Quantitative aspects of linear and affine closed lambda terms

Pierre Lescanne

École normale supérieure de Lyon

On ideas of Olivier Bodini

1. Affine, Linear, Closed
2. De Bruijn indices
3. Swiss Cheese
4. Counting closed linear terms
5. Counting closed affine terms
6. Generating functions
7. Effective computations
8. Generating terms
9. Other results

Lambda Terms

Lambda terms are abstractions for representing functions

A λ-term is

- *a variable x or*
- *an application $M\,N$ or*
- *an abstraction $\lambda x.M$.*

For instance,

$\lambda x.(x\,x)$
$\lambda x.\lambda y.x$
$\lambda x(x\,y)$
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$

Lambda Terms

Lambda terms are abstractions for representing functions

A λ-term is

- *a variable x or*
- *an application $M\,N$ or*
- *an abstraction $\lambda x.M$.*

For instance,

$\lambda x.(x\,x)$	is the same as	$\lambda y.(y\,y)$
$\lambda x.\lambda y.x$	is the same as	$\lambda y.\lambda xy$
$\lambda x(x\,y)$	is the same as	$\lambda z.(z\,y)$
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	is the same as	$(\lambda z.z)\,(\lambda x.\lambda z(z\,x))$

Lambda Terms

Lambda terms are abstractions for representing functions

A λ-term is

- *a variable x or*
- *an application $M\,N$ or*
- *an abstraction $\lambda x.M$.*

For instance,

$\lambda x.(x\,x)$	is the same as	$\lambda y.(y\,y)$
$\lambda x.\lambda y.x$	is the same as	$\lambda y.\lambda xy$
$\lambda x(x\,y)$	is the same as	$\lambda z.(z\,y)$
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	is the same as	$(\lambda z.z)\,(\lambda x.\lambda z(z\,x))$

We count λ-terms up-to α-conversion.

Closed terms

A variable x is bound if it appears in the scope of λx.
Otherwise x is free

In $\lambda x(x\, y)$

- x is bound,
- y is free.

Closed terms

A variable x is bound if it appears in the scope of λx.
Otherwise x is free

In $\lambda x(x\,y)$

- x is bound,
- y is free.

Definition

A term is closed if all its variables are bound.

$\lambda x.(x\,x)$
$\lambda x.\lambda y.x$
$\lambda x(x\,y)$
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$

Closed terms

A variable x is bound if it appears in the scope of λx.
Otherwise x is free

In $\lambda x(x\,y)$

- x is bound,
- y is free.

Definition

A term is closed if all its variables are bound.

$\lambda x.(x\,x)$ closed
$\lambda x.\lambda y.x$
$\lambda x(x\,y)$
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$

Closed terms

A variable x is bound if it appears in the scope of λx.
Otherwise x is free

In $\lambda x(x\,y)$

- x is bound,
- y is free.

Definition

A term is closed if all its variables are bound.

$\lambda x.(x\,x)$	closed
$\lambda x.\lambda y.x$	closed
$\lambda x(x\,y)$	
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	

Closed terms

A variable x is bound if it appears in the scope of λx.
Otherwise x is free

In $\lambda x (x\, y)$

- x is bound,
- y is free.

Definition

A term is closed if all its variables are bound.

$\lambda x.(x\, x)$	closed
$\lambda x.\lambda y.x$	closed
$\lambda x(x\, y)$	open
$(\lambda x.x)\,(\lambda x.\lambda y(y\, x))$	

Closed terms

A variable x is bound if it appears in the scope of λx.
Otherwise x is free

In $\lambda x(x\,y)$

- x is bound,
- y is free.

Definition

A term is closed if all its variables are bound.

$\lambda x.(x\,x)$	closed
$\lambda x.\lambda y.x$	closed
$\lambda x(x\,y)$	open
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	closed

Closed terms

A variable x is bound if it appears in the scope of λx.
Otherwise x is free

In $\lambda x(x\,y)$

- x is bound,
- y is free.

Definition

A term is closed if all its variables are bound.

$\lambda x.(x\,x)$	closed
$\lambda x.\lambda y.x$	closed
$\lambda x(x\,y)$	open
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	closed

We will count closed λ-terms.

