A Finite Semantics of Simply-Typed Lambda Terms for Infinite Runs of Automata

Klaus Aehlig

Finitely Presented Infinite structures

- Abstract interpretation of Programs;
  consider e.g.

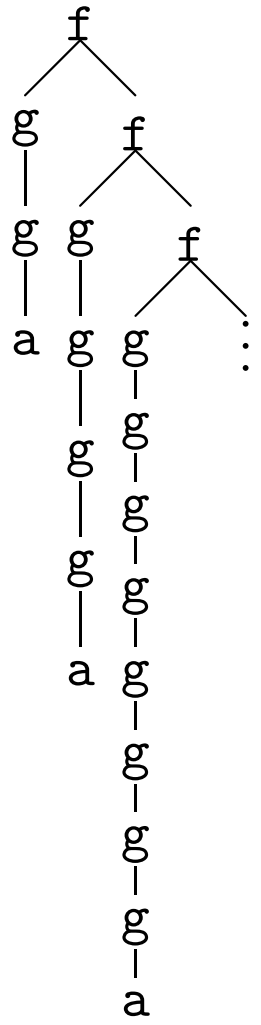
```
void foo(string φ)
  ...foo(φ ++ φ)...

int main()
  foo("gg")
```

  `foo` always called with even length strings

- Letters $\mathtt{g}^{\iota\to\iota}$ as "string transformers";
  event $\mathtt{f}^{\iota\to\iota\to\iota}$; read $\mathtt{f}sc$ as "$\mathtt{foo}(s)$ called, then c"

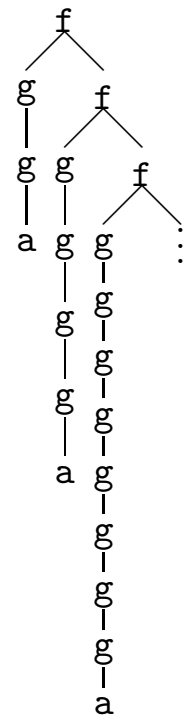
$\rightsquigarrow$ infinite tree over some first-order signature



Recursion Schemes

```
void foo(string φ)
  ...foo(φ ++ φ)...

int main()
  foo("gg")
```



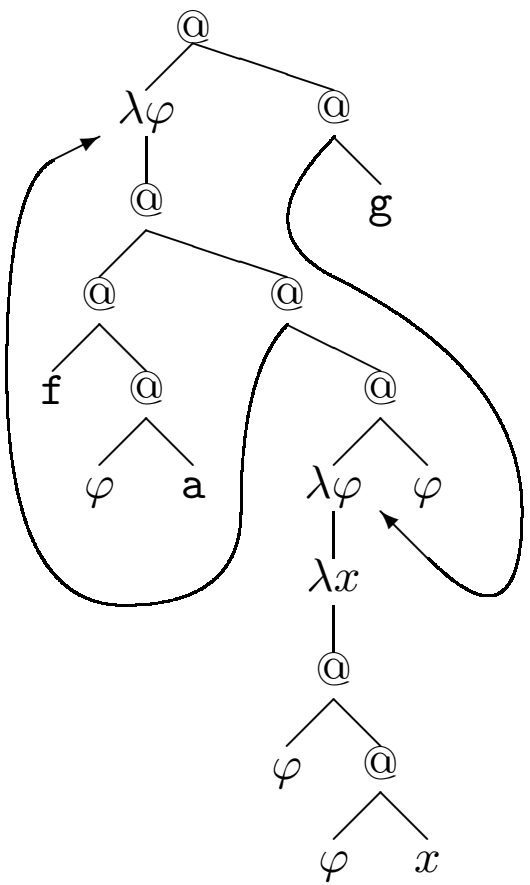
- Represent procedures/functions by typed symbols

  ... gives the following recursion scheme

$$\begin{aligned} S &= F(W\mathtt{g}) \\ W\varphi x &= \varphi(\varphi x) \\ F\varphi &= \mathtt{f}(\varphi\mathtt{a})(F(W\varphi)) \end{aligned}$$

- A recursion scheme consists of
  - typed non-terminals $S, F, \ldots$
  - first order terminals $\mathtt{f}, \mathtt{g}, \ldots$
  - equations $F\overrightarrow{x} = e$

$$
\begin{array}{lcl}
S & = & F(W\mathtt{g}) \\
W\varphi x & = & \varphi(\varphi x) \\
F\varphi & = & \mathtt{f}(\varphi\mathtt{a})(F(W\varphi))
\end{array}
$$



Infinite Lambda-Trees

- A recursion scheme equation

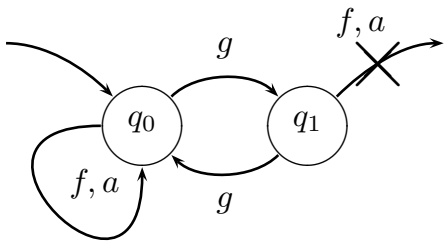
$$F\overrightarrow{x} = e$$

  has only variables on the left!

