

Guarded Kleene Algebra with Tests

Verification of Uninterpreted Programs in Nearly Linear Time

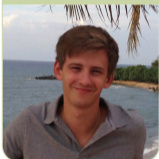
Tobias Kappé

Cornell University

Algebra|Coalgebra Seminar

Joint work with. . .

Steffen Smolka



Nate Foster



Justin Hsu



Dexter Kozen



Alexandra Silva



Published @ POPL 2020; see arxiv.org/abs/1907.05920.

```
while a and b do
  e;
end
while a do
  f;
  while a and b do
    e;
  end
end
```

Introduction

```
while a and b do
  e;
end
while a do
  f;
  while a and b do
    e;
  end
end
end
```

```
while a do
  if b then
    e;
  else
    f;
  end
end
```

Introduction

```
while a and b do
  e;
end
while a do
  f;
  while a and b do
    e;
  end
end
end
```

$\stackrel{?}{\equiv}$

```
while a do
  if b then
    e;
  else
    f;
  end
end
end
```

Contributions:

- Nearly linear time decision procedure for equivalence.¹

¹For fixed number of tests.

Contributions:

- Nearly linear time decision procedure for equivalence.¹
- Axiomatization of uninterpreted program equivalence.

¹For fixed number of tests.

Contributions:

- Nearly linear time decision procedure for equivalence.¹
- Axiomatization of uninterpreted program equivalence.
- Kleene theorem for uninterpreted programs.

¹For fixed number of tests.

$$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$$
$$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} +_{\mathbf{a}} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$$

$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$

a or b

$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} + \mathbf{a} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$

$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$

a and **b**

$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} +_{\mathbf{a}} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$

$\mathbf{a}, \mathbf{b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$

not \mathbf{a}

$\mathbf{e}, \mathbf{f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} +_{\mathbf{a}} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$

$\mathbf{a}, \mathbf{b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$

false

$\mathbf{e}, \mathbf{f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} +_{\mathbf{a}} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$

$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$

true

$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} + \mathbf{a} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$

$$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$$
$$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} +_{\mathbf{a}} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$$

assert \mathbf{a}

$$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$$
$$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} +_a \mathbf{f} \mid \mathbf{e}^{(a)}$$


$\mathbf{e; f}$

$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$

$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} +_{\mathbf{a}} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$

if \mathbf{a} then \mathbf{e} else \mathbf{f}

$\mathbf{a, b} ::= t \in T \mid \mathbf{a} + \mathbf{b} \mid \mathbf{ab} \mid \bar{\mathbf{a}} \mid 0 \mid 1$

$\mathbf{e, f} ::= \mathbf{a} \mid p \in \Sigma \mid \mathbf{ef} \mid \mathbf{e} + \mathbf{a} \mathbf{f} \mid \mathbf{e}^{(\mathbf{a})}$

while \mathbf{a} do \mathbf{e}

Syntax

```
while a do
  if b then
    e;
  else
    f;
  end
end
```

⇓

$(\mathbf{e} + \mathbf{b} \mathbf{f})^{(\mathbf{a})}$

```
while a and b do
  e;
end
while a do
  f;
  while a and b do
    e;
  end
end
```

⇓

$\mathbf{e}^{(\mathbf{a}\mathbf{b})} (\mathbf{f}\mathbf{e}^{(\mathbf{a}\mathbf{b})})^{(\mathbf{a})}$

$$\text{sat} : T \rightarrow 2^{\text{States}}$$

$$\text{sat} : T \rightarrow 2^{\text{States}}$$

$$\text{eval} : \Sigma \rightarrow \text{States} \rightarrow 2^{\text{States}}$$

$$sat : T \rightarrow 2^{States}$$

$$eval : \Sigma \rightarrow States \rightarrow 2^{States}$$

$$i = (sat, eval)$$

$$sat : T \rightarrow 2^{States}$$

$$eval : \Sigma \rightarrow States \rightarrow 2^{States}$$

$$i = (sat, eval)$$

$$\mathcal{R}_i[[\mathbf{e}]] : States \rightarrow 2^{States}$$

$$\text{sat} : T \rightarrow 2^{\text{States}}$$

$$\text{sat} : T \rightarrow 2^{\text{States}}$$

$$\text{eval} : \Sigma \rightarrow \text{States} \rightarrow \mathcal{D}(\text{States})$$

