On the Complexity of Translations of Lambda Calculus to Combinatory Logic

Łukasz Lachowski
lukasz.lachowski@tcs.uj.edu.pl

Theoretical Computer Science
Jagiellonian University

CLA
Novi Sad
26.05.2016

Overview

1. Complexity of the standard translation
   - Upper bound
   - Lower bound
   - Worst case predictions
2. Lower bound for any translation
3. Asymptotically optimal algorithm

Standard Translation

- λ-calculus $\to$ Combinatory Logic: a way to implement a lazy functional programming language.
- Standard Translation: algorithm which produces CL-terms that are extensionally equal to λ-terms.
- Complexity analysis?

Complexity

How big is a CL-term produced by the translation?

Input: a λ-term M of size n

Output: a CL-term T which is extensionally equal to M

Parameter: worst case size of T

Complexity

Following Wikipedia:
By 2002 it was $O(3^n)$.

Following Wikipedia:
By 2002 it was $O(3^n)$.
Improved to $\Theta(3^n)$.

Complexity

Following Wikipedia:
By 2002 it was $O(3^n)$.
Improved to $\Theta(3^n)$.

Our result: $\Theta(n^3)$.

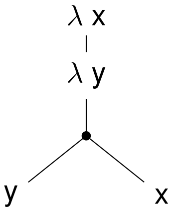


Figure: $\lambda xy.yx \in \Lambda$

$$
\begin{aligned}
|x| &= 1 \\
|MN| &= 1 + |M| + |N| \\
|\lambda x.M| &= 1 + |M|
\end{aligned}
$$

$$((\lambda x.M)N,\ M[x := N]) \in \beta$$

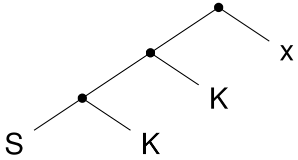


Figure: $(SKK)x \in \mathrm{CL}$

$$
\begin{aligned}
|x| &= 1 \\
|S| &= |K| = 1 \\
|MN| &= 1 + |M| + |N|
\end{aligned}
$$

$$
\begin{aligned}
(\mathbf{S}\, M\, N\, O,\ M\, O\, (N\, O)) &\in \mathbf{w} \\
(\mathbf{K}\, M\, N,\ M) &\in \mathbf{w}
\end{aligned}
$$

Simulating λ-abstraction in CL

$\lambda^* x : \mathrm{CL} \to \mathrm{CL}$

Motivation

- $(\lambda^* x.P)Q =_{\mathbf{w}} P[x := Q]$

Simulating λ-abstraction in CL

$\lambda^* x : \mathrm{CL} \to \mathrm{CL}$

Motivation

- $(\lambda^* x.P)Q =_{\mathbf{w}} P[x := Q]$

Bracket abstraction

$$\lambda^* x.x = SKK$$
$$\lambda^* x.P = KP \quad \text{if } x \notin \mathrm{FV}(P)$$
$$\lambda^* x.PQ = S(\lambda^* x.P)(\lambda^* x.Q) \quad \text{otherwise}$$

$$\mathbf{K}Mx \to_{\mathbf{w}} M$$
$$\mathbf{S}MNx \to_{\mathbf{w}} Mx(Nx)$$

Translation

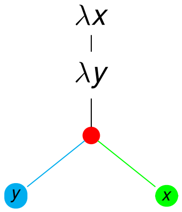
Standard Translation

$[\quad] : \Lambda \to \mathrm{CL}$

$$\begin{aligned} [\, x \,] &= x \\ [\, MN \,] &= [\, M \,]\,[\, N \,] \\ [\, \lambda x.M \,] &= \lambda^* x.[\, M \,] \end{aligned}$$

$$M =_{\beta\eta} N \;\Leftrightarrow\; [\, M \,] =_{\mathrm{ext}} [\, N \,]$$

Upper bound

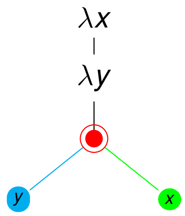


Colored term $\lambda xy.yx$

Color preserving bracket abstraction

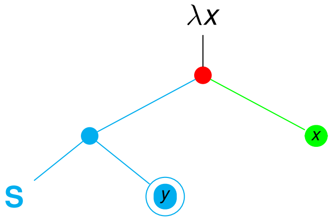
$$\begin{aligned}
\lambda^* x.x &= SKK \\
\lambda^* x.P &= KP \quad \text{if } x \notin \mathrm{FV}(P) \\
\lambda^* x.(PQ) &= (S(\lambda^* x.P)(\lambda^* x.Q)) \quad \text{otherwise}
\end{aligned}$$

