<table>
<thead>
<tr>
<th>Weak $\mu$-equality</th>
<th>Avoiding $\alpha$-conversion</th>
<th>First-order proof</th>
<th>Higher-order proof</th>
<th>Proof using tree automata</th>
<th>Summary</th>
</tr>
</thead>
</table>

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
On Equal $\mu$-Terms

Jörg Endrullis*  Clemens Grabmayer$^\triangle$
Jan Willem Klop*  Vincent van Oostrom$^\triangle$

$^\triangle$) Universiteit Utrecht
*) Vrije Universiteit Amsterdam

TeReSe (autumn),
VU Amsterdam
November 18, 2010
Overview

1. Weak \( \mu \)-equality

2. Avoiding \( \alpha \)-conversion in \( \mu \)-reductions

3. Decidability of \( =_{\mu/\alpha} \) by a first-order proof

4. Decidability of \( =_{\mu/\alpha} \) by a higher-order proof

5. Decidability of \( =_{\mu/\alpha} \) using regular languages

6. Summary
Finite representation of infinite pattern

\[ \cap \cup \cap \cup \cap \cup \cap \cup \cup \cdots \]

finite representation?
Finite representation of infinite pattern

\[ \cap \cap \cap \cap \cap \cdots \]

finite representation?

\[ \mu x . \cap x \]
Finite representation of infinite pattern

$\mu x. \cap x \cdots$

finite representation?

$\mu x. s \rightarrow s[x := \mu x. s]$
Finite representation of infinite pattern

\[ \mu x. \cap x \]

finite representation?

\[ \mu x.s \rightarrow s[ x := \mu x.s ] \]

\[ \mu x. \cap x \rightarrow \cap \mu x. \cap x \rightarrow \cap \cap \mu x. \cap x \rightarrow \cap \cap \cap \mu x. \cap x \rightarrow \ldots \]
Finite representation of infinite pattern

\[\cap \cap \cap \cap \cap \cap \ldots\]

finite representation?

\[\mu x. \cap x\]

with \(\mu\)-rule

\[\mu x.s \rightarrow s[x := \mu x.s]\]

\[\mu x. \cup x \rightarrow \cup \mu x. \cup x \rightarrow \cup \cup \mu x. \cup x \rightarrow \cup \cup \cup \mu x. \cup x \rightarrow \ldots\]

hieroglyph \(\sim\) \(\Rightarrow\) phoenician \(\breve{\sim}\) \(\Rightarrow\) greek \(\mu\)
Finite representations of infinite pattern

\[ \bigcap \ldots \bigcap \ldots \bigcap \ldots \bigcap \ldots \bigcap \ldots \]

represented by

\[ \mu x. \bigcap x \]

other representations of same pattern?

Endrullis, Grabmayer, Klop, van Oostrom

On Equal \( \mu \)-Terms
## Finite representations of infinite pattern

\[ \cap \cdots \cap \cup \cdots \cup \cap \cdots \cup \cdots \]

represented by

\[ \mu x. \cap x \]

other representations of same pattern?

\[ \mu x'. \cap x' \]

---

**Endrullis, Grabmayer, Klop, van Oostrom**

On Equal $\mu$-Terms
Finite representations of infinite pattern

\[ \cap \cdots \]

represented by

\[ \mu x . \cap x \]

other representations of same pattern?

\[ \mu x' . \cap x' \]

\[ \cap \mu y . \cup y \]
Finite representations of infinite pattern

\[ \cdots \]

represented by

\[ \mu x. \cap x \]

other representations of same pattern?

\[ \mu x'. \cap x' \]

\[ \cap \mu y. \cap y \]

\[ \cap \mu z. \cap z \]
Finite representations of infinite pattern

\[ \cap \cap \cap \cap \cap \cap \cap \cap \cap \cap \cap \cap \cdots \]

represented by

\[ \mu x. \cap x \]

other representations of same pattern?

\[ \mu x'. \cap x' \]

\[ \cap \mu y. \cup y \]

\[ \cap \mu z. \cup z \]

\[ \mu w. \cup \cup w \]
Finite representations of infinite pattern

\[ \cap \ldots \]
represented by
\[ \mu x. \cap x \]
other representations of same pattern?
\[ \mu x'. \cap x' \]
\[ \cap \mu y. \cap y \]
\[ \cap \mu z. \cap z \]
\[ \mu w. \cap \cap w \]
when are two representations the same (finitely)?
Weak $\mu$-equality

- Weak $\mu$-equality on $\mu$-terms:

$$=_{\mu} := (\leftarrow_{\mu} \cup \rightarrow_{\mu})^*$$

(convertibility with respect to $\rightarrow_{\mu}$).

- Weak $\mu$-equality on $\mu$-pseudoterms:

$$=_{\mu/\alpha} := (\leftarrow_{\mu/\alpha} \cup \rightarrow_{\mu/\alpha})^* \cup =_{\alpha}$$

(convertibility with respect to $\rightarrow_{\mu/\alpha} := =_{\alpha} \cdot \rightarrow_{\mu} \cdot =_{\alpha}$).
Weak $\mu$-equality

- **Weak $\mu$-equality on $\mu$-terms:**

  $=\mu := (\leftarrow_\mu \cup \rightarrow_\mu)^*$

  (convertibility with respect to $\rightarrow_\mu$).

- **Weak $\mu$-equality on $\mu$-pseudoterms:**

  $=_{\mu/\alpha} := (\leftarrow_{\mu/\alpha} \cup \rightarrow_{\mu/\alpha})^* \cup =_\alpha$

  (convertibility with respect to $\rightarrow_{\mu/\alpha} := =_\alpha \cdot \rightarrow_\mu =_\alpha$).

**Proposition**

*For all $M, N \in \text{Ter}(\mu)$ and $s, t \in \text{PTer}(\mu)$:*

$s =_{\mu/\alpha} t \iff [s] =_\mu [t]$
$\mu$-pseudoterms, $\mu$-terms

Inductive definition of the set $P\text{Ter}(\mu)$ of $\mu$-pseudoterms:

(i) $x, y, z, \ldots \in P\text{Ter}(\mu)$ (variables);
(ii) $c, d, e, \ldots \in P\text{Ter}(\mu)$ (constants);
(iii) $s, t \in P\text{Ter}(\mu) \implies F(s, t) \in P\text{Ter}(\mu)$;
(iv) $s \in P\text{Ter}(\mu)$ and $x$ a variable $\implies \mu x.s \in P\text{Ter}(\mu)$.

Notation:

- $s \rightarrow_{\alpha} t$ for $\alpha$-renaming, and $s =_{\alpha} t$ for $\alpha$-equivalence induced by $\alpha$-conversion $=_{\alpha} := (\leftarrow_{\alpha} \cup \rightarrow_{\alpha})^*$. 

- $s[x := t]$ for $\alpha$-converting substitution à la Curry.

The set $T\text{er}(\mu)$ of $\mu$-terms consists of $\alpha$-equivalence classes of $\mu$-pseudoterms.
<table>
<thead>
<tr>
<th>Weak $\mu$-equality</th>
<th>Avoiding $\alpha$-conversion</th>
<th>First-order proof</th>
<th>Higher-order proof</th>
<th>Proof using tree automata</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deciding weak $\mu$-equality by rewriting?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Deciding weak $\mu$-equality by rewriting?

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Deciding weak $\mu$-equality by rewriting?

- $\mu$-reduction $\mu x.s \rightarrow s[x := \mu x.s]$ confluent but not terminating

  $\mu x.F(c, x) \rightarrow F(c, \mu x.F(c, x)) \rightarrow F(c, F(c, \mu x.F(c, x))) \rightarrow \ldots$
Deciding weak $\mu$-equality by rewriting?