$$sat : T \rightarrow 2^{States}$$

$$eval : \Sigma \rightarrow States \rightarrow \mathcal{D}(States)$$

$$i = (sat, eval)$$

$$sat : T \rightarrow 2^{States}$$

$$eval : \Sigma \rightarrow States \rightarrow \mathcal{D}(States)$$

$$i = (sat, eval)$$

$$\mathcal{P}_i[\mathbf{e}] : States \rightarrow \mathcal{D}(States)$$

$$Atoms = 2^T$$

$$GS(\Sigma, T) = Atoms \cdot (\Sigma \cdot Atoms)^*$$

$$Atoms = 2^T$$

$$GS(\Sigma, T) = Atoms \cdot (\Sigma \cdot Atoms)^*$$

$$L \diamond K = \{w\alpha x : w\alpha \in L, \alpha x \in K\}$$

$$L^{(n)} = \underbrace{L \diamond \dots \diamond L}_{n \text{ times}}$$

$$L^{(*)} = \bigcup_{n \in \mathbb{N}} L^{(n)}$$

e	$\llbracket e \rrbracket$
$t \in T$	$\{\alpha \in Atoms : t \in \alpha\}$
$a + b$	$\llbracket a \rrbracket \cup \llbracket b \rrbracket$
ab	$\llbracket a \rrbracket \cap \llbracket b \rrbracket$
\bar{a}	$Atoms \setminus \llbracket a \rrbracket$
$p \in \Sigma$	$\{\alpha p \beta : \alpha, \beta \in Atoms\}$
$e +_a f$	$\llbracket a \rrbracket \diamond \llbracket e \rrbracket \cup \llbracket \bar{a} \rrbracket \diamond \llbracket f \rrbracket$
ef	$\llbracket e \rrbracket \diamond \llbracket f \rrbracket$
$e^{(a)}$	$(\llbracket a \rrbracket \diamond \llbracket e \rrbracket)^{(*)} \diamond \llbracket \bar{a} \rrbracket$

Theorem

The following are equivalent:

$$\llbracket \mathbf{e} \rrbracket = \llbracket \mathbf{f} \rrbracket$$

$$\forall i. \mathcal{R}_i \llbracket \mathbf{e} \rrbracket = \mathcal{R}_i \llbracket \mathbf{f} \rrbracket$$

$$\forall i. \mathcal{P}_i \llbracket \mathbf{e} \rrbracket = \mathcal{P}_i \llbracket \mathbf{f} \rrbracket$$

Theorem

The following are equivalent:

$$\llbracket \mathbf{e} \rrbracket = \llbracket \mathbf{f} \rrbracket$$

$$\forall i. \mathcal{R}_i \llbracket \mathbf{e} \rrbracket = \mathcal{R}_i \llbracket \mathbf{f} \rrbracket$$

$$\forall i. \mathcal{P}_i \llbracket \mathbf{e} \rrbracket = \mathcal{P}_i \llbracket \mathbf{f} \rrbracket$$

How to check $\llbracket \mathbf{e} \rrbracket = \llbracket \mathbf{f} \rrbracket$:

- 1 Create automata that accept $\llbracket \mathbf{e} \rrbracket$ and $\llbracket \mathbf{f} \rrbracket$.
- 2 Check automata for bisimilarity.

Axiomatization / without loops

$$e +_a e \equiv e$$

Axiomatization / without loops

$$e +_a e \equiv e \qquad e +_a f \equiv f +_{\bar{a}} e$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\bar{a}a \equiv 0$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\bar{a}a \equiv 0$$

$$0e \equiv 0$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\bar{a}a \equiv 0$$

$$0e \equiv 0$$

Example

if a then e else assert false = $e +_a 0$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv a e +_a f$$

$$\bar{a} a \equiv 0$$

$$0 e \equiv 0$$

Example

$$\text{if } a \text{ then } e \text{ else assert false} = e +_a 0 \equiv a e +_a 0$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\bar{a}a \equiv 0$$

$$0e \equiv 0$$

Example

$$\begin{aligned} \text{if } a \text{ then } e \text{ else assert false} &= e +_a 0 \equiv ae +_a 0 \\ &\equiv 0 +_{\bar{a}} ae \end{aligned}$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\bar{a}a \equiv 0$$