Affine terms

Definition

A term is affine if each λ binds at most one variable.

Affine terms

Definition

A term is affine if each λ binds at most one variable.

$\lambda x.(x\, x)$
$\lambda x.\lambda y.x$
$\lambda x(x\, y)$
$(\lambda x.x)\, (\lambda x.\lambda y(y\, x))$

Affine terms

Definition

A term is affine if each λ binds at most one variable.

$\lambda x.(x\,x)$	no
$\lambda x.\lambda y.x$	
$\lambda x(x\,y)$	
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	

Affine terms

Definition

A term is affine if each λ binds at most one variable.

$\lambda x.(x\, x)$	no
$\lambda x.\lambda y.x$	yes
$\lambda x(x\, y)$	
$(\lambda x.x)\,(\lambda x.\lambda y(y\, x))$	

Affine terms

Definition

A term is affine if each λ binds at most one variable.

$\lambda x.(x\,x)$	no
$\lambda x.\lambda y.x$	yes
$\lambda x(x\,y)$	yes
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	

Affine terms

Definition

A term is affine if each λ binds at most one variable.

$\lambda x.(x\, x)$	no
$\lambda x.\lambda y.x$	yes
$\lambda x(x\, y)$	yes
$(\lambda x.x)\,(\lambda x.\lambda y(y\, x))$	yes

Affine terms

Definition

A term is affine if each λ binds at most one variable.

$\lambda x.(x\,x)$	no
$\lambda x.\lambda y.x$	yes
$\lambda x(x\,y)$	yes
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	yes

Affine terms are simply typable

Linear terms

Definition

A term is linear if

1. it is affine and
2. moreover each λ binds at least one variable (λI terms)

Each λ binds one and only one variable.

Linear terms

Definition

A term is linear if

1. it is affine and
2. moreover each λ binds at least one variable (λIterms)

Each λ binds one and only one variable.

$\lambda x.(x\,x)$
$\lambda x.\lambda y.x$
$\lambda x(x\,y)$
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$

Linear terms

Definition

A term is linear if

1. it is affine and
2. moreover each λ binds at least one variable (λIterms)

Each λ binds one and only one variable.

$\lambda x.(x\, x)$ no

$\lambda x.\lambda y.x$

$\lambda x(x\, y)$

$(\lambda x.x)\,(\lambda x.\lambda y(y\, x))$

Linear terms

Definition

A term is linear if

1. it is affine and
2. moreover each λ binds at least one variable (λIterms)

Each λ binds one and only one variable.

$\lambda x.(x\,x)$	no
$\lambda x.\lambda y.x$	no
$\lambda x(x\,y)$	
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	

Linear terms

Definition

A term is linear if

1. it is affine and
2. moreover each λ binds at least one variable (λIterms)

Each λ binds one and only one variable.

$\lambda x.(x\, x)$	no
$\lambda x.\lambda y.x$	no
$\lambda x(x\, y)$	yes
$(\lambda x.x)\,(\lambda x.\lambda y(y\, x))$	

Linear terms

Definition

A term is linear if

1. it is affine and
2. moreover each λ binds at least one variable (λIterms)

Each λ binds one and only one variable.

$\lambda x.(x\,x)$	no
$\lambda x.\lambda y.x$	no
$\lambda x(x\,y)$	yes
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	yes

1 Affine, Linear, Closed

2 De Bruijn indices

3 Swiss Cheese

4 Counting closed linear terms

5 Counting closed affine terms

6 Generating functions

7 Effective computations

8 Generating terms

9 Other results

De Bruijn indices

We want to count representatives of equivalence classes of λ-terms modulo α conversion.

Variables are replaced by natural numbers.

If x is replaced by n if to reach the binder λx of x
one crosses n λ .

De Bruijn indices

We want to count representatives of equivalence classes of λ-terms modulo α conversion.

Variables are replaced by natural numbers.

If x is replaced by n if to reach the binder λx of x one crosses n λ .

$\lambda x.(x\, x)$
$\lambda x.\lambda y.x$
$\lambda x(x\, y)$
$(\lambda x.x)\,(\lambda x.\lambda y(y\, x))$

De Bruijn indices

We want to count representatives of equivalence classes of λ-terms modulo α conversion.

