Lambda terms counting and leader-switching tours.

Jakub Kozik

Theoretical Computer Science
Jagiellonian University

5 June 2009
Computational Logic and Applications

Lambda terms

Size of a lambda term is a number of its **internal** nodes.
$L(n)$ - number of lambda terms of size n.

objective

Asymptotic approximation for $L(n)$.

Trivial bounds - exponential gap

$$(4-\varepsilon)^{n-\frac{n}{Log(n)}}\left(\frac{n}{Log(n)}\right)^{n-\frac{n}{Log(n)}} \leq L(n) \leq (5.83)^n\left(\frac{n}{Log(n)}\right)^{n-\frac{n}{Log(n)}}$$

Lambda terms

Size of a lambda term is a number of its **internal** nodes.
$L(n)$ - number of lambda terms of size n.

objective

Asymptotic approximation for $L(n)$.

Trivial bounds - exponential gap

$$(4-\varepsilon)^{n-\frac{n}{Log(n)}}\left(\frac{n}{Log(n)}\right)^{n-\frac{n}{Log(n)}} \leq L(n) \leq (5.83)^n\left(\frac{n}{Log(n)}\right)^{n-\frac{n}{Log(n)}}$$

Lambda terms

Size of a lambda term is a number of its **internal** nodes.
$L(n)$ - number of lambda terms of size n.

objective

Asymptotic approximation for $L(n)$.

Trivial bounds - exponential gap

$$(4-\varepsilon)^{n-\frac{n}{Log(n)}}\left(\frac{n}{Log(n)}\right)^{n-\frac{n}{Log(n)}} \leq L(n) \leq (5.83)^n\left(\frac{n}{Log(n)}\right)^{n-\frac{n}{Log(n)}}$$

L_1 - set terms in which:

all binding lambdas lie on one path.

Observation

$$c_1 \cdot L(n) \leq L_1(n) \leq L(n)$$

for some $c_1 \geq 1/2$.

L_2 - set terms in which:

All ~~binding~~ lambdas lie on one path.

Observation

$$c_2 \cdot L_1(n)/n \leq L_1(n-1) \leq L_2(n) \leq L_1(n)$$

L_1 - set terms in which:

all binding lambdas lie on one path.

Observation

$$c_1 \cdot L(n) \leq L_1(n) \leq L(n)$$

for some $c_1 \geq 1/2$.

L_2 - set terms in which:

All ~~binding~~ lambdas lie on one path.

Observation

$$c_2 \cdot L_1(n)/n \leq L_1(n-1) \leq L_2(n) \leq L_1(n)$$

L_1 - set terms in which:

all binding lambdas lie on one path.

Observation

$$c_1 \cdot L(n) \leq L_1(n) \leq L(n)$$

for some $c_1 \geq 1/2$.

L_2 - set terms in which:

All ~~binding~~ lambdas lie on one path.

Observation

$$c_2 \cdot L_1(n)/n \leq L_1(n-1) \leq L_2(n) \leq L_1(n)$$

L_1 - set terms in which:

all binding lambdas lie on one path.

Observation

$$c_1 \cdot L(n) \leq L_1(n) \leq L(n)$$

for some $c_1 \geq 1/2$.

L_2 - set terms in which:

All ~~binding~~ lambdas lie on one path.

Observation

$$c_2 \cdot L_1(n)/n \leq L_1(n-1) \leq L_2(n) \leq L_1(n)$$

Structural decomposition

Fix $k \in \mathbb{N}$.
Every term $l \in L_2$ with k-lambdas decomposes into a sequence

$$(t'_1, t'_2, t'_3, \ldots, t'_{k-1}, t_k)$$

where

- t'_j is **pointed** binary tree with leaves labelled by numbers $\{1, \ldots, j\}$
- t_k is binary tree with leaves labelled by numbers $\{1, \ldots, k\}$

$h_k(z)$ - g.f. enumerating terms from L_2 with k lambdas:

$$h_k(z) = \frac{t(kz)}{z} \prod_{j=1,\ldots,k-1} z \cdot t'(jz)$$

where $t(z)$ -g.f. for binary trees.

