduction: the contraction of a redex

$$x : \Box_\epsilon\,\theta_1 \vdash x : \theta_1 \qquad x : \Box_\epsilon\,\theta_2 \vdash x : \theta_2$$



$$y : \Box_\epsilon\,\sigma_i \vdash y : \sigma_i$$

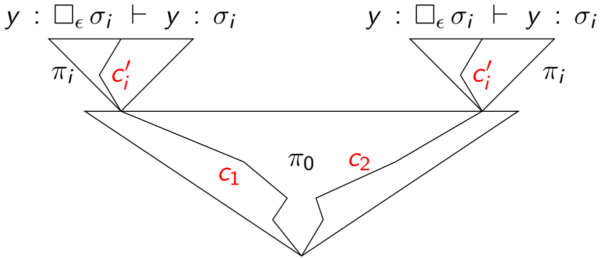


$$\frac{\dfrac{\Delta, x : \Box_{c_1}\ \theta_1\ \wedge \cdots \wedge \Box_{c_k}\ \theta_k\ \vdash\ t : \theta}{\Delta\ \vdash\ \lambda x.\, t\ :\ (\Box_{c_1}\ \theta_1\ \wedge \cdots \wedge \Box_{c_k}\ \theta_k) \rightarrow \theta} \qquad \Delta_i\ \vdash\ u : \theta_i}{\Delta + \Box_{c_1}\Delta_1 + \ldots + \Box_{c_k}\Delta_k \vdash (\lambda x.\, t)\ u : \theta}$$

A type-system for verification

App $$\frac{\Delta \vdash t : (\square_{c_1}\ \theta_1\ \wedge \cdots \wedge \square_{c_k}\ \theta_k) \rightarrow \theta\ ::\ \kappa \rightarrow \kappa' \qquad \Delta_i \vdash u\ :\ \theta_i\ ::\ \kappa}{\Delta\ +\ \square_{c_1}\Delta_1\ +\ \ldots\ +\ \square_{c_k}\Delta_k\ \vdash\ t\,u\ :\ \theta\ ::\ \kappa'}$$

gives a proof of the same sequent:



$$\Delta\ +\ \square_{c_1}\Delta_1\ +\ \ldots\ +\ \square_{c_k}\Delta_k \vdash\ t[x \leftarrow u]\ :\ \theta$$

A type system for verification

We rephrase the parity condition to typing trees, and now capture all MSO:

Theorem (G.-Melliès 2014, from Kobayashi-Ong 2009)

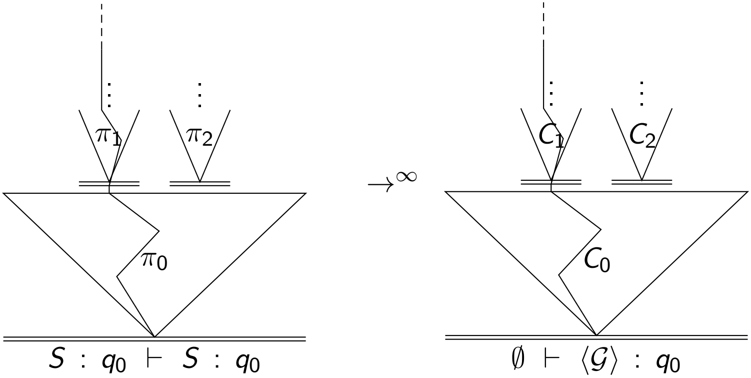
$S : q_0 \vdash S : q_0$ admits a winning typing derivation

iff

the alternating parity automaton $\mathcal{A}$ has a winning run-tree over $\langle \mathcal{G} \rangle$.

- Soundness: infinitary (in fact, coinductive) subject reduction + study of color preservation on infinite branches
- Completeness: build an optimal derivation for $\mathcal{G}$

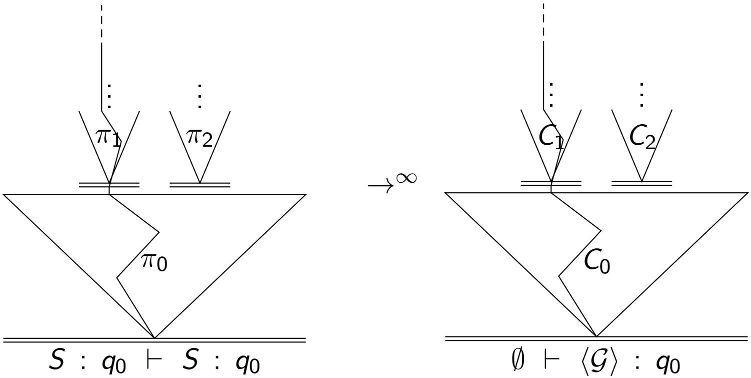
Soundness



Crucial lemma of Kobayashi-Ong (unpublished and reformulated here): every infinite branch of $\emptyset \vdash \langle \mathcal{G} \rangle : q_0$ comes from an infinite branch of $S : q_0 \vdash S : q_0$.