Example



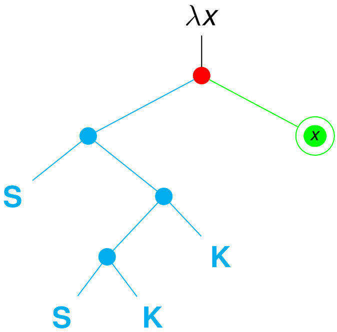
$$\lambda^* y.(yx) = (S(\lambda^* y.y)(\lambda^* y.x))$$

Example



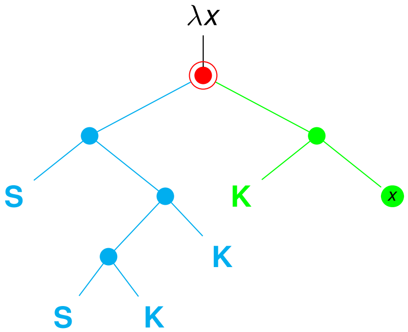
$$\lambda^* y.y = SKK$$

Example



$$\lambda^* y.x = Kx$$

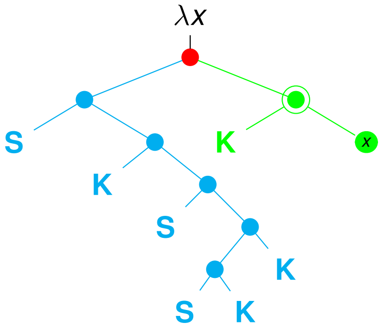
Example



$$\lambda^* x.((S(SKK)) \quad (Kx)) = ((S\ (\lambda^* x.S(SKK))) \quad (\lambda^* x.Kx))$$

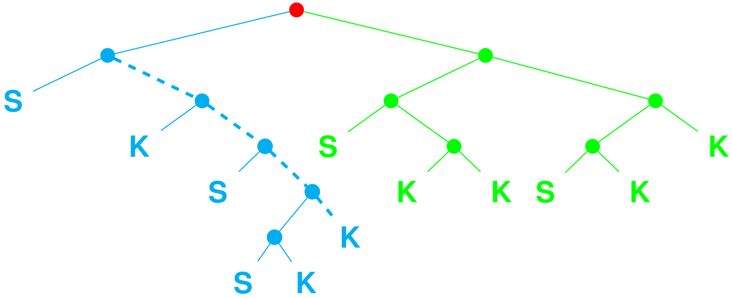
$$\lambda^* x.S(SKK) = K(S(SKK))$$

Example

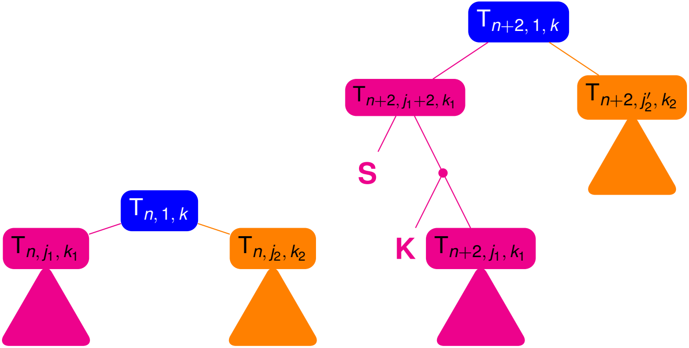


$$\lambda^* x.(Kx) = (S(\lambda^* x.K)(\lambda^* x.x))$$
$$\lambda^* x.K = KK$$
$$\lambda^* x.x = SKK$$

Colored right leaning path

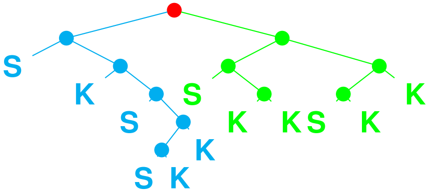


$$[\ \lambda xy.yx\ ] = \lambda^* x.\lambda^* y.yx$$



(a) Term M (b) $\lambda^* x.M$

Figure: A term before and after removal of a variable which occurs only in its right sub-term.



$$[\ \lambda xy.yx\ ] = \lambda^* x.\lambda^* y.yx$$

Theorem

$M \in \Lambda$, $|M| = n$, for $[\ M\ ]$ we have:

1. *For every color its right leaning path is of length $O(n)$.*
2. *For every color there is $O(n^2)$ nodes of that color.*
3. *There is at most n different colors.*
4. $|\,[\ M\ ]\,| = O(n^3)$.