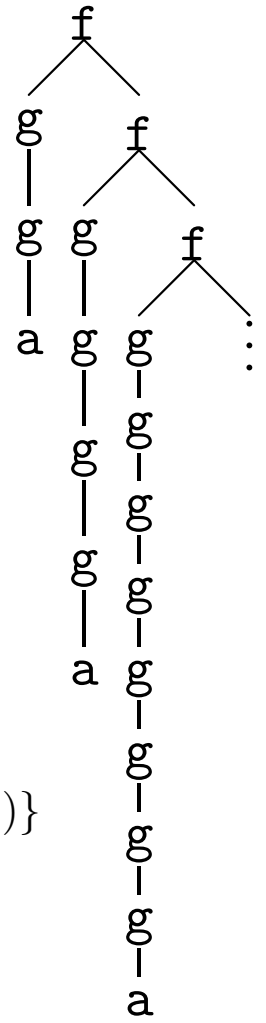
- read $F\overrightarrow{x} = e$ as $F = \lambda\overrightarrow{x}.e$
- Unfolding them gives an infinite, *but regular*, lambda-tree

Automata

- Want to decide properties of these infinite trees
- Typically described by MSO formulae
  . . . or, equivalently, by appropriate tree automata
- here: just non-deterministic tree automata with *trivial* acceptance condition
  i.e., we just ask for an infinite run



$$\delta(\mathtt{f}, q_0) = \{(q_0, q_0)\} \quad \delta(\mathtt{f}, q_1) = \emptyset$$
$$\delta(\mathtt{g}, q_0) = \{(q_1, *)\} \quad \delta(\mathtt{g}, q_1) = \{(q_0, *)\}$$
$$\delta(\mathtt{a}, q_0) = \{(*, *)\} \quad \delta(\mathtt{a}, q_1) = \emptyset$$



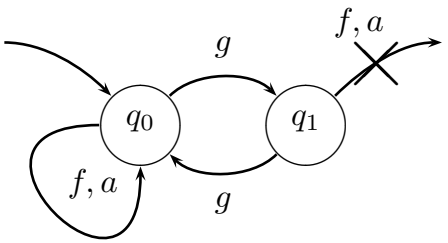
Related Work

- Knapik, Niwinski, Urzyczyn (Fossacs '02)
  MSO theory of "safe" recursion schemes is decidable
- Aehlig, de Miranda, Ong (TLCA '05) *and*
  Knapik, Niwinski, Urzyczyn, Walukiewicz (ICALP '05)
  MSO theory of level-2 recursion schemes is decidable
- Ong (LICS '06)
  MSO theory of all recursion schemes is decidable
  . . . technically involved proof based, on game semantics
- Here: weaker result (less properties), but
  - Argument conceptually simple
  - Finite, automata-based semantics for the simple types
    . . . that might be useful elsewhere

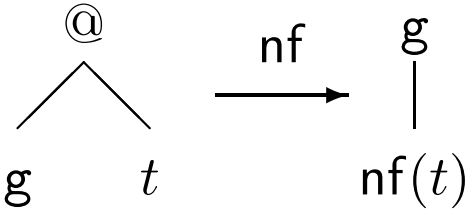
Semantics for Simple Types

- recall that all terms are typed; $\rho, \sigma \ ::= \ \iota \mid \rho \to \sigma$
- Can define a set-theoretic semantics by $[\![\rho \to \sigma]\!] = {}^{[\![\rho]\!]}[\![\sigma]\!]$
  free parameter: $[\![\iota]\!]$
- Want a semantics "as seen by $\mathfrak{A}$"
  $\rightsquigarrow$ *set* $[\![\iota]\!] = \mathfrak{P}(Q)$*; "these states have an infinite run"*
- All $[\![\rho]\!]$ are finite sets!
  $\rightsquigarrow$ *Can decide whether appropriate annotations exist.*

Semantics for Simple Types (cont'd)



- At base type *"These states have an infinite run"*
- At type $\rho \to \sigma$ a value f for t says
  "If s has semantics a, then ts has semantics $f(a)$"
- E.g., g can have semantics swap
  $\text{swap}(\{q_0\}) = \{q_1\}$, $\text{swap}(\{q_1\}) = \{q_0\}$,
  and, of course, $\text{swap}(\emptyset) = \emptyset$, $\text{swap}(Q) = Q$.