Variables are replaced by natural numbers.

If x is replaced by n if to reach the binder λx of x one crosses n λ .

$\lambda x.(x\,x)$	$\lambda(0\ 0)$
$\lambda x.\lambda y.x$	
$\lambda x(x\,y)$	
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	

De Bruijn indices

We want to count representatives of equivalence classes of λ-terms modulo α conversion.

Variables are replaced by natural numbers.

If x is replaced by n if to reach the binder λx of x
one crosses n λ .

$\lambda x.(x\,x)$	$\lambda(0\ 0)$
$\lambda x.\lambda y.x$	$\lambda\lambda 1$
$\lambda x(x\,y)$	
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	

De Bruijn indices

We want to count representatives of equivalence classes of λ-terms modulo α conversion.

Variables are replaced by natural numbers.

If x is replaced by n if to reach the binder λx of x
one crosses n λ .

$\lambda x.(x\,x)$	$\lambda(0\ 0)$
$\lambda x.\lambda y.x$	$\lambda\lambda 1$
$\lambda x(x\,y)$	???
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	

De Bruijn indices

We want to count representatives of equivalence classes of λ-terms modulo α conversion.

Variables are replaced by natural numbers.

If x is replaced by n if to reach the binder λx of x
one crosses n λ .

$\lambda x.(x\,x)$	$\lambda(0\ 0)$
$\lambda x.\lambda y.x$	$\lambda\lambda 1$
$\lambda x(x\,y)$	???
$(\lambda x.x)\,(\lambda x.\lambda y(y\,x))$	$(\lambda 0)\ (\lambda\lambda 0\ 1)$

Size of terms

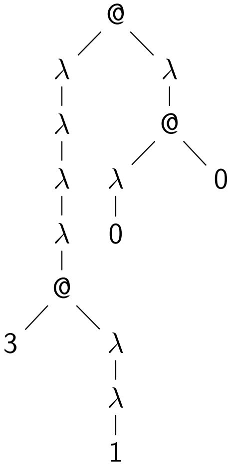
- *Abstractions* λ are of size 1.
- *Applications* are of size 0.

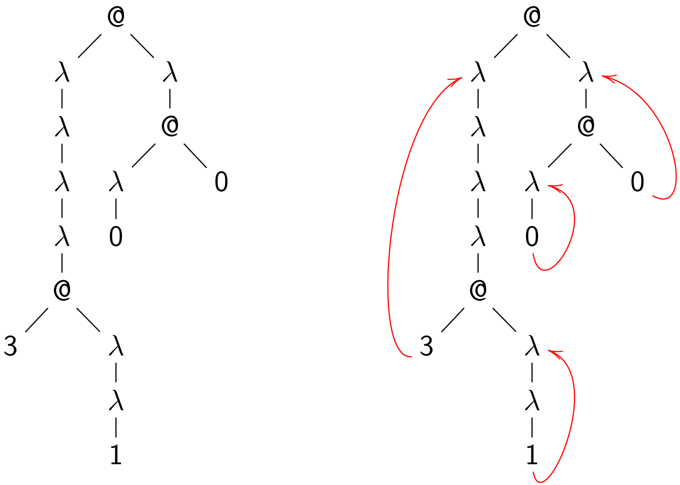
Three methods for counting de Bruijn indices:

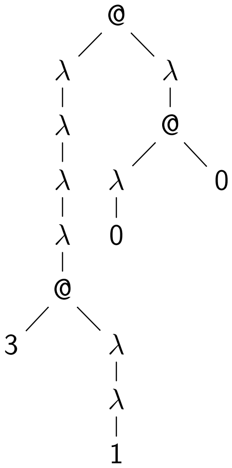
- De Bruijn indices have size 0,
- De Bruijn indices have size 1
- De Bruijn indices n have size $n+1$

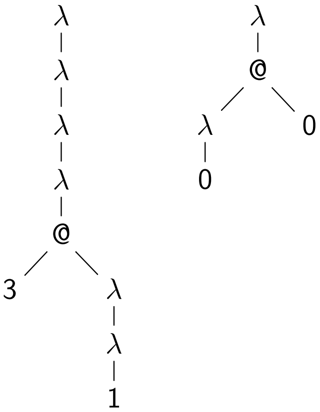
	Variable size 0	Variable size 1	natural size
$\lambda(0\ 0)$	2	4	4
$\lambda\lambda 1$	2	3	4
$(\lambda 0)\,(\lambda\lambda(0\ 1))$	5	8	9