- **$\mu$-reduction** $\mu x. s \rightarrow s[x := \mu x. s]$ confluent but not terminating
  
  $\mu x. F(c, x) \rightarrow F(c, \mu x. F(c, x)) \rightarrow F(c, F(c, \mu x. F(c, x))) \rightarrow \ldots$

- **$\mu$-expansion** $s[x := \mu x. s] \rightarrow \mu x. s$ terminating but not confluent
  
  $\not\Rightarrow F(M, M) \rightarrow N \leftarrow \mu x. F(M, F(c, x))$ for

  $M = \mu y. F(c, \mu x. F(y, F(c, x)))$

  $N = F(M, F(c, \mu x. F(M, F(c, x))))$

How to overcome?
Deciding weak $\mu$-equality by rewriting!

$\mu$-reduction non-terminating but active part repeats

$$\mu x. F(c, x) \rightarrow F(c, \mu x. F(c, x)) \rightarrow F(c, F(c, \mu x. F(c, x))) \rightarrow \ldots$$
Deciding weak $\mu$-equality by rewriting!

$\mu$-reduction non-terminating but active part repeats

$$\mu x. F(c, x) \rightarrow F(c, \mu x. F(c, x)) \rightarrow F(c, F(c, \mu x. F(c, x))) \rightarrow \ldots$$

Active part and repetition intuitions formalised in rest of talk

- Clemens: proof system
- Jörg: automata

Allows to bound the search space (loop checking).
Deciding weak $\mu$-equality by rewriting!

$\mu$-reduction non-terminating but **active part repeats**

\[ \mu x.F(c, x) \rightarrow F(c, \mu x.F(c, x)) \rightarrow F(c, F(c, \mu x.F(c, x))) \rightarrow \ldots \]

Active part and repetition intuitions formalised in rest of talk
- Clemens: proof system
- Jörg: automata

Allows to bound the search space (loop checking).
Problem dealt with now: dealing with $\alpha$-equivalence

\[ \mu x.F(c, x) \rightarrow F(c, \mu y.F(c, y)) \rightarrow F(c, F(c, \mu z.F(c, z))) \rightarrow \ldots \]

Repetition?
Overview

1. Weak $\mu$-equality

2. Avoiding $\alpha$-conversion in $\mu$-reductions

3. Decidability of $\equiv_{\mu/\alpha}$ by a first-order proof

4. Decidability of $\equiv_{\mu/\alpha}$ by a higher-order proof

5. Decidability of $\equiv_{\mu/\alpha}$ using regular languages

6. Summary
\( \alpha \)-conversion unavoidable in \( \lambda \)-calculus

\[
(\lambda w.ww)\lambda x y. x y
\]
α-conversion unavoidable in λ-calculus

\[(\lambda w. ww) \lambda x y. x y\]

\[\rightarrow (\lambda x y. x y) \lambda x y. x y\]
\(\alpha\)-conversion unavoidable in \(\lambda\)-calculus

\[(\lambda w.ww)\lambda xy.xy\]

\[\rightarrow (\lambda xy.xy)\lambda xy.xy\]

\[\rightarrow \lambda y.(\lambda xy.xy)y\]
α-conversion unavoidable in λ-calculus

\((\lambda w.ww)\lambda xy.xy\)

\(\rightarrow (\lambda xy.xy)\lambda xy.xy\)

\(\rightarrow \lambda y.(\lambda xy.xy)y\)

\(\rightarrow \lambda y.(\lambda y.yy) \quad \text{wrong!}\)
\(\alpha\)-conversion unavoidable in \(\lambda\)-calculus

\[
(\lambda w.ww)\lambda xy.xy \\
\rightarrow (\lambda xy.xy)\lambda xy.xy \\
\rightarrow \lambda y.(\lambda xy.xy)y \\
\rightarrow \lambda y.(\lambda y.yy) \text{ wrong!}
\]

- first step: non-linear (duplicating)
- second step: non-development (redex was created by first)
- third step: non-weak (redex below \(\lambda\))

\(\alpha\)-conversion can be avoided if one of these does hold.
Safe reduction

term is safe if \( \alpha \)-free substitution \( s[x := t] \) correct during reduction

Definition (\( \alpha \)-free substitution)

- \( x[x := t] = t \)
- \( y[x := t] = y \)
- \( (F(s, s'))[x := t] = F(s[x := t], s'[x := t]) \)
- \( (\mu x.s)[x := t] = \mu x.s \)
- \( (\mu y.s)[x := t] = \mu y.s[x := t] \)
Is the following term safe?

\[ \mu x.F(y, \mu y.x) \]
Unsafe $\mu$-terms

Is the following term safe?

$$\mu x. F(y, \mu y.x)$$

No:

$$\rightarrow F(y, \mu y.\mu x.F(y, \mu y.x))$$
Unsafe $\mu$-terms

Is the following term safe?

$$\mu x. F(y, \mu y.x)$$

No:

$$\rightarrow F(y, \mu y.\mu x. F(y, \mu y.x))$$

but can be $\alpha$-converted to safe $\mu$-term

$$\mu x. F(y, \mu z.x)$$

$$\rightarrow F(y, \mu z.\mu x. F(y, \mu z.x))$$

$$\rightarrow \ldots$$
Unsafe $\mu$-terms

Is the following term safe?

$\mu x. F(y, \mu y. x)$

No:

$\rightarrow F(y, \mu y. \mu x. F(y, \mu y. x))$

but can be $\alpha$-converted to safe $\mu$-term

$\mu x. F(y, \mu z. x)$

$\rightarrow F(y, \mu z. \mu x. F(y, \mu z. x))$

$\rightarrow \ldots$

can this always be done?

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Analysis of problem: self-capturing chains

A self-capturing chain of length 5 for the term $\mu x. F(y, \mu z.F(x, \mu y.z))$. 
Self-capture-freeness guarantees safety

Definition
Term is self-capture-free if no self-capturing chains
Self-capture-freeness guarantees safety

Definition
Term is self-capture-free if no self-capturing chains

Theorem (Preservation of Self-capture-freeness)
If $s \rightarrow t$ and $s$ self-capture-free then $t$ self-capture-free.
Definition

Term is self-capture-free if no self-capturing chains

Theorem (Preservation of Self-capture-freeness)

If $s \rightarrow t$ and $s$ self-capture-free then $t$ self-capture-free.

Theorem (Self-capture-free $\alpha$-conversion)

Every term can be $\alpha$-converted to a self-capture-free term.

Proof.

Choose all bound-variables distinct and distinct from free ones.
Overview

1. Weak $\mu$-equality

2. Avoiding $\alpha$-conversion in $\mu$-reductions

3. Decidability of $\equiv_{\mu/\alpha}$ by a first-order proof

4. Decidability of $\equiv_{\mu/\alpha}$ by a higher-order proof

5. Decidability of $\equiv_{\mu/\alpha}$ using regular languages

6. Summary
Decision problem for weak $\mu$-equality

We address:

**WEAK $\mu$-EQUALITY PROBLEM**

*Instance:* $\mu$-terms $M, N$

*Question:* Does $M =_\mu N$ hold?

and its ‘first-order’ version:

**WEAK $\mu$-EQUALITY PROBLEM on $\mu$-pseudoterms**

*Instance:* $\mu$-pseudoterms $s, t$

*Question:* Does $s =_{\mu/\alpha} t$ hold?
Structure of the first-order proof

\[ s \overset{\mu/\alpha}{=} t \]
Structure of the first-order proof
Structure of the first-order proof

\[ s =_{\mu/\alpha} t \]

\[ r \]
Structure of the first-order proof

Weak $\mu$-equality  Avoiding $\alpha$-conversion  First-order proof  Higher-order proof  Proof using tree automata  Summary