Consequence: derivation of $S : q_0 \vdash S : q_0$ is winning $\Longrightarrow$ the run-tree computed by subject reduction is as well.

Soundness



Reformulated in this non-idempotent setting, this lemma seems to induce:

Conjecture

There is an injection from the infinite branches of the run-tree to these of the derivation of $S : q_0 \vdash S : q_0$.

Completeness and optimality

Completeness: build a derivation for $S\ :\ q_0\ \vdash\ S\ :\ q_0$ from a run-tree (= a derivation for $\emptyset\ \vdash\ \langle \mathcal{G} \rangle\ :\ q_0$).

Main challenge: consider

$$\mathcal{G} \quad = \quad \begin{cases} S & = \quad F\ H \\ F & = \quad \lambda x.\, a\ (\, F\ x\,) \\ H & = \quad b\ H \end{cases}$$

producing

$$\langle \mathcal{G} \rangle \quad = \quad a\, a\, a \cdots$$

via finite reductions

$$S \quad \rightarrow_{\mathcal{G}}^{*} \quad a \cdots a\ F\ (\, b \cdots b\ H\,)$$

(only head reduction is optimal)

Completeness and optimality

Completeness: build a derivation for $S : q_0 \vdash S : q_0$ from a run-tree (= a derivation for $\emptyset \vdash \langle \mathcal{G} \rangle : q_0$).

Main challenge: consider

$$\mathcal{G} = \begin{cases} S & = & F\ H \\ F & = & \lambda x.\, a\ (\, F\ x\,) \\ H & = & b\ H \end{cases}$$

We need to detect that H is never contributes to $\langle \mathcal{G} \rangle$, else we can take:

$$F : \Box_0\, (\Box_0\ q_0 \to q_0)$$

and introduce a loosing branch for H.

Completeness proof $\to$ define an optimal derivation (relying on the optimality of head reduction).

Almost-sure MSO properties

Almost-sure MSO properties

In the spirit of qualitative tree languages, consider an APT $\mathcal{A}$ with the following acceptance condition:

a run-tree is almost winning

iff

the set of its branches loosing for the parity condition has measure 0.

This allows to check whether a MSO property is almost-surely satisfied.

Almost-sure MSO properties

Consider the same non-idempotent intersection type system as for MSO.

But change the winning condition accordingly.

Conjecture

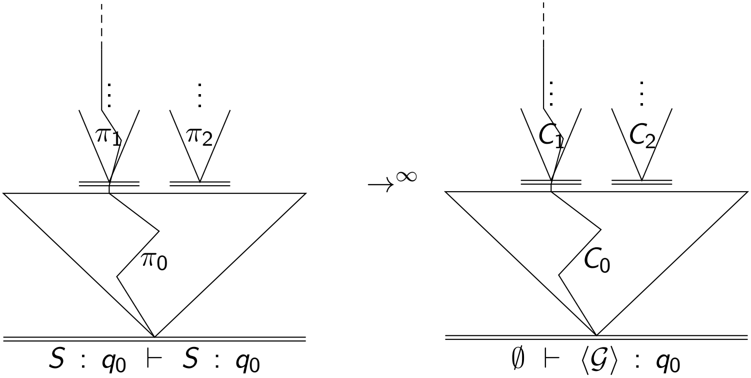
$S : q_0 \vdash S : q_0$ admits an almost winning typing derivation

iff

the APT $\mathcal{A}$ has an almost winning run-tree over $\langle \mathcal{G} \rangle$.

Almost-sure MSO properties

Soundness: reduction process which may drop branches (cf. the infinite branch injection conjecture).



So the size of the set of loosing branches decreases.

Almost-sure MSO properties

Completeness: from the proof of Kobayashi and Ong, in a non-idempotent setting:

Conjecture

There is a 1-to-1 correspondence between the infinite branches of the run-tree and these of the optimal derivation of $S : q_0 \vdash S : q_0$ built by the completeness proof.