$$0e \equiv 0$$

Example

$$\begin{aligned} \text{if } a \text{ then } e \text{ else assert false} &= e +_a 0 \equiv ae +_a 0 \\ &\equiv 0 +_{\bar{a}} ae \\ &\equiv 0e +_{\bar{a}} ae \end{aligned}$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\boxed{\bar{a}a \equiv 0}$$

$$0e \equiv 0$$

Example

$$\begin{aligned} \text{if } a \text{ then } e \text{ else assert false} &= e +_a 0 \equiv ae +_a 0 \\ &\equiv 0 +_{\bar{a}} ae \\ &\equiv 0e +_{\bar{a}} ae \\ &\equiv \bar{a}ae +_{\bar{a}} ae \end{aligned}$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\bar{a}a \equiv 0$$

$$0e \equiv 0$$

Example

$$\begin{aligned} \text{if } a \text{ then } e \text{ else assert false} &= e +_a 0 \equiv ae +_a 0 \\ &\equiv 0 +_{\bar{a}} ae \\ &\equiv 0e +_{\bar{a}} ae \\ &\equiv \bar{a}ae +_{\bar{a}} ae \\ &\equiv ae +_{\bar{a}} ae \end{aligned}$$

Axiomatization / without loops

$$e +_a e \equiv e$$

$$e +_a f \equiv f +_{\bar{a}} e$$

$$e +_a f \equiv ae +_a f$$

$$\bar{a}a \equiv 0$$

$$0e \equiv 0$$

Example

$$\begin{aligned} \text{if } a \text{ then } e \text{ else assert false} &= e +_a 0 \equiv ae +_a 0 \\ &\equiv 0 +_{\bar{a}} ae \\ &\equiv 0e +_{\bar{a}} ae \\ &\equiv \bar{a}ae +_{\bar{a}} ae \\ &\equiv ae +_{\bar{a}} ae \\ &\equiv ae &= \text{assert } a; e \end{aligned}$$

$$\frac{e \equiv fe + a g}{e \equiv f^{(a)}g}$$

$$\frac{e \equiv fe + a g}{e \equiv f^{(a)}g}$$

Allows to derive $1 \equiv 1^{(1)}$, i.e.,

`assert true` \equiv `while true do assert true`

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Lemma

For every e , there exists a productive \hat{e} such that $e^{(b)} \equiv \hat{e}^{(b)}$; also, $\widehat{ae} \equiv a\hat{e}$.

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Lemma

For every e , there exists a productive \hat{e} such that $e^{(b)} \equiv \hat{e}^{(b)}$; also, $\widehat{ae} \equiv a\hat{e}$.

Lemma

$$e^{(a)}$$

$$(ae)^{(a)}$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Lemma

For every e , there exists a productive \hat{e} such that $e^{(b)} \equiv \hat{e}^{(b)}$; also, $\widehat{ae} \equiv a\hat{e}$.

Lemma

$$e^{(a)} \equiv \hat{e}^{(a)}$$

$$(ae)^{(a)}$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Lemma

For every e , there exists a productive \hat{e} such that $e^{(b)} \equiv \hat{e}^{(b)}$; also, $\widehat{ae} \equiv a\hat{e}$.

Lemma

$$e^{(a)} \equiv \hat{e}^{(a)} \equiv \hat{e}\hat{e}^{(a)} +_a 1$$

$$(ae)^{(a)}$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Lemma

For every e , there exists a productive \hat{e} such that $e^{(b)} \equiv \hat{e}^{(b)}$; also, $\widehat{ae} \equiv a\hat{e}$.

Lemma

$$e^{(a)} \equiv \hat{e}^{(a)} \equiv \hat{e}\hat{e}^{(a)} +_a 1 \equiv a\hat{e}\hat{e}^{(a)} +_a 1 \quad (ae)^{(a)}$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Lemma

For every e , there exists a productive \hat{e} such that $e^{(b)} \equiv \hat{e}^{(b)}$; also, $\widehat{ae} \equiv a\hat{e}$.