Endrullis, Grabmayer, Klop, van Oostrom  On Equal $\mu$-Terms
Structure of the first-order proof

capture-avoiding

\[ s' =_\alpha s \]

\[ s =_{\mu/\alpha} t \]

\[ t =_\alpha t' \]

capture-avoiding

\[ r \]
Structure of the first-order proof

capture-avoiding

\[ s' =_\alpha s \]
\[ \mu/\alpha \]
\[ r =_\alpha \]

capture-avoiding

\[ t =_\alpha t' \]
\[ s'' =_\alpha \]
\[ =_\alpha \]

\[ s'\]
\[ s \]
\[ r \]
\[ t \]
\[ t' \]

\[ s'' \]

Weak $\mu$-equality
Avoiding $\alpha$-conversion
First-order proof
Higher-order proof
Proof using tree automata
Summary

Endrullis, Grabmayer, Klop, van Oostrom
On Equal $\mu$-Terms
Structure of the first-order proof

capture-avoiding

<table>
<thead>
<tr>
<th>Weak $\mu$-equality</th>
<th>Avoiding $\alpha$-conversion</th>
<th>First-order proof</th>
<th>Higher-order proof</th>
<th>Proof using tree automata</th>
<th>Summary</th>
</tr>
</thead>
</table>

On Equal $\mu$-Terms

Endrullis, Grabmayer, Klop, van Oostrom
Thus:

- the weak \( \mu \)-equality problem for \( \mu \)-terms can be reduced to:

  \[
  \text{JOINABILITY PROBLEM up to } \equiv_\alpha \text{ for } \rightarrow_\mu \text{ on capture-avoiding } \mu\text{-pseudoterms}
  \]

  \[
  \text{Instance: capture avoiding } \mu\text{-pseudoterms } s, t
  \]

  \[
  \text{Question: Are there } s', t' \text{ with } s \rightarrow_{\text{std}} s' =_\alpha t' \leftarrow_{\text{std}} t ?
  \]
Structure of the first-order proof

Thus:

- the weak $\mu$-equality problem for $\mu$-terms can be reduced to:

```
JOINABILITY PROBLEM UP TO $\equiv_\alpha$ FOR $\rightarrow_\mu$ on capture-avoiding $\mu$-pseudoterms
```

- Instance: capture avoiding $\mu$-pseudoterms $s, t$
- Question: Are there $s', t'$ with $s \rightarrow_{\text{std}} s' \equiv_\alpha t' \iff_{\text{std}} t$?

Further proof strategy. Obtain a proof system $S$ such that:

1. $S$ is complete for $\rightarrow_\mu$-joinability up to $\equiv_\alpha$ on capture-avoiding $\mu$-pseudoterms.
2. the search-space for irredundant derivations in $S$ is always finite.
Complete proof system (I) for $=_{\mu/\alpha}$ on $\mu$-pseudoterms

\[
\begin{align*}
(\mu\text{-unfolding}) & \\
\mu x . s & = s[x := \mu x . s] \\
(\alpha\text{-renaming}) & \\
\mu x . s & = \mu y . s[x := y]
\end{align*}
\]

\[
\begin{align*}
(\text{REFL}) & \\
s & = t & \text{SYMM} \\
s & = t & \text{SYMM} \\
\mu x . s & = \mu x . t & \mu\text{-COMPAT} \\
s_1 & = t_1 & s_2 = t_2 & F(s_1, s_2) = F(t_1, t_2) & \text{F-COMPAT}
\end{align*}
\]

- extension of a complete proof system for $=_{\alpha}$ (i.e. $\rightarrow_{\alpha}$-conversion)
- derivations correspond to $\rightarrow_{\mu/\alpha}$-conversions
Complete proof system (I) for $=_{\mu/\alpha}$ on $\mu$-pseudoterms

(\mu\text{-unfolding})
\[
\mu x.s = s[x := \mu x.s]
\]

(\alpha\text{-renaming})
\[
\mu x.s = \mu y.s[x := y]
\]

\begin{align*}
\text{(REFL)} & \quad s = t \quad \text{SYMM} \\
\frac{s = t}{s = s} & \quad \frac{t = s}{t = s}
\end{align*}

\begin{align*}
\mu x.s & = \mu x.t \quad \mu\text{-COMPAT} \\
\frac{s_1 = t_1}{s_1 = t_1} & \quad \frac{s_2 = t_2}{s_2 = t_2}
\end{align*}

\begin{align*}
\frac{s = r}{s = t} & \quad \text{TRANS} \\
\frac{r = t}{r = t}
\end{align*}

\begin{align*}
\text{F}(s_1, s_2) & = \text{F}(t_1, t_2) \quad \text{F-COMPAT}
\end{align*}

- extension of a complete proof system for $=_{\alpha}$ (i.e. $\rightarrow_{\alpha}$-conversion)
- derivations correspond to $\rightarrow_{\mu/\alpha}$-conversions
- **Disadvantages:**
  - complex search space for proofs (no subformula property)
  - does not directly give rise to a decision method
Example

\[ \mu x_3 x_2 x_1 . x_2 =_{\mu/\alpha} \mu y z . y \]

holds because of:

\[ \mu x_3 x_2 x_1 . x_2 \rightarrow_{\mu} \mu x_2 x_1 . x_2 \rightarrow_{\mu} \mu x_2 . x_2 =_\alpha \mu y . y \leftarrow_{\mu} \mu y z . y \]

which gives rise to the derivation:

\[
\begin{array}{c}
\text{(\(\mu\)-unfolding)} \\
\mu x_1 . x_2 = x_2 \\
\mu x_2 x_1 . x_2 = \mu x_2 . x_2 \\
\mu x_3 x_2 x_1 . x_2 = \mu x_2 . x_2 \\
\mu x_3 x_2 x_1 . x_2 = \mu y . y \\
\mu x_3 x_2 x_1 . x_2 = \mu y z . y
\end{array}
\]
Complete proof system (II) for $=_{\mu/\alpha}$ on $\mu$-pseudoterms

- $s = s$ (if $s$ a variable or a constant)
- $s[x := z] = t[y := z]$ \quad $\mu x.s = \mu y.t$ ($z$ fresh)
- $s_1 = t_1$ \quad $s_2 = t_2$ \quad $F(s_1, s_2) = F(t_1, t_2)$ \quad F-COMPAT
- $s[s[x := \mu x.s]] = t$ \quad $\mu x.s = t$ \quad FOLD$_l$
- $s = t[y := \mu y.t]$ \quad $s = \mu y.t$ \quad FOLD$_r$

- extension of Schroer’s characterisation of $\rightarrow_{\alpha}$-conversion
- derivations can be obtained by transitivity/symmetry-elimination in derivations of the previous system.
- derivations correspond to $\rightarrow_{\mu/\alpha}$-standard reductions
Complete proof system (II) for $\equiv_{\mu/\alpha}$ on $\mu$-pseudoterm

- Extension of Schroer's characterisation of $\rightarrow_{\alpha}$-conversion
- Derivations can be obtained by transitivity/symmetry-elimination in derivations of the previous system.
- Derivations correspond to $\rightarrow_{\mu/\alpha}$-standard reductions
- *Advantage:* (much more) restricted search space for derivations
- Certain *disadvantage:* capture of free variables in $\mu$-applications
<table>
<thead>
<tr>
<th>Weak $\mu$-equality</th>
<th>Avoiding $\alpha$-conversion</th>
<th>First-order proof</th>
<th>Higher-order proof</th>
<th>Proof using tree automata</th>
<th>Summary</th>
</tr>
</thead>
</table>