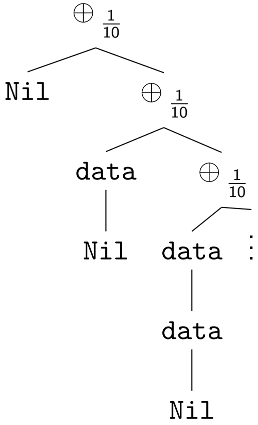
In other words: in this particular case of an optimal proof, the injection of the previous conjecture becomes a bijection.

It follows that the existence of an almost winning run-tree over $\langle \mathcal{G} \rangle$ gives an almost winning derivation of $S : q_0 \vdash S : q_0$.

Probabilistic automata

Probabilistic HOMC

```
IntList random_list() {
  IntList list = Nil;
  while(rand() > 0.1) {
    list := rand_int()::list;
  }
  return l;
}
```



Probabilistic automata

Idea: check that ϕ holds with probability $\geq p$ i.e. that it holds on a subtree of measure $\geq p$.
We extend an ATA $\mathcal{A}$ with some quantitative behavior.

Probabilistic automata (PATA):

- ATA on non-probabilistic symbols
- $+$ probabilistic behavior on choice symbol $\oplus_p$

Run-tree: labels $(q,\ p_b,\ p_f)$.

The root of a run-tree of probability p is labeled $(q_0, 1, p)$, where p is the probability with which we want the tree to satisfy the formula.

Probabilistic automata

Probabilistic behavior:

$$\bigoplus_p \quad (q,\ p_b,\ p_f)$$

$$b \qquad\qquad c$$

is labeled as

$$\bigoplus_p \quad (q,\ p_b,\ p_f)$$

$$b \quad (q,\ p \times p_b,\ p'_f) \qquad\qquad c \quad (q,\ (1-p) \times p_b,\ p_f - p'_f)$$

for some $p'_f \in [0, p_f]$ such that $p'_f \leq p \times p_b$ and $p_f - p'_f \leq (1-p) \times p_b$.

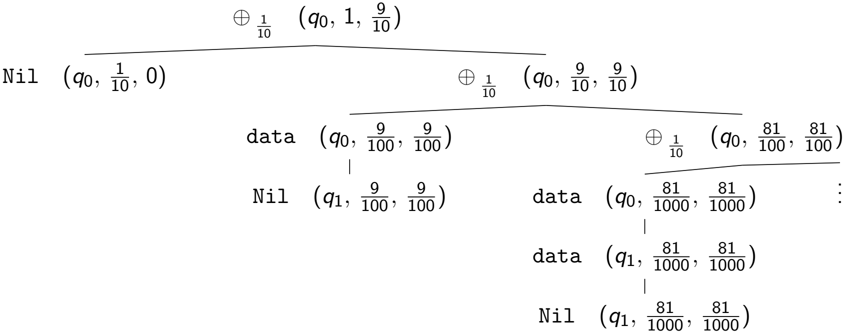
Example of PATA run

$\phi \quad = \quad$ "all the branches of the tree contain data"

is modeled by the PATA:

- $\delta_1(q_0, \texttt{data}) \; = \; (1, q_1)$,
- $\delta_1(q_1, \texttt{data}) \; = \; (1, q_1)$,
- $\delta_1(q_0, \texttt{Nil}) \; = \; \bot$,
- $\delta_1(q_1, \texttt{Nil}) \; = \; \top$.

Example of PATA run



Intersection types for PATA

As for ATA, except for tree constructors:

$$\frac{\{\,(i, q_{ij}) \mid 1 \leq i \leq n, 1 \leq j \leq k_i\}\quad \text{satisfies} \quad \delta_A(q, a)}{\emptyset \vdash a \,:\, \bigwedge_{j=1}^{k_1} (q_{1j}, p_b, p_f) \rightarrow \ldots \rightarrow \bigwedge_{j=1}^{k_n} (q_{nj}, p_b, p_f) \rightarrow (q, p_b, p_f)}$$

$$\frac{p'_f \in]0, p_f[\quad \text{and} \quad p'_f \leq p \times p_b \quad \text{and} \quad p_f - p'_f \leq (1-p) \times p_b}{\emptyset \vdash \oplus_p \,:\, (q, p \times p_b, p'_f) \rightarrow (q, (1-p) \times p_b, p_f - p'_f) \rightarrow (q, p_b, p_f)}$$

$$\frac{q \in Q \quad \text{and} \quad p \times p_b \geq p_f}{\emptyset \vdash \oplus_p \,:\, (q, p \times p_b, p_f) \rightarrow \emptyset \rightarrow (q, p_b, p_f)}$$

$$\frac{q \in Q \quad \text{and} \quad (1-p) \times p_b \geq p_f}{\emptyset \vdash \oplus_p \,:\, \emptyset \rightarrow (q, (1-p) \times p_b,\ p_f) \rightarrow (q, p_b, p_f)}$$

Intersection types for PATA

Conjecture

$$\emptyset \vdash t : (q_0, 1, p)$$

iff

the PATA $\mathcal{A}$ has a run-tree of probability p over the tree $\langle t \rangle$ generated by t (which is a term or an unfolded λY-term).