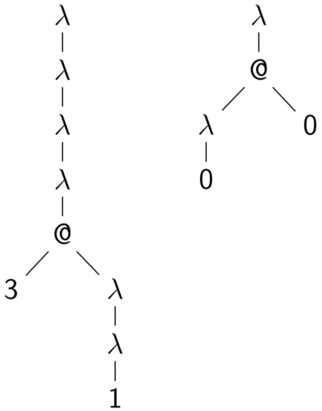
1 Affine, Linear, Closed
2 De Bruijn indices
3 Swiss Cheese
4 Counting closed linear terms
5 Counting closed affine terms
6 Generating functions
7 Effective computations
8 Generating terms
9 Other results

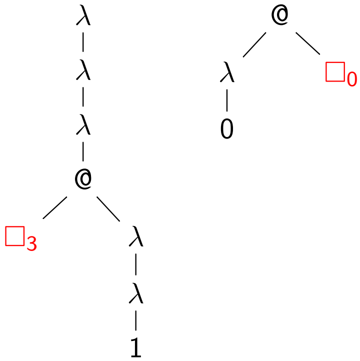

@
λ λ
λ @
λ λ 0
λ 0
@
3 λ
λ
1

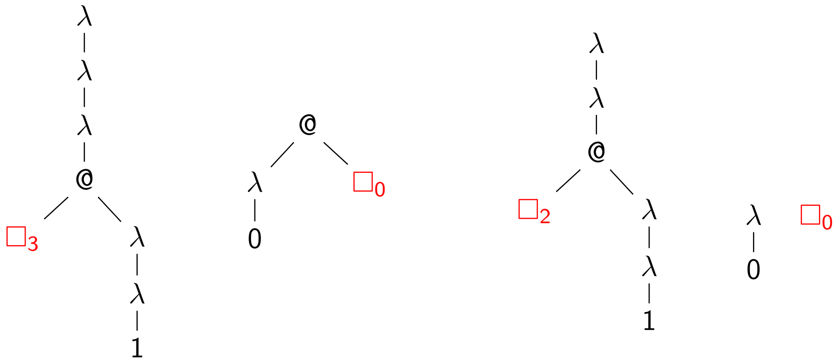


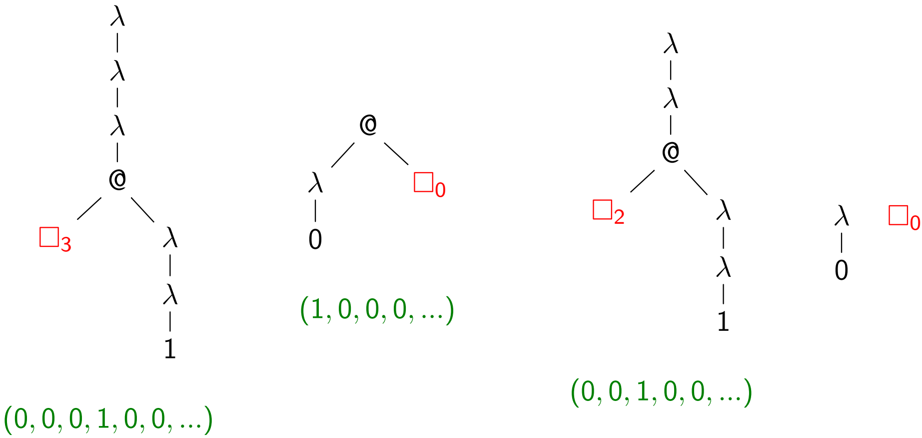








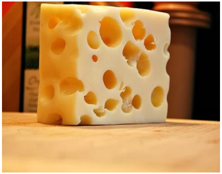



λ
λ
λ
@
□3
λ
λ
1
(0, 0, 0, 1, 0, 0, ...)
@
λ
□0
0
(1, 0, 0, 0, ...)
λ
λ
@
□2
λ
λ
1
(0, 0, 1, 0, 0, ...)
λ
□0
0

Swiss Cheeses

To count closed affine or linear λ-terms, we are interested in λ-terms with holes, which we call SwissCheeses.