## Example

**Proof System (II)**

\[
\begin{align*}
\mu x_1 . u &= \mu z . u \\
\mu x_2 x_1 . x_2 &= \mu y z . y \\
\mu x_3 x_2 x_1 . x_2 &= \mu y z . y
\end{align*}
\]

- $u = u$ (FOLD$_r$)
- $u = \mu z . u$ (FOLD$_l$)
- $\mu x_1 . u = \mu z . u$ (FOLD$_l$)
- $\mu x_2 x_1 . x_2 = \mu y z . y$ (FOLD$_l$)
- $\mu x_3 x_2 x_1 . x_2 = \mu y z . y$ (FOLD$_l$)
Example

Proof System (II)

\[
\begin{align*}
\mu x_1.u &= \mu z.u \\
\mu x_2 x_1.x_2 &= \mu y z.y \\
\mu x_3 x_2 x_1.x_2 &= \mu y z.y
\end{align*}
\]

Proof System (III)

\[
\begin{align*}
x_2 = y &\vdash x_2 = y \\
x_2 = y &\vdash x_2 = \mu z.y \\
x_2 = y &\vdash x_1.x_2 = \mu z.y \\
\vdash \mu x_2 x_1.x_2 &= \mu y z.y \\
\vdash \mu x_3 x_2 x_1.x_2 &= \mu y z.y
\end{align*}
\]
Complete proof system (III) for $\equiv_{\mu/\alpha}$ on $\mu$-pseudoterms

---

\[ x = y \vdash x = y \quad \text{(restr-REFL)} \]
\[ \vdash s = s \quad \text{(if } s \text{ a variable or a constant)} \]
\[ \Gamma, \bar{z} = \bar{u} \vdash s = t \quad \text{COMPR (if } x \not\in \text{FV}(\mu \bar{z}.s) \text{ and } y \not\in \text{FV}(\mu \bar{u}.t)) \]
\[ \Gamma, x = y, \bar{z} = \bar{u} \vdash s = t \]
\[ \Gamma \vdash \mu x.s = \mu y.t \quad \mu \]
\[ \Gamma \vdash s_1 = t_1 \quad \Gamma \vdash s_2 = t_2 \quad \text{F} \]
\[ \Gamma \vdash F(s_1, s_2) = F(t_1, t_2) \]
\[ \Gamma \vdash s[x := \mu x.s] = t \quad \text{FOLD}_l \]
\[ \Gamma \vdash s = t[y := \mu y.t] \quad \text{FOLD}_r \]
\[ \Gamma \vdash \mu x.s = t \]
\[ \Gamma \vdash s = \mu y.t \]

- extension of Kahrs’ characterisation of $\alpha$-conversion
- der’s obtainable by trans./symm.-elim. from der’s in system (I)
- derivations correspond to $\rightarrow_{\mu/\alpha}$-standard reductions
Complete proof system (III) for $\equiv_{\mu/\alpha}$ on $\mu$-pseudoterms

\[
\frac{x = y}{x = y} \vdash x = y
\]

\[
\frac{s = s}{s = s}
\]

\[
\frac{\Gamma, \tilde{z} = \tilde{u} \vdash s = t}{\Gamma, x = y, \tilde{z} = \tilde{u} \vdash s = t}
\]

\[
\frac{\Gamma, \tilde{z} = \tilde{u} \vdash x = y}{\Gamma, \tilde{z} = \tilde{u} \vdash s = t}
\]

\[
\frac{\Gamma, x = y \vdash s = t}{\Gamma \vdash \mu x.a = \mu y.b}
\]

\[
\frac{\Gamma \vdash s_1 = t_1 \quad \Gamma \vdash s_2 = t_2}{\Gamma \vdash F(s_1, s_2) = F(t_1, t_2)}
\]

\[
\frac{\Gamma \vdash s[x := \mu x.a]}{\Gamma \vdash s = \mu x.a}
\]

\[
\frac{\Gamma \vdash s[t y := \mu y.b]}{\Gamma \vdash s = \mu y.b}
\]

- extension of Kahrs’ characterisation of $\alpha$-conversion
- der’s obtainable by trans./symm.-elim. from der’s in system (I)
- derivations correspond to $\rightarrow_{\mu/\alpha}$-standard reductions
- advantage: restricted search space for derivations

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Complete proof system (III) for $\mu/\alpha$ on $\mu$-pseudoterms

\[
\frac{x = y \vdash x = y}{\text{restriction-REFL}}
\]

\[
\Gamma, x = y, \bar{z} = \bar{u} \vdash s = t \quad \text{COMPR} \quad (\text{if } x \notin \text{FV}(\mu \bar{z}.s) \text{ and } y \notin \text{FV}(\mu \bar{u}.t))
\]

\[
\Gamma \vdash \mu x.s = \mu y.t \quad \mu
\]

\[
\Gamma, x = y \vdash s = t \quad \text{UNFOLD}_l
\]

\[
\Gamma \vdash F(s_1, s_2) = F(t_1, t_2) \quad F
\]

\[
\Gamma \vdash s_1 = t_1 \quad \Gamma \vdash s_2 = t_2
\]

\[
\Gamma \vdash s = \mu y.t \quad \text{UNFOLD}_r
\]

- extension of Kahrs' characterisation of $\alpha$-conversion
- der's obtainable by trans./symm.-elim. from der's in system (I)
- derivations correspond to $\rightarrow_{\mu/\alpha}$-standard reductions
- advantage: restricted search space for derivations
Complete proof system (III) for $\equiv_{\mu/\alpha}$ on $\mu$-pseudoterm

\[
\begin{align*}
(\mu x)x &= (\mu y)y \\
(\mu z_1 x z_2)s &= (\mu u_1 y u_2)t \\
(\mu z_1 z_2)s &= (\mu u_1)\tilde{u}_2 t \\
(\mu z)\mu x.s &= (\mu u)\mu y.t \\
(\mu z)x.s &= (\mu u)\tilde{u}_2 t \\
(\mu z)\mu x.s &= (\mu u)t \\
(\mu z)s[x := \mu x.s] &= (\mu u)t \\
(\mu z)F(s_1, s_2) &= (\mu u)F(t_1, t_2) \\
(\mu z)s_1 &= (\mu u)t_1 \\
(\mu z)s_2 &= (\mu u)t_2 \\
(\mu z)s &= (\mu u)\mu y.t \\
(\mu z)s &= (\mu u)t[y := \mu y.t]
\end{align*}
\]

- extension of Kahrs’ characterisation of $\alpha$-conversion
- der’s obtainable by trans./symm.-elim. from der’s in system (I)
- derivations correspond to $\rightarrow_{\mu/\alpha}$-standard reductions
- advantage: restricted search space for derivations
<table>
<thead>
<tr>
<th>Weak $\mu$-equality</th>
<th>Avoiding $\alpha$-conversion</th>
<th>First-order proof</th>
<th>Higher-order proof</th>
<th>Proof using tree automata</th>
<th>Summary</th>
</tr>
</thead>
</table>

## Example

\[
\begin{align*}
\Gamma & : \mu x_3 x_2 x_1 . x_2 = \mu y z . y \\
\Gamma & : \mu x_2 x_1 . x_2 = \mu y z . y \\
\Gamma & : x_2 = y \vdash \mu x_1 . x_2 = \mu z . y \\
\Gamma & : x_2 = y \vdash x_2 = \mu z . y \\
\Gamma & : x_2 = y \vdash x_2 = y
\end{align*}
\]

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Example

\[
(\mu x_3) x_2 x_1 \cdot x_2 = (\mu y) z \cdot y \\
\mu x_2 x_1 \cdot x_2 = (\mu y) z \cdot y \\
(\mu x_2) \mu x_1 \cdot x_2 = (\mu y) \mu z \cdot y \\
(\mu x_2) x_2 = (\mu y) z \cdot y \\
(\mu x_2) x_2 = (\mu y) y
\]