To check: that the former proof works with an infinite amount of types refining o.

Intersection types for PATA

Conjecture

$$\emptyset \vdash t \ : \ (q_0, 1, p)$$

iff

the PATA $\mathcal{A}$ has a run-tree of probability p over the tree $\langle t \rangle$ generated by t (which is a term or an unfolded λY-term).

Under connection Rel/non-idempotent types, we obtain a similar denotational theorem.

Note that $[\![o]\!] \ = \ Q \times [0,1] \times [0,1]$.

Automata are counter-programs with effects

Grellois-Melliès, CSL 2015:

With a linear logic point of view: HOMC is a dual process between

a program: the recursion scheme $\mathcal{G}$,

and

a counter-program with (co)effects: the APT $\mathcal{A}$.

Tensorial logic with effects and PATA

Tensorial logic

- A refinement of linear logic
- A logic of tensor, sum and negation where $A \ncong \neg\neg A$
- Purpose: conciliate linear logic with algebraic effects
- Deeply related to game semantics: it is the syntax of dialogue games...
- ...and more generally related to dialogue categories

Tensorial logic with effects (Melliès) connects with semantics (dialogue categories with effects)

States in tensorial logic

$$Lookup \quad \frac{\Gamma \vdash \bot \qquad \cdots \qquad \Gamma \vdash \bot}{\Gamma \vdash \bot}$$

$$Update_{val} \quad \frac{\Gamma \vdash \bot}{\Gamma \vdash \bot}$$

and equations such as

$$\frac{\begin{array}{c}\pi \\ \vdots\end{array}}{\Gamma \vdash \bot} \quad = \quad \begin{array}{c}Update_{val_1} \\ Lookup\end{array} \frac{\frac{\begin{array}{c}\pi \\ \vdots \\ \Gamma \vdash \bot\end{array}}{\Gamma \vdash \bot} \qquad \cdots \qquad \frac{\begin{array}{c}\pi \\ \vdots \\ \Gamma \vdash \bot\end{array}}{\Gamma \vdash \bot} Update_{val_n}}{\Gamma \vdash \bot}$$

Tensorial logic and PATA

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash t_1 : \bot}{\Gamma \vdash t_1 : \bot}\ \mathit{Update}_{q_0, p \times p_b, p_f'} \qquad \dfrac{\Gamma \vdash t_2 : \bot}{\Gamma \vdash t_2 : \bot}\ \mathit{Update}_{q_0, (1-p) \times p_b, p_f - p_f'}}{\Gamma \vdash \oplus_p\ t_1\ t_2 : \bot}}{\dfrac{\Gamma \vdash \oplus_p\ t_1\ t_2 : \bot \quad \mathit{Choice}_{p_f'} \qquad \cdots}{\Gamma \vdash \oplus_p\ t_1\ t_2 : \bot}\ \mathit{Lookup}_{p_b, p_f}}$$

where $\oplus_p : \bot \multimap \bot \multimap \bot \ \in \ \Gamma$

Fundamental idea: the state of the automaton is a state in the sense of the state monad. Non-determinism is handled by a monadic effect as well.

Tensorial logic and PATA

$$\delta(a, q_0) \;=\; (1, q_0) \wedge (1, q_1) \qquad \delta(a, q_1) \;=\; \bot$$

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash t \,:\, \bot}{\Gamma \vdash t \,:\, \bot}\;\textit{Update}_{q_0,p_b,p_f} \qquad \dfrac{\Gamma \vdash t \,:\, \bot}{\Gamma \vdash t \,:\, \bot}\;\textit{Update}_{q_1,p_b,p_f}}{\Gamma \vdash t \,:\, !\bot}\;\textit{Promotion} \qquad \dfrac{\textit{fail}}{\Gamma \vdash a\ t \,:\, \bot}}{\Gamma \vdash a\ t \,:\, \bot}\;\textit{Lookup}_{p_b,p_f}$$

where $a \,:\, !\bot \multimap \bot \;\in\; \Gamma$

Exceptions when δ is not defined.

Automata are counter-programs with effects

What's next

- A non-idempotent variant of Kobayashi-Ong's result (in a coinductive way?)
- Find a less naive type system
- Connection with denotational models: Rel, dialogue categories with effects