Lemma

$$e^{(a)} \equiv \hat{e}^{(a)} \equiv \hat{e}\hat{e}^{(a)} +_a 1 \equiv a\hat{e}\hat{e}^{(a)} +_a 1 \equiv \widehat{ae}\hat{e}^{(a)} +_a 1 \quad (ae)^{(a)}$$

Axiomatization / with loops

$$\frac{e \equiv fe +_a g \quad f \text{ is productive}}{e \equiv f^{(a)}g}$$

$$e^{(a)} \equiv ee^a +_a 1$$

$$(e +_a 1)^{(b)} \equiv (ae)^{(b)}$$

Lemma

For every e , there exists a productive \hat{e} such that $e^{(b)} \equiv \hat{e}^{(b)}$; also, $\widehat{ae} \equiv a\hat{e}$.

Lemma

$$e^{(a)} \equiv \hat{e}^{(a)} \equiv \hat{e}\hat{e}^{(a)} +_a 1 \equiv a\hat{e}\hat{e}^{(a)} +_a 1 \equiv \widehat{ae}\hat{e}^{(a)} +_a 1 \equiv (\widehat{ae})^{(a)} \equiv (ae)^{(a)}$$

Axiomatization / soundness & completeness

Theorem (Soundness)

If $e \equiv f$, then $\llbracket e \rrbracket = \llbracket f \rrbracket$.

Theorem (Soundness)

If $e \equiv f$, then $\llbracket e \rrbracket = \llbracket f \rrbracket$.

How about the converse?

- 1 $A \mapsto S(A)$ and $e \mapsto A_e$ with $e \equiv S(A_e)$.
- 2 If $A \sim A'$, then $S(A) \equiv S(A')$.

Theorem (Soundness)

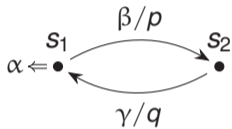
If $\mathbf{e} \equiv \mathbf{f}$, then $\llbracket \mathbf{e} \rrbracket = \llbracket \mathbf{f} \rrbracket$.

How about the converse?

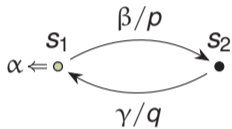
- 1 $A \mapsto S(A)$ and $\mathbf{e} \mapsto A_{\mathbf{e}}$ with $\mathbf{e} \equiv S(A_{\mathbf{e}})$.
- 2 If $A \sim A'$, then $S(A) \equiv S(A')$.

$$\begin{aligned}\llbracket \mathbf{e} \rrbracket = \llbracket \mathbf{f} \rrbracket &\implies L(A_{\mathbf{e}}) = L(A_{\mathbf{f}}) \\ &\implies A_{\mathbf{e}} \sim A_{\mathbf{f}} \\ &\implies S(A_{\mathbf{e}}) \equiv S(A_{\mathbf{f}}) \\ &\implies \mathbf{e} \equiv \mathbf{f}\end{aligned}$$

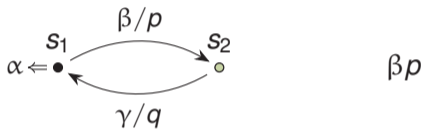
A Kleene theorem / automata model



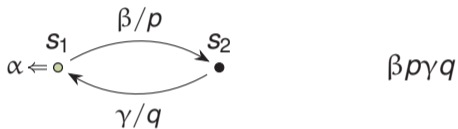
A Kleene theorem / automata model



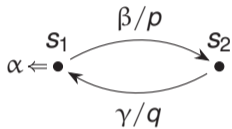
A Kleene theorem / automata model



A Kleene theorem / automata model

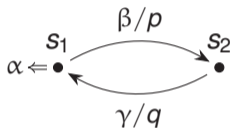


A Kleene theorem / automata model



$$\beta p \gamma q \alpha \in L(A)$$

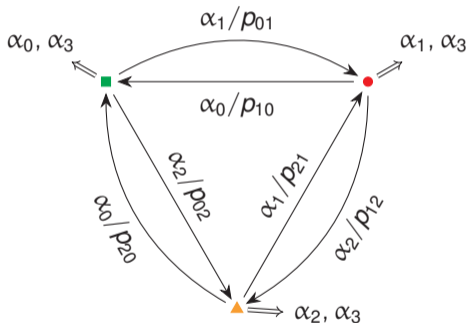
A Kleene theorem / automata model



$$\beta p \gamma q \alpha \in L(A)$$

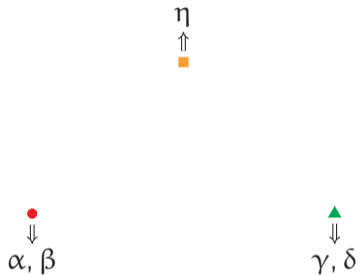
$$(X, \delta : X \rightarrow (2 + \Sigma \times X)^{Atoms})$$