UNFOLD$_l$
UNFOLD$_l$
UNFOLD$_l$
UNFOLD$_r$
Example

\[
\frac{(\mu x_3 x_2 x_1 . x_2) = (\mu y z . y)}{\text{UNFOLD}_l}
\]

\[
\frac{(\mu x_2) x_2 = (\mu y) \mu z . y}{\mu}
\]

\[
\frac{(\mu x_2) x_2 = (\mu y) \mu z . y}{\text{UNFOLD}_l}
\]

\[
\frac{(\mu x_2) x_2 = (\mu y) \mu z . y}{\text{UNFOLD}_r}
\]

Extraction of reductions:

\[
\mu x_3 x_2 x_1 . x_2 \rightarrow_\mu \mu x_2 x_1 . x_2 \triangleright_{\text{frz}} (\mu x_2) \mu x_1 . x_2 \rightarrow_\mu (\mu x_2) x_2
\]

\[
=_{\alpha} (\mu y) y \leftarrow_\mu (\mu y) \mu z . y \triangleleft_{\text{frz}} \mu y z . y
\]
Example

\[
\begin{align*}
(\mu x_3 x_2 x_1 . x_2) &= (\mu y z . y) \\
(\mu x_2 x_1 . x_2) &= (\mu y z . y) \\
(\mu x_2) \mu x_1 . x_2 &= (\mu y) \mu z . y \\
(\mu x_2) x_2 &= (\mu y) \mu z . y \\
(\mu x_2) x_2 &= (\mu y) y
\end{align*}
\]

Extraction of reductions:

\[
\begin{align*}
\mu x_3 x_2 x_1 . x_2 &\rightarrow_\mu \mu x_2 x_1 . x_2 \\
\mu x_2 x_1 . x_2 &\triangleright_{\text{frz}} (\mu x_2) \mu x_1 . x_2 \\
&\rightarrow_\mu (\mu x_2) x_2 \\
&=_{\alpha} (\mu y) y \\
&\leftarrow_\mu (\mu y) \mu z . y \\
&\triangleleft_{\text{frz}} \mu y z . y
\end{align*}
\]
gives a joining pair of standard reductions:

\[
\begin{align*}
\mu x_3 x_2 x_1 . x_2 &\rightarrow_\mu \mu x_2 x_1 . x_2 \\
\mu x_2 x_1 . x_2 &\rightarrow_\mu \mu x_2 . x_2 \\
\mu x_2 . x_2 &\rightarrow_\alpha \mu y . y \\
&\leftarrow_\mu \mu y z . y
\end{align*}
\]
Subterm closure

The $\mu\pi$-calculus on $\mu$-pseudoterm:\n
$$F(s_1, s_2) \rightarrow s_i \quad \text{for } i \in \{1, 2\} \quad \text{(F-projection)}$$
$$\mu x.s \rightarrow s \quad \text{(}\mu\text{-projection)}$$
$$\mu x.s \rightarrow s[x := \mu x.s] \quad \text{(}\mu\text{-reduction)}$$

By $\rightarrow^{\varepsilon}_{\mu\pi}$ we denote $\mu\pi$-root-reduction.

The subterm closure $SC(s)$ of a capture-avoiding $s \in PTer(\mu)$ is:

$$SC(s) := \{ t \in PTer(\mu) \mid s \rightarrow^{\varepsilon}_{\mu\pi} t \}.$$ 

**Theorem**

For all capture-avoiding $s \in PTer(\mu)$, $SC(s)$ is finite.
The subterm closure of $\mu xyz.y$. 
The subterm closure of $\mu x. F(x, \mu y. F(x, y))$.
Decidability of $\equiv_{\mu/\alpha}$ by a first-order proof

**Lemma**

*Provability in system (III) of formulas $\vdash s = t$, where $s, t \in PTer(\mu)$ are capture-avoiding, is decidable.*

**Proof.**

- subformula property: for an equation $(\mu \ldots)s' = (\mu \ldots)t'$ in a derivation $\mathcal{D}$ with conclusion $(\ )s = (\ )t$ it holds that $s' \in SC(s)$ and $t' \in SC(t)$. 

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Decidability of $\equiv_{\mu/\alpha}$ by a first-order proof

**Lemma**

*Provability in system (III) of formulas* $\vdash s = t$, where $s, t \in PTer(\mu)$ *are capture-avoiding, is decidable.*

**Proof.**

- **subformula property:** for an equation $(\mu \ldots)s' = (\mu \ldots)t'$ in a derivation $\mathcal{D}$ with conclusion $(.)s = (.)t$ it holds that $s' \in SC(s)$ and $t' \in SC(t)$.

- **bound $L$ on annotation lengths:** if rule COMPR is applied ‘greedily’, $L :=$ no. of binder occurr’s in ps.terms in conclusion.
Decidability of $=_{\mu/\alpha}$ by a first-order proof

Lemma

Provability in system (III) of formulas $\vdash s = t$, where $s, t \in PTer(\mu)$ are capture-avoiding, is decidable.

Proof.

- **Subformula property**: for an equation $(\mu \ldots) s' = (\mu \ldots) t'$ in a derivation $\mathcal{D}$ with conclusion $( ) s = ( ) t$ it holds that $s' \in \text{SC}(s)$ and $t' \in \text{SC}(t)$.

- **Bound $L$ on annotation lengths**: if rule COMPR is applied ‘greedily’, $L := \text{no. of binder occurr's in ps.terms in conclusion}$.

- **Bound on the size of irredundant derivations**: as a consequence, the size of an irredundant derivation (no formula repetitions) with conclusion $s = t$ is bounded by $|\pi(L)| \cdot |\text{SC}(s)| \cdot |\text{SC}(t)|$. 
Decidability of $\mu/\alpha$ by a first-order proof

**Lemma**

Provability in system (III) of formulas $\vdash s = t$, where $s, t \in PTer(\mu)$ are capture-avoiding, is decidable.

**Proof.**

- **subformula property**: for an equation $(\mu \ldots)s' = (\mu \ldots)t'$ in a derivation $D$ with conclusion $(s) = (t)$ it holds that $s' \in SC(s)$ and $t' \in SC(t)$.

- **bound $L$ on annotation lengths**: if rule COMPR is applied ‘greedily’, $L := \text{no. of binder occurr's in ps.terms in conclusion}$.

- **bound on the size of irredundant derivations**: as a consequence, the size of an irredundant derivation (no formula repetitions) with conclusion $s = t$ is bounded by $|\pi(L)| \cdot |SC(s)| \cdot |SC(t)|$.