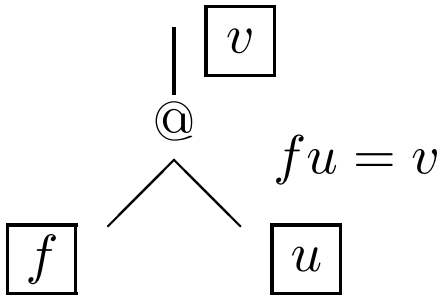
- Having something better suffices...
  $\subset$ induces pointwise order $\sqsubseteq$ on $[\![\tau]\!]$

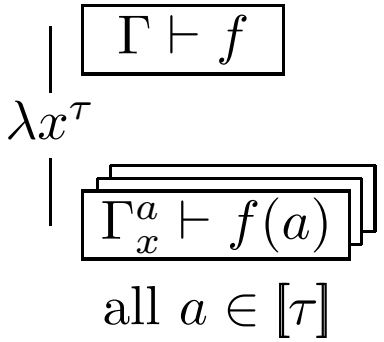
Semantical Annotations

- Annotate lambda-tree;
  nodes of type τ get values from $[\![\tau]\!]$
- $[\![\tau]\!]$ is finite; only finitely many nodes

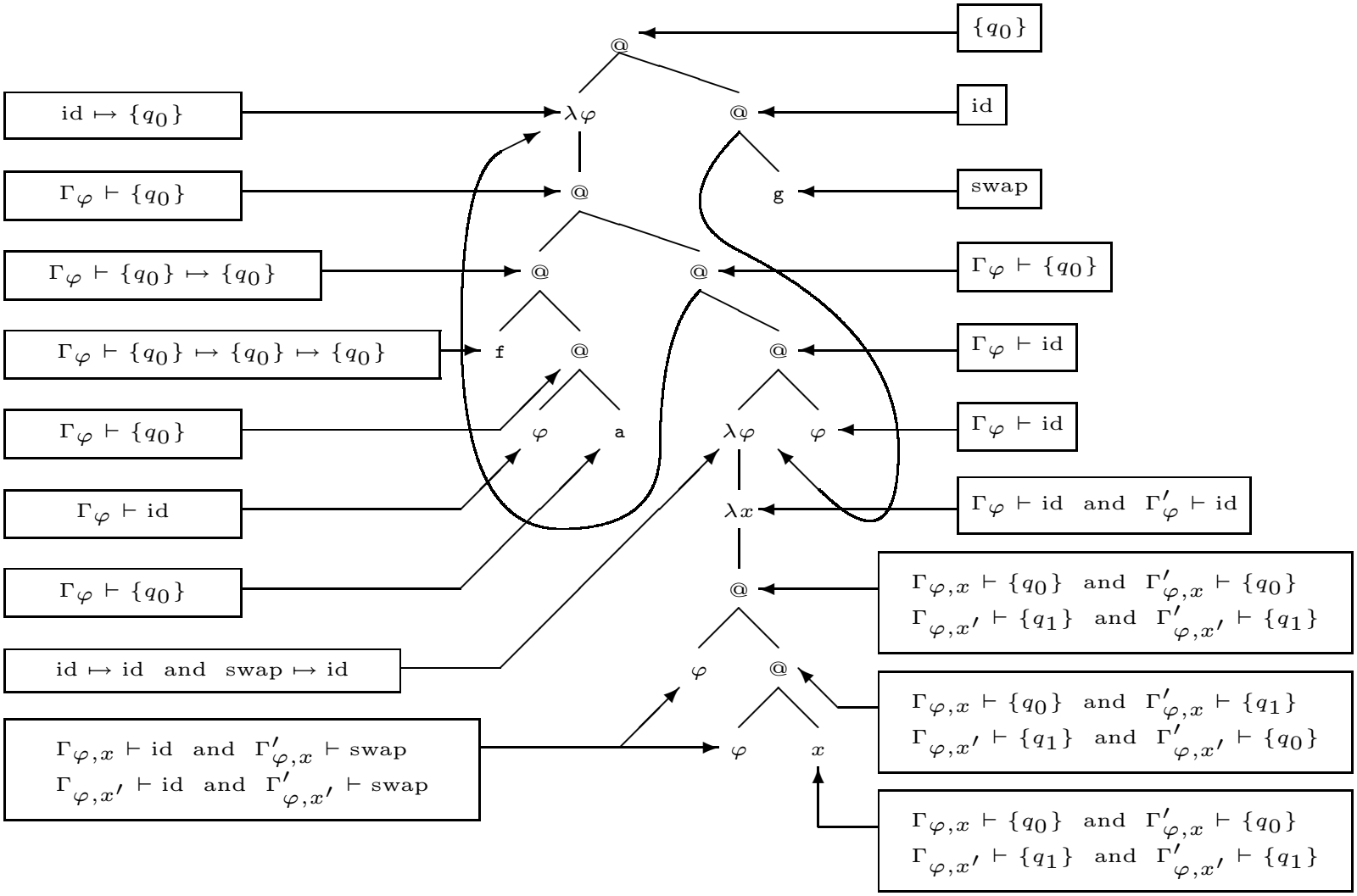
$\therefore$ Only finitely many possibilities
$\rightsquigarrow$ decidable!