Not described by an expression e :

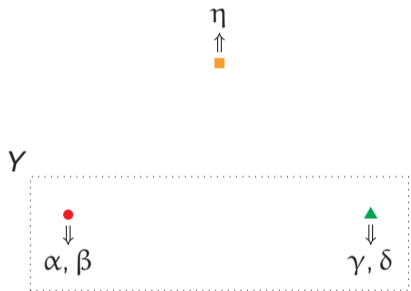


See [Kozen and Tseng 2008].

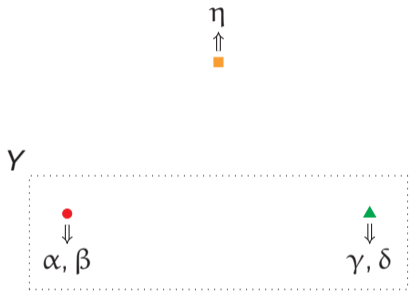
A Kleene theorem / automata model



A Kleene theorem / automata model



A Kleene theorem / automata model



$$h : Atoms \rightarrow 2 + \Sigma \times X$$

$$h(\alpha) = (p, \blacksquare)$$

$$h(\beta) = 0$$

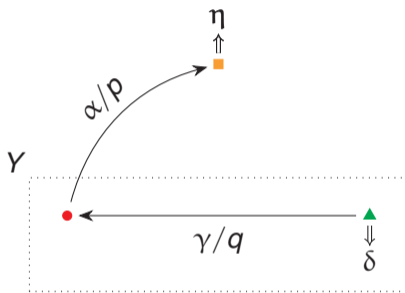
$$h(\gamma) = (q, \bullet)$$

$$h(\delta) = 1$$

$$h(-) = 0$$

NB: This slide had the wrong signature for h when I gave the talk; this is fixed now.

A Kleene theorem / automata model



$$h : Atoms \rightarrow 2 + \Sigma \times X$$

$$h(\alpha) = (p, \blacksquare)$$

$$h(\beta) = 0$$

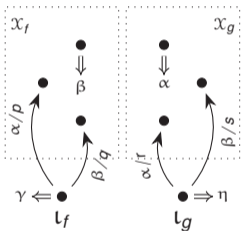
$$h(\gamma) = (q, \bullet)$$

$$h(\delta) = 1$$

$$h(-) = 0$$

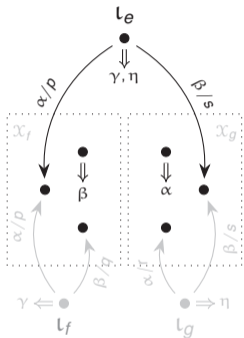
NB: This slide had the wrong signature for h when I gave the talk; this is fixed now.