**Theorem**

Weak $\mu$-equality is decidable.
Overview

1. Weak $\mu$-equality

2. Avoiding $\alpha$-conversion in $\mu$-reductions

3. Decidability of $=_{\mu/\alpha}$ by a first-order proof

4. Decidability of $=_{\mu/\alpha}$ by a higher-order proof

5. Decidability of $=_{\mu/\alpha}$ using regular languages

6. Summary
Structure of the higher-order proof

\[ M \overset{=} {\mu} N \]
Structure of the higher-order proof

\[ M \overset{=\mu}{\Rightarrow} N \]
Structure of the higher-order proof

\[
M \overset{\mu}{=} N
\]

\[
P
\]

Endrullis, Grabmayer, Klop, van Oostrom

On Equal \(\mu\)-Terms
Structure of the higher-order proof

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Complete proof system (II) for \(=_{\mu/\alpha}\) on \(\mu\)-pseudoterms

\[
\begin{align*}
s[x := z] = t[y := z] & \quad \mu \text{ (z fresh)} \\
\mu x.s = \mu y.t & \\
\mu x.s = t & \quad \text{FOLD}_l \\
F(s_1, s_2) = F(t_1, t_2) & \quad \text{F-COMPAT} \\
s[y := \mu y.t] & \quad \text{FOLD}_r \\
s = \mu y.t & \quad \text{(if s a variable or a constant)}
\end{align*}
\]

- extension of Schroer’s characterisation of \(\rightarrow_{\alpha}\)-conversion
- derivations correspond to \(\rightarrow_{\mu/\alpha}\)-standard reductions
- *advantage*: restricted search space for derivations
- *disadvantage*: capture of free variables in \(\mu\)-applications
Complete proof system for $=_{\mu}$ on $\mu$-terms

\begin{align*}
\text{weak } \mu\text{-equality} & \quad \text{Avoiding } \alpha\text{-conversion} & \quad \text{First-order proof} & \quad \text{Higher-order proof} & \quad \text{Proof using tree automata} & \quad \text{Summary} \\
\end{align*}

\[ s = s \]  
(if $s$ a variable or a constant)

\[ s[x := n] = t[y := n] \quad \mu \quad \mu y.t \quad \mu x.s = \mu y.t \quad \mu (n\text{ fresh numeral}) \]

\[
\begin{align*}
\frac{s[x := \mu x.s] = t}{\mu x.s = t} & \quad \text{FOLD}_l \\
\frac{s = t[y := \mu y.t]}{s = \mu y.t} & \quad \text{FOLD}_r \\
\frac{s_1 = t_1}{s_2 = t_2} & \quad \text{F-COMPAT} \\
\frac{F(s_1, s_2) = F(t_1, t_2)}{F(s_1, s_2) = F(t_1, t_2)}
\end{align*}
\]

- extension of Schroer’s characterisation of $\rightarrow_{\alpha}$-conversion
- derivations correspond to $\rightarrow_{\mu/\alpha}$-standard reductions
- advantage: restricted search space for derivations
- a certain subformula property
<table>
<thead>
<tr>
<th>Weak $\mu$-equality</th>
<th>Avoiding $\alpha$-conversion</th>
<th>First-order proof</th>
<th>Higher-order proof</th>
<th>Proof using tree automata</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Weak $\mu$-equality

2. Avoiding $\alpha$-conversion in $\mu$-reductions

3. Decidability of $\equiv_{\mu/\alpha}$ by a first-order proof

4. Decidability of $\equiv_{\mu/\alpha}$ by a higher-order proof

5. Decidability of $\equiv_{\mu/\alpha}$ using regular languages

6. Summary
<table>
<thead>
<tr>
<th>Weak $\mu$-equality</th>
<th>Avoiding $\alpha$-conversion</th>
<th>First-order proof</th>
<th>Higher-order proof</th>
<th>Proof using tree automata</th>
<th>Summary</th>
</tr>
</thead>
</table>

**Decidability of $=_{\mu/\alpha}$ using regular languages**

An alternative approach: using regular languages.
Decidability of $\mu/\alpha$ using regular languages

An alternative approach: using regular languages.

- For a capture-avoiding $M$ we construct a regular grammar $G_M$ generating the set of reducts of $M$ (without $\alpha$-conversion).
Decidability of $\equiv_{\mu/\alpha}$ using regular languages

An alternative approach: using regular languages.

- For a capture-avoiding $M$ we construct a regular grammar $G_M$ generating the set of reducts of $M$ (without $\alpha$-conversion).

- Given a regular grammar $G$, we construct a grammar $G^\alpha$ generating the closure of $G$ under $\alpha$-conversion.
Decidability of $\equiv_{\mu/\alpha}$ using regular languages

An alternative approach: using regular languages.

- For a capture-avoiding $M$ we construct a regular grammar $G_M$ generating the set of reducts of $M$ (without $\alpha$-conversion).

- Given a regular grammar $G$, we construct a grammar $G^\alpha$ generating the closure of $G$ under $\alpha$-conversion.

- Then $M =_{\alpha} N$ boils down to the question:

$$\mathcal{L}(G_M^\alpha) \cap \mathcal{L}(G_N^\alpha) \neq \emptyset?$$
Decidability of $\mu/\alpha$ using regular languages

An alternative approach: using regular languages.

- For a capture-avoiding $M$ we construct a regular grammar $G_M$ generating the set of reducts of $M$ (without $\alpha$-conversion).

- Given a regular grammar $G$, we construct a grammar $G^\alpha$ generating the closure of $G$ under $\alpha$-conversion.

- Then $M =_\alpha N$ boils down to the question:

  $$\mathcal{L}(G_M^\alpha) \cap \mathcal{L}(G_N^\alpha) \neq \emptyset?$$

This problem is known to be decidable.
Step 1: a regular grammar for $\mu$-reducts

Let $M \in Ter(\mu)$ be a capture-avoiding $\mu$-pseudoterm.
Step 1: a regular grammar for \( \mu \)-reducts

Let \( M \in \text{Ter}(\mu) \) be a capture-avoiding \( \mu \)-pseudoterm.

We construct a regular grammar \( G_M \) for the \( \mu \)-reducts of \( M \):

The start symbol of \( G_M \) is \( V_M \), and the rules are:

\[
\begin{align*}
V_{\mu \cdot x} N &\Rightarrow V_N[x:=\mu \cdot x. N] \\
V_{\mu \cdot x} N &\Rightarrow \mu \cdot x. V_N \\
V_{F(N,N')} &\Rightarrow F(V_N, V_{N'}) \\
V_x &\Rightarrow x
\end{align*}
\]

for every \( V_s \) such that \( s \in \text{SC}(M) \).
Step 1: a regular grammar for $\mu$-reducts

Let $M \in \text{Ter}(\mu)$ be a capture-avoiding $\mu$-pseudoterm.

We construct a regular grammar $\mathcal{G}_M$ for the $\mu$-reducts of $M$:

The start symbol of $\mathcal{G}_M$ is $V_M$, and the rules are:

1. $V_{\mu x.N} \Rightarrow V_N[x:=\mu x.N]$  
2. $V_{\mu x.N} \Rightarrow \mu x.V_N$  
3. $V_{F(N,N')} \Rightarrow F(V_N, V_{N'})$  
4. $V_x \Rightarrow x$

for every $V_s$ such that $s \in \text{SC}(M)$.

Lemma

$\mathcal{L}(\mathcal{G}_M) = \{ N \mid M \rightarrow^* N \}$

where $\rightarrow$ is $\alpha$-conversion free $\mu$-reduction.
Step 1: a regular grammar for $\mu$-reducts

Example

Let $M \equiv \mu y. F(x, y)$, then $G_M$ consists of:

- $V_{\mu y. F(x, y)} \Rightarrow (1) \ V_{F(x, \mu y. F(x, y))}$
- $V_{\mu y. F(x, y)} \Rightarrow (2) \ \mu y \cdot V_{F(x, y)}$
- $V_{F(x, y)} \Rightarrow (3) \ F(V_x, V_y)$
- $V_{F(x, \mu y. F(x, y))} \Rightarrow (3) \ F(V_x, V_{\mu y. F(x, y)})$
- $V_x \Rightarrow (4) \ x$
- $V_y \Rightarrow (4) \ y$

The start symbol of $G_M$ is $V_{\mu y. F(x, y)}$. 
Step 2: $\alpha$-conversion

Let $\mathcal{G}$ be **normalised** with start variable $V$ over a finite set of binder $B$.