A Swiss Cheese is a closed λ-term with holes at p levels.



The p levels of holes are $\Box_0, \ldots \Box_{p-1}$.

A hole $\Box_i$ is a location for a variable

- at level i,
- that is under i λ's.

An **m**-SwissCheese has

- m_0 holes at level 0,
- m_1 holes at level 1,
- ...
- m_{p-1} holes at level $p-1$.

$$\mathbf{m} = (m_0, \ldots, m_{p-1})$$

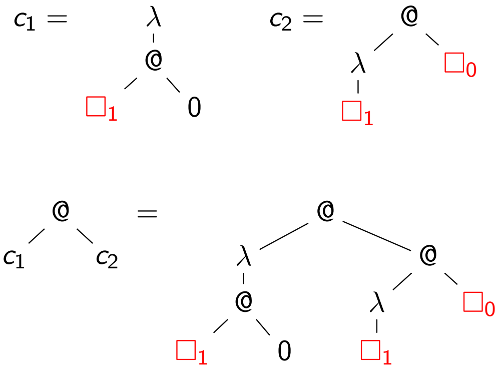
the characteristics

Building Swiss Cheeses

We assume that

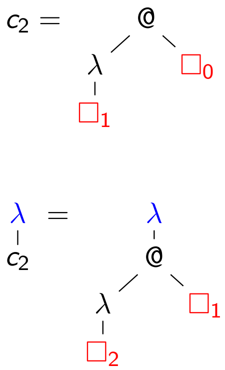
- either bound variables occur at most once (affine SwissCheeses),
- or bound variables occur once and only once (linear SwissCheese),

Building a SwissCheese by application



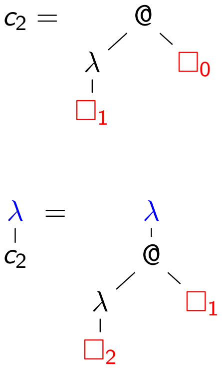
Building a SwissCheese by abstraction

Abstraction with no binding



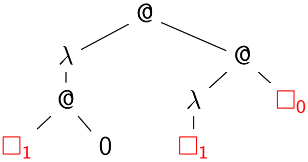
Building a SwissCheese by abstraction

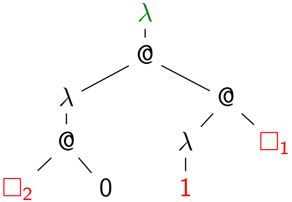
Abstraction with no binding



- λ is put on the top.
- Indices of boxes are incremented.

Abstraction with binding





- A box $\square_i$ is chosen.
- The box $\square_i$ is replaced by index i.
- Indices of boxes are incremented.

1 Affine, Linear, Closed

2 De Bruijn indices

3 Swiss Cheese

4 Counting closed linear terms

5 Counting closed affine terms

6 Generating functions

7 Effective computations

8 Generating terms

9 Other results

Counting closed linear terms

We consider natural size.
$l^{\nu}_{n,\mathbf{m}}$ is the number of linear Swiss Cheeses of size n with characteristics $\mathbf{m} = (m_0, ..., m_{p-1})$.

$n = 0$

There is only one Swiss Cheese of size 0 namely $\Box_0$.
This means that the number of SwissCheeses of size 0 is 1
if and only if
$\mathbf{m} = (1, 0, 0, ...)$:

$$l_{0,\mathbf{m}} \quad = \quad [m_0 = 1 \land \bigwedge_{j=1}^{p-1} m_j = 0]$$

Counting closed linear terms

$n+1$ and Application

$$\sum_{\mathbf{q} \oplus \mathbf{r} = \mathbf{m}} \sum_{k=0}^{n} l_{k,\mathbf{q}}\, l_{n-1-k,\mathbf{r}}$$

$\oplus$ is the componentwise addition of tuples.

$n+1$ and Abstraction with binding

$$\sum_{i=0}^{p-1} (m_i + 1)\; l^{\nu}_{n-i,\mathbf{m}^{\uparrow i}}$$

$\mathbf{m}^{\uparrow i} = (\ldots, m_i + 1, \ldots)$.