A Kleene theorem / expressions to automata



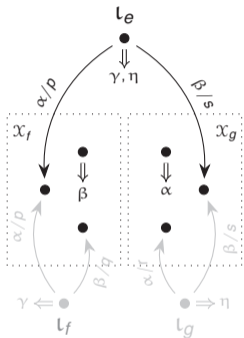
$$e = f + a g$$

A Kleene theorem / expressions to automata

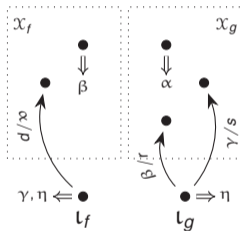


$$e = f + a g$$

A Kleene theorem / expressions to automata

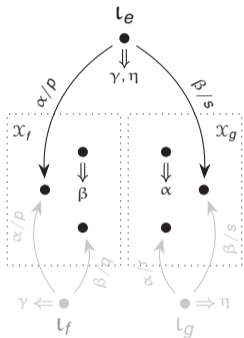


$$e = f + a g$$

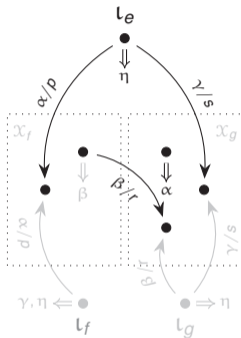


$$e = fg$$

A Kleene theorem / expressions to automata

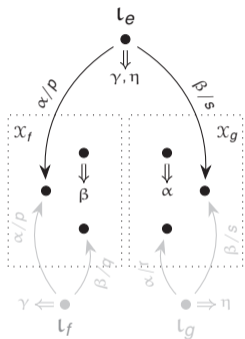


$$e = f + a g$$

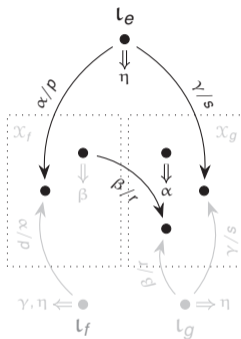


$$e = fg$$

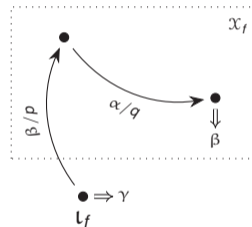
A Kleene theorem / expressions to automata



$$e = f + a g$$

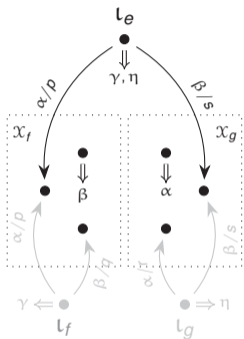


$$e = f g$$

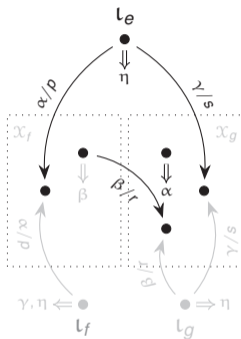


$$e = f(a)$$

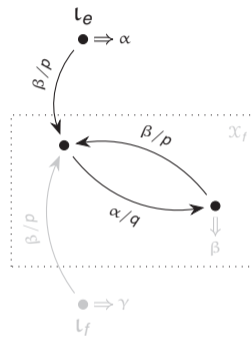
A Kleene theorem / expressions to automata



$$e = f + a g$$

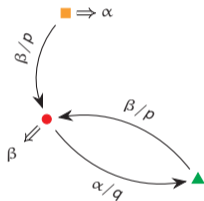


$$e = f g$$

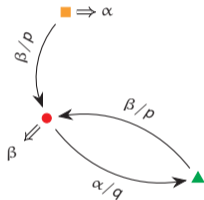


$$e = f(a)$$

A Kleene theorem / automata to expressions



A Kleene theorem / automata to expressions

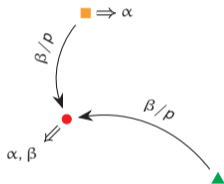


$$x_{\blacksquare} \equiv 1 \quad +_{\alpha} p \cdot x_{\bullet} \quad +_{\beta} 0$$

$$x_{\bullet} \equiv q \cdot x_{\blacktriangle} \quad +_{\alpha} 1 \quad +_{\beta} 0$$

$$x_{\blacktriangle} \equiv 0 \quad +_{\alpha} p \cdot x_{\bullet} \quad +_{\beta} 0$$

A Kleene theorem / automata to expressions



$$x_{\square} \equiv 1 \quad +_{\alpha} p \cdot x_{\bullet} \quad +_{\beta} 0$$

$$x_{\bullet} \equiv 1 \quad +_{\alpha} 1 \quad +_{\beta} 0$$

$$x_{\blacktriangle} \equiv 0 \quad +_{\alpha} p \cdot x_{\bullet} \quad +_{\beta} 0$$

A Kleene theorem / main result

Theorem

Let $L \subseteq \text{GS}(\Sigma, T)$. The following are equivalent:

- 1 $L = \llbracket \mathbf{e} \rrbracket$ for some \mathbf{e} .
- 2 L is accepted by a well-nested and finite automaton.

Further work

- Coequations for well-nestedness?
- Instantiation to more concrete theories.
- Fully algebraic axiomatization.

kap.pe/slides

arxiv.org/abs/1907.05920