We define a grammar accepting all $\alpha$-equivalent terms over $B$:
Step 2: $\alpha$-conversion

Let $G$ be normalised with start variable $V$ over a finite set of binder $B$. We define a grammar accepting all $\alpha$-equivalent terms over $B$:

Let $G^\alpha$ have start variable $V_{id,\varnothing}$, and for all:

- $\sigma : B \rightarrow B$ (renaming map),
- $\dagger \subseteq B$ (forbidden variables),

consist of rules:

- $V_{\sigma,\dagger} \Rightarrow \sigma(x) \in G^\alpha$ (renaming) if $V \Rightarrow x \in G$ and $x \notin \dagger$
- $V_{\sigma,\dagger} \Rightarrow \bot \in G^\alpha$ (name clash) if $V \Rightarrow x \in G$ and $x \in \dagger$
- $V_{\sigma,\dagger} \Rightarrow F(V'_{\sigma,\dagger}, V''_{\sigma,\dagger}) \in G^\alpha$ (propagation) if $V \Rightarrow F(V', V'') \in G$
- $V_{\sigma,\dagger} \Rightarrow \mu y(V'_{\sigma',\dagger'}) \in G$ (pick renaming) if $V \Rightarrow \mu x(V') \in G$

where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$. 
Step 2: $\alpha$-conversion

pick renaming: $V_{\sigma, \dagger} \Rightarrow \mu y(V'_{\sigma', \dagger'}) \in G$ if $V \Rightarrow \mu x(V') \in G$

where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example

Let $G$ have start variable $V_1$ and consist of the rules:

$V_1 \Rightarrow \mu x. V_2 \quad V_2 \Rightarrow \mu y. V_3 \quad V_3 \Rightarrow F(V_4, V_5) \quad V_4 \Rightarrow x \quad V_5 \Rightarrow y$
Step 2: $\alpha$-conversion

pick renaming: $V_{\sigma, \dagger} \Rightarrow \mu y(V'_{\sigma', \dagger'}) \in G$ if $V \Rightarrow \mu x(V') \in G$

where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example

Let $G$ have start variable $V_1$ and consist of the rules:

$V_1 \Rightarrow \mu x.V_2 \quad V_2 \Rightarrow \mu y.V_3 \quad V_3 \Rightarrow F(V_4, V_5) \quad V_4 \Rightarrow x \quad V_5 \Rightarrow y$

Note that $G$ generates the term $\mu x.\mu y.F(x, y)$. 
Step 2: $\alpha$-conversion

pick renaming: $V_{\sigma, \dagger} \Rightarrow \mu y (V'_{\sigma', \dagger'}) \in G$ if $V \Rightarrow \mu x (V') \in G$

where $y \in B$, $\sigma' = \sigma [x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example

Let $G$ have start variable $V_1$ and consist of the rules:

$V_1 \Rightarrow \mu x . V_2$  $V_2 \Rightarrow \mu y . V_3$  $V_3 \Rightarrow F(V_4, V_5)$  $V_4 \Rightarrow x$  $V_5 \Rightarrow y$

Note that $G$ generates the term $\mu x . \mu y . F(x, y)$.

Let $G^\alpha$ has start variable $V_1, \{x \mapsto x, y \mapsto y\}, \emptyset$, and contains rules:
Step 2: $\alpha$-conversion

pick renaming: $V_{\sigma, \dagger} \Rightarrow \mu y(V'_{\sigma', \dagger'}) \in G$ if $V \Rightarrow \mu x(V') \in G$

where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example

Let $G$ have start variable $V_1$ and consist of the rules:

$V_1 \Rightarrow \mu x.V_2 \quad V_2 \Rightarrow \mu y.V_3 \quad V_3 \Rightarrow F(V_4, V_5) \quad V_4 \Rightarrow x \quad V_5 \Rightarrow y$

Note that $G$ generates the term $\mu x.\mu y.F(x, y)$.

Let $G^\alpha$ has start variable $V_1,\{x \mapsto x, y \mapsto y\}, \emptyset$, and contains rules:

$V_1,\{x \mapsto x, y \mapsto y\}, \emptyset \Rightarrow \mu y.V_2,\{x \mapsto y, y \mapsto y\}, \{y\}$
Step 2: $\alpha$-conversion

**pick renaming:** $V_{\sigma, \dagger} \Rightarrow \mu y(V'_{\sigma', \dagger'}) \in G$ if $V \Rightarrow \mu x(V') \in G$

where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

**Example**

Let $G$ have start variable $V_1$ and consist of the rules:

- $V_1 \Rightarrow \mu x.V_2$
- $V_2 \Rightarrow \mu y.V_3$
- $V_3 \Rightarrow F(V_4, V_5)$
- $V_4 \Rightarrow x$
- $V_5 \Rightarrow y$

Note that $G$ generates the term $\mu x.\mu y.F(x, y)$.

Let $G^\alpha$ has start variable $V_1, \{x \mapsto x, y \mapsto y\}, \emptyset$, and contains rules:

- $V_1, \{x \mapsto x, y \mapsto y\}, \emptyset \Rightarrow \mu y.V_2, \{x \mapsto y, y \mapsto y\}, \{y\}$
- $V_2, \{x \mapsto y, y \mapsto y\}, \{y\} \Rightarrow \mu y.V_3, \{x \mapsto y, y \mapsto y\}, \{x\}$
Step 2: $\alpha$-conversion

pick renaming: $V_{\sigma,\dagger} \Rightarrow \mu y(V'_{\sigma',\dagger'}) \in G$ if $V \Rightarrow \mu x(V') \in G$

where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example

Let $G$ have start variable $V_1$ and consist of the rules:

$V_1 \Rightarrow \mu x.V_2 \quad V_2 \Rightarrow \mu y.V_3 \quad V_3 \Rightarrow F(V_4, V_5) \quad V_4 \Rightarrow x \quad V_5 \Rightarrow y$

Note that $G$ generates the term $\mu x.\mu y.F(x, y)$.

Let $G^\alpha$ has start variable $V_1,\{x\mapsto x,y\mapsto y\},\emptyset$, and contains rules:

$V_1,\{x\mapsto x,y\mapsto y\},\emptyset \Rightarrow \mu y.V_2,\{x\mapsto y,y\mapsto y\},\{y\}$

$V_2,\{x\mapsto y,y\mapsto y\},\{y\} \Rightarrow \mu y.V_3,\{x\mapsto y,y\mapsto y\},\{x\}$

$V_3,\{x\mapsto y,y\mapsto y\},\{x\} \Rightarrow F(V_4,\{x\mapsto y,y\mapsto y\},\{x\}), V_5,\{x\mapsto y,y\mapsto y\},\{x\}$
Step 2: $\alpha$-conversion

pick renaming: $V_{\sigma,\dagger} \Rightarrow \mu y(V'_{\sigma',\dagger'}) \in G$ if $V \Rightarrow \mu x(V') \in G$

where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example

Let $G$ have start variable $V_1$ and consist of the rules:

$V_1 \Rightarrow \mu x.V_2 \quad V_2 \Rightarrow \mu y.V_3 \quad V_3 \Rightarrow F(V_4, V_5) \quad V_4 \Rightarrow x \quad V_5 \Rightarrow y$

Note that $G$ generates the term $\mu x.\mu y.F(x, y)$.