Structural decomposition

Fix $k \in \mathbb{N}$.
Every term $l \in L_2$ with k-lambdas decomposes into a sequence

$$(t'_1, t'_2, t'_3, \ldots, t'_{k-1}, t_k)$$

where

- t'_j is **pointed** binary tree with leaves labelled by numbers $\{1, \ldots, j\}$
- t_k is binary tree with leaves labelled by numbers $\{1, \ldots, k\}$

$h_k(z)$ - g.f. enumerating terms from L_2 with k lambdas:

$$h_k(z) = \frac{t(kz)}{z} \prod_{j=1,\ldots,k-1} z \cdot t'(jz)$$

where $t(z)$ -g.f. for binary trees.

Structural decomposition

Fix $k \in \mathbb{N}$.
Every term $l \in L_2$ with k-lambdas decomposes into a sequence

$$(t'_1, t'_2, t'_3, \ldots, t'_{k-1}, t_k)$$

where

- t'_j is **pointed** binary tree with leaves labelled by numbers $\{1, \ldots, j\}$
- t_k is binary tree with leaves labelled by numbers $\{1, \ldots, k\}$

$h_k(z)$ - g.f. enumerating terms from L_2 with k lambdas:

$$h_k(z) = \frac{t(kz)}{z} \prod_{j=1,\ldots,k-1} z \cdot t'(jz)$$

where $t(z)$ -g.f. for binary trees.

Structural decomposition

Fix $k \in \mathbb{N}$.
Every term $l \in L_2$ with k-lambdas decomposes into a sequence

$$(t'_1, t'_2, t'_3, \ldots, t'_{k-1}, t_k)$$

where

- t'_j is **pointed** binary tree with leaves labelled by numbers $\{1, \ldots, j\}$
- t_k is binary tree with leaves labelled by numbers $\{1, \ldots, k\}$

$h_k(z)$ - g.f. enumerating terms from L_2 with k lambdas:

$$h_k(z) = \frac{t(kz)}{z} \prod_{j=1,\ldots,k-1} z \cdot t'(jz)$$

where $t(z)$ -g.f. for binary trees.

Why not count pointed L_2 terms?

$s_k(z)$ - g.f. enumerating something with k lambdas:

$$\hat{h}_k(z) = \frac{k}{z} \prod_{j=1,\dots,k} z \cdot t'(jz)$$

where $t(z)$ -g.f. for binary trees.

Observation

$$[z^n]h_k(z) \leq [z^n]\hat{h}_k(z) \leq n \cdot [z^n]h_k(z)$$

Even simpler

$$s_k(z) = \prod_{j=1,\dots,k} z \cdot t'(jz)$$

Why not count pointed L_2 terms?

$s_k(z)$ - g.f. enumerating something with k lambdas:

$$\hat{h}_k(z) = \frac{k}{z} \prod_{j=1,\dots,k} z \cdot t'(jz)$$

where $t(z)$ -g.f. for binary trees.

Observation

$$[z^n]h_k(z) \leq [z^n]\hat{h}_k(z) \leq n \cdot [z^n]h_k(z)$$

Even simpler

$$s_k(z) = \prod_{j=1,\dots,k} z \cdot t'(jz)$$

Why not count pointed L_2 terms?

$s_k(z)$ - g.f. enumerating something with k lambdas:

$$\hat{h}_k(z) = \frac{k}{z} \prod_{j=1,\ldots,k} z \cdot t'(jz)$$

where $t(z)$ -g.f. for binary trees.

Observation

$$[z^n]h_k(z) \leq [z^n]\hat{h}_k(z) \leq n \cdot [z^n]h_k(z)$$

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} z \cdot t'(jz)$$

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} \frac{z}{\sqrt{1-4\cdot j\cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n+k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - go down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\dots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\dots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\dots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\dots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Leader switching tours

- You start at level 0 with one Sherpa.
- You have $2n + k$ days to spend in the mountains.
- Each day you can
  - go up (one level),
  - do down (one level),
  - stay in place and recruit one Sherpa, if you are at level 0.
- Each time you go up, you have to designate one Sherpa to pave the way.
- You have to end on the level 0 with exactly k Sherpas.