Lower bound

$$|\ [\ \lambda x_1 \dots x_k . x_k \dots x_1\ ]\ | = \frac{2}{81} n^3 + \frac{8}{27} n^2 + \frac{80}{27} n - \frac{187}{81}$$

where $n = |\lambda x_1 \dots x_k . x_k \dots x_1| = 3k - 1$.

Result

Theorem

For every size n the worst case size of a translation of a λ-term of size n is of order $\Theta(n^3)$.

Worst case terms

$$| \; [\; \lambda x_1 \ldots x_k . x_k \ldots x_1 \;] \; | = \frac{2}{81} n^3 + \frac{8}{27} n^2 + \frac{80}{27} n - \frac{187}{81}$$

where $n = |\lambda x_1 \ldots x_k . x_k \ldots x_1| = 3k - 1$.

Size	Terms scheme	Expected size
$n = 3k - 1$	$\lambda x_1 \ldots x_k \, . \, x_1 \ldots x_k$	$\frac{2}{81} n^3 + \frac{14}{27} n^2 + \frac{74}{27} n - \frac{223}{81}$
$n = 3k$	$\lambda x_1 \ldots x_k x_{k+1} \, . \, x_1 \ldots x_k$	$\frac{2}{81} n^3 + \frac{4}{9} n^2 + \frac{28}{9} n - 3$
$n = 3k + 1$	$\lambda x_1 \ldots x_k \, . \, x_1 \ldots x_k x_1$	$\frac{2}{81} n^3 + \frac{16}{27} n^2 + \frac{68}{27} n - \frac{659}{81}$

Table: Predictions of the worst case terms for the standard translation.

Lower bound

First idea: We count how many combinators we need to injectively express all λ-terms of a given size?

Lower bound

First idea: We count how many combinators we need to injectively express all λ-terms of a given size?
Sounds easy, but...

- We do not know a closed form formula for the number of λ-terms of a given size using this notion of size.

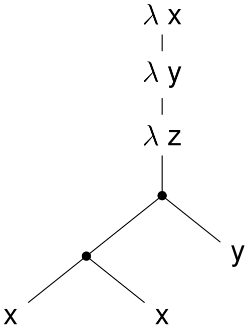
Lower bound

First idea: We count how many combinators we need to injectively express all λ-terms of a given size?
Sounds easy, but...

- We do not know a closed form formula for the number of λ-terms of a given size using this notion of size.
- Why should a translation algorithm be injective?

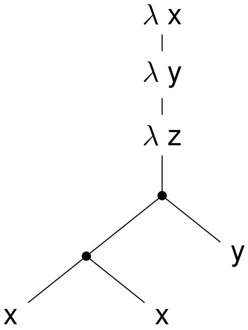
Lower bound

Consider a set of terms of the form $\lambda x_1 \dots x_n . T$ where T has size $|T| = s = 2n - 1$ and is built only using $x_1, \dots, x_n$. Hence, T is a binary tree with n leaves. We denote it by L.



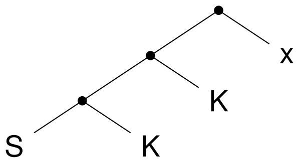
Lower bound

Consider a set of terms of the form $\lambda x_1 \ldots x_n . T$ where T has size $|T| = s = 2n - 1$ and is built only using $x_1, \ldots, x_n$. Hence, T is a binary tree with n leaves. We denote it by L.



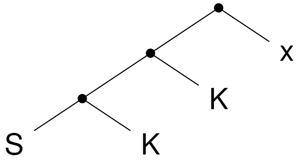
$$e := |L| = n^n \cdot Catalan_{n-1}$$

Lower bound



- The number of different *SK*-combinators of size $2s - 1$ is $2^s \cdot Catalan_{s-1}$.

Lower bound



- The number of different SK-combinators of size $2s - 1$ is $2^s \cdot Catalan_{s-1}$.
- Let $m := \max\{|[\ T\ ]| : \ T \in L\}$. The number of combinators of size $\leq m$ is

$$z := \sum_{i=1}^{(m+1)/2} 2^i \cdot Catalan_{i-1}.$$

Lower bound

Clearly $z \geq e$. Each term of the form $\lambda x_1 \dots x_n . T$ where T is built only using $x_1, \dots, x_n$ has a different translation, otherwise a translation algorithm produces terms that are not extensionally equal. Let m be the size of the biggest translation among these terms and let l denotes the number of leaves of that term ($m := \max\{|[\ T\ ]| : \ T \in L\} = 2l - 1$).