- Automata move local
  $\rightsquigarrow$ *local consistency conditions*
- Environments for free variables
- Infinite runs via the backwards arcs
  (in the lambda-tree) — well, almost...
  $\rightsquigarrow$ Use continuous normalisation to ensure that
  infinite *computations* produce infinite *value* trees!





A Proof


$\{q_0\}$
@
id $\mapsto \{q_0\}$
$\lambda\varphi$
@
id
$\Gamma_\varphi \vdash \{q_0\}$
@
g
swap
$\Gamma_\varphi \vdash \{q_0\} \mapsto \{q_0\}$
@
@
$\Gamma_\varphi \vdash \{q_0\}$
$\Gamma_\varphi \vdash \{q_0\} \mapsto \{q_0\} \mapsto \{q_0\}$
f
@
@
$\Gamma_\varphi \vdash$ id
$\Gamma_\varphi \vdash \{q_0\}$
φ
a
$\lambda\varphi$
φ
$\Gamma_\varphi \vdash$ id
$\Gamma_\varphi \vdash$ id
λx
$\Gamma_\varphi \vdash$ id and $\Gamma'_\varphi \vdash$ id
$\Gamma_\varphi \vdash \{q_0\}$
@
$\Gamma_{\varphi,x} \vdash \{q_0\}$ and $\Gamma'_{\varphi,x} \vdash \{q_0\}$
$\Gamma_{\varphi,x'} \vdash \{q_1\}$ and $\Gamma'_{\varphi,x'} \vdash \{q_1\}$
id $\mapsto$ id and swap $\mapsto$ id
φ
@
$\Gamma_{\varphi,x} \vdash \{q_0\}$ and $\Gamma'_{\varphi,x} \vdash \{q_1\}$
$\Gamma_{\varphi,x'} \vdash \{q_1\}$ and $\Gamma'_{\varphi,x'} \vdash \{q_0\}$
$\Gamma_{\varphi,x} \vdash$ id and $\Gamma'_{\varphi,x} \vdash$ swap
$\Gamma_{\varphi,x'} \vdash$ id and $\Gamma'_{\varphi,x'} \vdash$ swap
φ
x
$\Gamma_{\varphi,x} \vdash \{q_0\}$ and $\Gamma'_{\varphi,x} \vdash \{q_0\}$
$\Gamma_{\varphi,x'} \vdash \{q_1\}$ and $\Gamma'_{\varphi,x'} \vdash \{q_1\}$

Completeness

- Are these annotations enough to prove every infinite run?
- Define a canonical $\langle\!\langle t \rangle\!\rangle$ semantics of every lambda tree $t^{\overrightarrow{\rho}\to\iota}$

$$\langle\!\langle t \rangle\!\rangle(\overrightarrow{a}) = \{q \mid \exists \overrightarrow{s}.\ \langle\!\langle \overrightarrow{s} \rangle\!\rangle \sqsubseteq \overrightarrow{a} \wedge \mathfrak{A}, q \models^{\infty} \mathsf{nf}(t\overrightarrow{s})\}$$

- show $\langle\!\langle st \rangle\!\rangle \sqsubseteq \langle\!\langle s \rangle\!\rangle\langle\!\langle t \rangle\!\rangle$ and $\langle\!\langle \lambda x t \rangle\!\rangle^{\Gamma}(a) \sqsubseteq \langle\!\langle t \rangle\!\rangle^{\Gamma^a_x}$

$\therefore$ $\langle\!\langle t \rangle\!\rangle$ is always a valid annotation for t
... and $\langle\!\langle t^{\iota} \rangle\!\rangle = \{q \mid \mathfrak{A}, q \models^{\infty} \mathsf{nf}t\}$