Counting closed linear terms

$$
\begin{aligned}
l^{\nu}_{n+1,0:\mathbf{m}} &= \sum_{\mathbf{q}\oplus\mathbf{r}\,=\,0:\mathbf{m}} \; \sum_{k=0}^{n} l^{\nu}_{k,\mathbf{q}}\, l^{\nu}_{n-k,\mathbf{r}} + \sum_{i=0}^{p-1} (m_i+1)\; l^{\nu}_{n-i,\mathbf{m}^{\uparrow i}} \\
l^{\nu}_{n+1,(h+1):\mathbf{m}} &= \sum_{\mathbf{q}\oplus\mathbf{r}=(h+1):\mathbf{m}} \; \sum_{k=0}^{n} l^{\nu}_{k,\mathbf{q}}\, l^{\nu}_{n-k,\mathbf{r}}
\end{aligned}
$$

1 Affine, Linear, Closed

2 De Bruijn indices

3 Swiss Cheese

4 Counting closed linear terms

5 **Counting closed affine terms**

6 Generating functions

7 Effective computations

8 Generating terms

9 Other results

Counting closed affine terms

Like closed linear terms except that Abstraction with no binding is added.

$$a^{\nu}_{n,\mathbf{m}}$$

Counting closed affine terms

Like closed linear terms except that Abstraction with no binding is added.

$$a^{\nu}_{n,\mathbf{m}}$$

$$a^{\nu}_{n+1,0:\mathbf{m}} = \sum_{\mathbf{q}\oplus\mathbf{r} = 0:\mathbf{m}} \sum_{k=0}^{n} a^{\nu}_{k,\mathbf{q}}\, a^{\nu}_{n-k,\mathbf{r}} + \sum_{i=0}^{p-1} (m_i + 1)\; a^{\nu}_{n-i,\mathbf{m}\uparrow i} + a^{\nu}_{n,\mathbf{m}}$$

$$a^{\nu}_{n+1,(h+1):\mathbf{m}} = \sum_{\mathbf{q}\oplus\mathbf{r}=(h+1):\mathbf{m}} \sum_{k=0}^{n} a^{\nu}_{k,\mathbf{q}}\, a^{\nu}_{n-k,\mathbf{r}}$$

Counting closed affine terms

Like closed linear terms except that Abstraction with no binding is added.

$$a^{\nu}_{n,\mathbf{m}}$$

$$a^{\nu}_{n+1,0:\mathbf{m}} = \sum_{\mathbf{q}\oplus\mathbf{r} = 0:\mathbf{m}} \sum_{k=0}^{n} a^{\nu}_{k,\mathbf{q}}\, a^{\nu}_{n-k,\mathbf{r}} + \sum_{i=0}^{p-1} (m_i + 1)\ a^{\nu}_{n-i,\mathbf{m}\uparrow^i} + a^{\nu}_{n,\mathbf{m}}$$

$$a^{\nu}_{n+1,(h+1):\mathbf{m}} = \sum_{\mathbf{q}\oplus\mathbf{r}=(h+1):\mathbf{m}} \sum_{k=0}^{n} a^{\nu}_{k,\mathbf{q}}\, a^{\nu}_{n-k,\mathbf{r}}$$

Number of closed affine terms up to 50

1	0	26	3513731861
2	1	27	9843728012
3	1	28	27633400879
4	2	29	77721141911
5	5	30	218984204904
6	12	31	618021576627
7	25	32	1746906189740
8	64	33	4945026080426
9	166	34	14017220713131
10	405	35	39784695610433
11	1050	36	113057573020242
12	2763	37	321649935953313
13	7239	38	916096006168770
14	19190	39	2611847503880831
15	51457	40	7453859187221508
16	138538	41	21292177500898858
17	374972	42	60875851617670699
18	1020943	43	174195916730975850
19	2792183	44	498863759031591507
20	7666358	45	1429753835635525063
21	21126905	46	4100730353324163138
22	58422650	47	11769771167532816128
23	162052566	48	33804054749367200891
24	450742451	49	97151933333668422006
25	1256974690	50	279385977720772581435