Lower bound

Clearly $z \geq e$. Each term of the form $\lambda x_1 \dots x_n.T$ where T is built only using $x_1, \dots, x_n$ has a different translation, otherwise a translation algorithm produces terms that are not extensionally equal. Let m be the size of the biggest translation among these terms and let l denotes the number of leaves of that term ($m := \max\{|[\, T\,]| : \ T \in L\} = 2l - 1$).

$2^{n-1} \leq Catalan_n < 4^n$

$$e := |L| = n^n \cdot Catalan_{n-1}$$

$$z := \sum_{i=1}^{l} 2^i \cdot Catalan_{i-1}$$

Lower bound

Clearly $z \geq e$. Each term of the form $\lambda x_1 \dots x_n . T$ where T is built only using $x_1, \dots, x_n$ has a different translation, otherwise a translation algorithm produces terms that are not extensionally equal. Let m be the size of the biggest translation among these terms and let l denotes the number of leaves of that term ($m := \max \{ | [\ T\ ] | : \ T \in L \} = 2l - 1$).

$2^{n-1} \leq Catalan_n < 4^n$

$$e := |L| = n^n \cdot Catalan_{n-1} \qquad \geq n^n \cdot 2^{n-2}$$

$$z := \sum_{i=1}^{l} 2^i \cdot Catalan_{i-1}$$

Lower bound

Clearly $z \geq e$. Each term of the form $\lambda x_1 \dots x_n.T$ where T is built only using $x_1, \dots, x_n$ has a different translation, otherwise a translation algorithm produces terms that are not extensionally equal. Let m be the size of the biggest translation among these terms and let l denotes the number of leaves of that term ($m := \max\{|[\ T\ ]| :\ T \in L\} = 2l - 1$).

$2^{n-1} \leq Catalan_n < 4^n$

$$e := |L| = n^n \cdot Catalan_{n-1} \geq n^n \cdot 2^{n-2}$$

$$z := \sum_{i=1}^{l} 2^i \cdot Catalan_{i-1} \leq \sum_{i=1}^{l} 2^i \cdot 4^{i-1} \leq 2^l \cdot \sum_{i=1}^{l} 4^{i-1} \leq 8^l = 8^{(m+1)/2}$$

Lower bound

$$z \geq e$$

Lower bound

$$n^n \cdot 2^{n-2} \geq \quad z \geq e \quad \geq 8^{(m+1)/2}$$

Lower bound

$$n^n \cdot 2^{n-2} \geq \quad z \geq e \quad \geq 8^{(m+1)/2}$$

$$n \cdot \log n + (n-2) \cdot \log 2 \geq (m+1)/2 \cdot \log 8$$

Lower bound

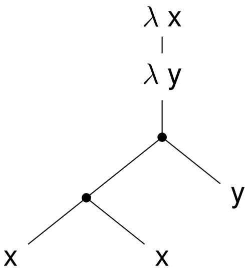
$$n^n \cdot 2^{n-2} \geq \quad z \geq e \quad \geq 8^{(m+1)/2}$$

$$n \cdot \log n + (n-2) \cdot \log 2 \geq (m+1)/2 \cdot \log 8$$

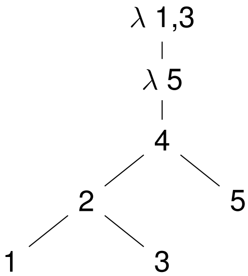
Substituting $(s+1)/2$ for n, where s is the size of a term from the set L, we have

$$m = \Omega(s \cdot \log s)$$

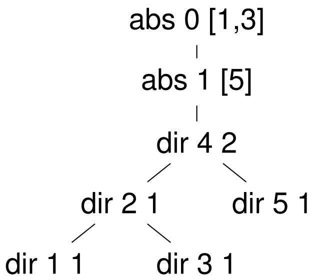
Translation



Translation



Translation



Translation

```
fix abs . λ absVars nodes term .
  if (absVars = 0) then
    λ var . term var nodes
  else
    λ var varNodes .
      abs (absVars−1) nodes (term var varNodes)
```

Translation

```
fix dir . λ nodeId absVars left right .
  if (absVars = 0) then (left right)
  else λ var varNodes .
    if (varNodes = []) then
      dir nodeId absVars left right
    else if (varNodes = [nodeId]) then var
         else
           dir nodeId (absVars−1)
             (forwardVars left var varNodes)
             (forwardVars right var varNodes)
```

Va'esse deireádh aep eigean - Geralt of Rivia

Thank you!