Even simpler

$$s_k(z) = \prod_{j=1,\ldots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Something similar:

$$g_k(z) = \prod_{j=1,\ldots,k} \frac{z}{1 - j \cdot z}$$

We have:

$$[z^n]g_k(z) = S2(n, k)$$

Even simpler

$$s_k(z) = \prod_{j=1,\dots,k} \frac{z}{\sqrt{1 - 4 \cdot j \cdot z}}$$

Something similar:

$$g_k(z) = \prod_{j=1,\dots,k} \frac{z}{1 - j \cdot z}$$

We have:

$$[z^n]g_k(z) = S2(n, k)$$

Suppose that k is even.

$$s_k(z) =$$

$$\left(\frac{z}{\sqrt{1-4z}} \cdot \frac{z}{\sqrt{1-8z}}\right)\left(\frac{z}{\sqrt{1-12z}} \cdot \frac{z}{\sqrt{1-16z}}\right)\cdots$$

$$\cdots\left(\frac{z}{\sqrt{1-4(k-1)z}} \cdot \frac{z}{\sqrt{1-4kz}}\right)$$

$$\leq \left(\frac{z}{\sqrt{1-8z}} \cdot \frac{z}{\sqrt{1-8z}}\right)\left(\frac{z}{\sqrt{1-16z}} \cdot \frac{z}{\sqrt{1-16z}}\right)\cdots$$

$$\cdots\left(\frac{z}{\sqrt{1-4kz}} \cdot \frac{z}{\sqrt{1-4kz}}\right)$$

$$= \prod_{j=1,\ldots,k/2} \frac{z^2}{1-8jz} = \left(\frac{z}{8}\right)^{k/2} \cdot g_{k/2}(8z)$$

Suppose that k is even.

$$s_k(z) = \left(\frac{z}{\sqrt{1-4z}} \cdot \frac{z}{\sqrt{1-8z}}\right)\left(\frac{z}{\sqrt{1-12z}} \cdot \frac{z}{\sqrt{1-16z}}\right) \cdots \cdots \left(\frac{z}{\sqrt{1-4(k-1)z}} \cdot \frac{z}{\sqrt{1-4kz}}\right)$$

$$\leq \left(\frac{z}{\sqrt{1-8z}} \cdot \frac{z}{\sqrt{1-8z}}\right)\left(\frac{z}{\sqrt{1-16z}} \cdot \frac{z}{\sqrt{1-16z}}\right) \cdots \cdots \left(\frac{z}{\sqrt{1-4kz}} \cdot \frac{z}{\sqrt{1-4kz}}\right)$$

$$= \prod_{j=1,\ldots,k/2} \frac{z^2}{1-8jz} = \left(\frac{z}{8}\right)^{k/2} \cdot g_{k/2}(8z)$$

Therefore

$$[z^n]s_k(z) \leq 8^{n-k} \cdot S2(N - \frac{k}{2}, \frac{k}{2})$$

Similarly, we get

$$[z^n]s_k(z) \geq \frac{1}{lp(n)} 8^{n-k} \cdot S2(N - \frac{k}{2}, \frac{k}{2})$$

for some polynomial lp.

Therefore

$$[z^n]s_k(z) \leq 8^{n-k} \cdot S2(N - \frac{k}{2}, \frac{k}{2})$$

Similarly, we get

$$[z^n]s_k(z) \geq \frac{1}{lp(n)} 8^{n-k} \cdot S2(N - \frac{k}{2}, \frac{k}{2})$$

for some polynomial lp.

Let

$$F(n,k) = 8^{n-k} \cdot S2(N - \frac{k}{2}, \frac{k}{2})$$

Theorem

There exists polynomials lp, up such that:

$$\frac{1}{lp(n)} \cdot F(n,k(n)) \leq L(n) \leq up(n) \cdot F(n,k(n))$$

where $k(n)$ is the number k from $\{1,\ldots,n\}$ that maximizes $F(n,k)$.

Let

$$F(n,k) = 8^{n-k} \cdot S2(N - \frac{k}{2}, \frac{k}{2})$$

Theorem

There exists polynomials lp, up such that:

$$\frac{1}{lp(n)} \cdot F(n,k(n)) \le L(n) \le up(n) \cdot F(n,k(n))$$

where $k(n)$ is the number k from $\{1,\ldots,n\}$ that maximizes $F(n,k)$.