Let $G^\alpha$ has start variable $V_1,\{x\mapsto x, y\mapsto y\},\varnothing$, and contains rules:

$V_1,\{x\mapsto x, y\mapsto y\},\varnothing \Rightarrow \mu y.V_2,\{x\mapsto y, y\mapsto y\},\{y\}$

$V_2,\{x\mapsto y, y\mapsto y\},\{y\} \Rightarrow \mu y.V_3,\{x\mapsto y, y\mapsto y\},\{x\}$

$V_3,\{x\mapsto y, y\mapsto y\},\{x\} \Rightarrow F(V_4,\{x\mapsto y, y\mapsto y\},\{x\}, V_5,\{x\mapsto y, y\mapsto y\},\{x\})$

$V_3,\{x\mapsto y, y\mapsto y\},\{x\} \Rightarrow F(V_4,\{x\mapsto y, y\mapsto y\},\{x\}, V_5,\{x\mapsto y, y\mapsto y\},\{x\})$
Step 2: $\alpha$-conversion

pick renaming: $V_\sigma,t \Rightarrow \mu y(V'_\sigma,t) \in G$ if $V \Rightarrow \mu x(V') \in G$
where $y \in B$, $\sigma' = \sigma[x \mapsto y]$, $t' = (t \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example

Let $G$ have start variable $V_1$ and consist of the rules:

$V_1 \Rightarrow \mu x.V_2 \quad V_2 \Rightarrow \mu y.V_3 \quad V_3 \Rightarrow F(V_4, V_5) \quad V_4 \Rightarrow x \quad V_5 \Rightarrow y$

Note that $G$ generates the term $\mu x.\mu y.F(x, y)$.

Let $G^\alpha$ has start variable $V_1,\{x\mapsto x, y\mapsto y\},\emptyset$, and contains rules:

$V_1,\{x\mapsto x, y\mapsto y\},\emptyset \Rightarrow \mu y.V_2,\{x\mapsto y, y\mapsto y\},\{y\}$
$V_2,\{x\mapsto y, y\mapsto y\},\{y\} \Rightarrow \mu y.V_3,\{x\mapsto y, y\mapsto y\},\{x\}$
$V_3,\{x\mapsto y, y\mapsto y\},\{x\} \Rightarrow F(V_4,\{x\mapsto y, y\mapsto y\},\{x\}, V_5,\{x\mapsto y, y\mapsto y\},\{x\})$
$V_3,\{x\mapsto y, y\mapsto y\},\{x\} \Rightarrow F(V_4,\{x\mapsto y, y\mapsto y\},\{x\}, V_5,\{x\mapsto y, y\mapsto y\},\{x\})$
$V_4,\{x\mapsto y, y\mapsto y\},\{x\} \Rightarrow \bot$
**Step 2: \(\alpha\)-conversion**

**pick renaming:** \(V_{\sigma, \dagger} \Rightarrow \mu y (V'_{\sigma', \dagger'}) \in G \)

if \(V \Rightarrow \mu x (V') \in G\)

where \(y \in \mathbb{B}, \sigma' = \sigma[x \mapsto y], \dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}\).

---

**Example**

Let \(G\) have start variable \(V_1\) and consist of the rules:

\[
\begin{align*}
V_1 & \Rightarrow \mu x . V_2 \\
V_2 & \Rightarrow \mu y . V_3 \\
V_3 & \Rightarrow F(V_4, V_5) \\
V_4 & \Rightarrow x \\
V_5 & \Rightarrow y
\end{align*}
\]

Note that \(G\) generates the term \(\mu x . \mu y . F(x, y)\).

Let \(G^\alpha\) has start variable \(V_1, \{x \mapsto x, y \mapsto y\}, \emptyset\), and contains rules:

\[
\begin{align*}
V_1, \{x \mapsto x, y \mapsto y\}, \emptyset & \Rightarrow \mu y . V_2, \{x \mapsto y, y \mapsto y\}, \{y\} \\
V_2, \{x \mapsto y, y \mapsto y\}, \{y\} & \Rightarrow \mu y . V_3, \{x \mapsto y, y \mapsto y\}, \{x\} \\
V_3, \{x \mapsto y, y \mapsto y\}, \{x\} & \Rightarrow F(V_4, \{x \mapsto y, y \mapsto y\}, \{x\}, V_5, \{x \mapsto y, y \mapsto y\}, \{x\}) \\
V_3, \{x \mapsto y, y \mapsto y\}, \{x\} & \Rightarrow F(V_4, \{x \mapsto y, y \mapsto y\}, \{x\}, V_5, \{x \mapsto y, y \mapsto y\}, \{x\}) \\
V_4, \{x \mapsto y, y \mapsto y\}, \{x\} & \Rightarrow \perp \\
V_5, \{x \mapsto y, y \mapsto y\}, \{x\} & \Rightarrow y
\end{align*}
\]
Theorem

The following problem is decidable:

- **Input:** two $\mu$-terms $M$ and $N$.
- **Answer:** are $M$ and $N$ convertible?

Proof.

The decision procedure proceeds in the following steps:
Step 3: deciding $\mu/\alpha$

Theorem

The following problem is decidable:

- **Input**: two $\mu$-terms $M$ and $N$.
- **Answer**: are $M$ and $N$ convertible?

Proof.

The decision procedure proceeds in the following steps:

1. take capture-avoiding $\mu$-pseudoterms $M' =_\alpha M$, $N' =_\alpha N$, 

---

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Step 3: deciding $=_{\mu/\alpha}$

**Theorem**

The following problem is decidable:

- Input: two $\mu$-terms $M$ and $N$.
- Answer: are $M$ and $N$ convertible?

**Proof.**

The decision procedure proceeds in the following steps:

1. take capture-avoiding $\mu$-pseudoterms $M' =_\alpha M$, $N' =_\alpha N$,
2. construct $G_{M'}$ and $G_{N'}$. 

Endrullis, Grabmayer, Klop, van Oostrom

On Equal $\mu$-Terms
Step 3: deciding $\equiv_{\mu/\alpha}$

Theorem

The following problem is decidable:

- **Input:** two $\mu$-terms $M$ and $N$.
- **Answer:** are $M$ and $N$ convertible?

Proof.

The decision procedure proceeds in the following steps:

1. take capture-avoiding $\mu$-pseudoterms $M' =_{\alpha} M$, $N' =_{\alpha} N$,
2. construct $G_{M'}$ and $G_{N'}$,
3. construct $G_{M'}^{\alpha}$ over the set of binders of $M'$ and $N'$, and
Theorem

The following problem is decidable:

- **Input:** two \( \mu \)-terms \( M \) and \( N \).
- **Answer:** are \( M \) and \( N \) convertible?

Proof.

The decision procedure proceeds in the following steps:

1. take capture-avoiding \( \mu \)-pseudoterms \( M' =_{\alpha} M \), \( N' =_{\alpha} N \),
2. construct \( G_{M'} \) and \( G_{N'} \),
3. construct \( G_{M'}^{\alpha} \) over the set of binders of \( M' \) and \( N' \), and
4. answer **yes** if \( L(G_{M'}^{\alpha}) \cap L(G_{N'}) \neq \emptyset \), and **no**, otherwise.
Overview

1. Weak $\mu$-equality

2. Avoiding $\alpha$-conversion in $\mu$-reductions

3. Decidability of $=_{\mu/\alpha}$ by a first-order proof

4. Decidability of $=_{\mu/\alpha}$ by a higher-order proof

5. Decidability of $=_{\mu/\alpha}$ using regular languages

6. Summary
We established **decidability of the weak $\mu$-equality problem** by:

- a proof using ‘first-order’ techniques:
  - characterising $\mu$-pseudoterms that **can be reduced without the need for $\alpha$-renaming**:
  - a **complete proof system** à la Coppo/Cardone for $=_{\mu/\alpha}$ on $\mu$-pseudoterms
  - showing finiteness of proof-search by establishing **finiteness of the subterm closure** for capture-avoiding $\mu$-terms

- a proof using ‘higher-order’ techniques

- another proof using ‘first-order’ techniques:
  - the set of reducts of $\mu$-pseudoterms form a **regular tree language**
  - weak $\mu$-equality **reduces to the emptiness problem for the intersection of regular tree languages**