1 Affine, Linear, Closed

2 De Bruijn indices

3 Swiss Cheese

4 Counting closed linear terms

5 Counting closed affine terms

6 Generating functions

7 Effective computations

8 Generating terms

9 Other results

Generating functions for linear Swiss Cheeses

$\mathbf{m}$ is an infinite tuple.

$$L^{\nu}_{0:\mathbf{m}}(z) \;=\; z \sum_{\mathbf{m}'\oplus\mathbf{m}''=0:\mathbf{m}} L^{\nu}_{\mathbf{m}'}(z) L^{\nu}_{\mathbf{m}''}(z) + z \sum_{i=0}^{\infty} (m_i + 1) z^i L^{\nu}_{\mathbf{m}\uparrow i}(z)$$

$$L^{\nu}_{(h+1):\mathbf{m}}(z) \;=\; [h = 0 + \bigwedge_{i=0}^{\infty} m_i = 0] + z \sum_{\mathbf{m}'\oplus\mathbf{m}''=(h+1):\mathbf{m}} L^{\nu}_{\mathbf{m}'}(z) L^{\nu}_{\mathbf{m}''}(z)$$

Double series for linear Swiss Cheeses

$\mathbf{u}$ is $u_0, u_1, \ldots$ ad infinitum.

Double series (Bodini)

$$\mathcal{L}^{\nu}(z, \mathbf{u}) \;=\; u_0 + z(\mathcal{L}^{\nu}(z, \mathbf{u}))^2 + \sum_{i=1}^{\infty} z^i \frac{\partial \mathcal{L}^{\nu}(z, \mathrm{tail}\,(\mathbf{u}))}{\partial u^i}$$

$L^{\nu}_{0^\omega}(z) = \mathcal{L}^{\nu}(z, 0^{\omega})$ is the generating function for closed linear λ-terms.

Generating functions for affine Swiss Cheeses

$$A^{\nu}_{0:\mathbf{m}}(z) = z \sum_{\mathbf{m}' \oplus \mathbf{m}'' = 0:\mathbf{m}} A^{\nu}_{\mathbf{m}'}(z) A^{\nu}_{\mathbf{m}''}(z) + z \sum_{i=0}^{\infty} (m_i + 1) z^i A^{\nu}_{\mathbf{m}\uparrow i}(z) + z\, A^{\nu}_{\mathbf{m}}(z)$$

$$A^{\nu}_{(h+1):\mathbf{m}}(z) = [h = 0 + \bigwedge_{i=0}^{\infty} m_i = 0] + z \sum_{\mathbf{m}' \oplus \mathbf{m}'' = (h+1):\mathbf{m}} A^{\nu}_{\mathbf{m}'}(z) A^{\nu}_{\mathbf{m}''}(z)$$

Double series (Bodini)

$$\mathcal{A}^{\nu}(z, \mathbf{u}) = u_0 + z(\mathcal{A}^{\nu}(z, \mathbf{u}))^2 + \sum_{i=1}^{\infty} z^i \frac{\partial \mathcal{A}^{\nu}(z, \mathrm{tail}\,(\mathbf{u}))}{\partial u^i} + z\mathcal{A}^{\nu}(z, \mathrm{tail}\,(\mathbf{u}))$$

1 Affine, Linear, Closed

2 De Bruijn indices

3 Swiss Cheese

4 Counting closed linear terms

5 Counting closed affine terms

6 Generating functions

7 Effective computations

8 Generating terms

9 Other results

We compute `a n m`.
where `m` is a tuple of size `n`.

- If we look for `a 0 m` for a given `n`, we set length of `m` to `n`.
- Due to the high level of recursion one needs a memoization mechanism.
  One uses a memory and the call by need of Haskell.
- One counts all the value `a n m'` for `m'` componentwise less than `m`.

1 Affine, Linear, Closed

2 De Bruijn indices

3 Swiss Cheese

4 Counting closed linear terms

5 Counting closed affine terms

6 Generating functions

7 Effective computations

8 Generating terms

9 Other results

Generating terms

Slight changes on the programs.
Due to the many terms to be generated
the process is limited: 23 for linear terms (1420053 terms) and 19 for affine terms (2792183 terms).

Random terms

- A random closed affine term of size 19:

$$\lambda\lambda\lambda\lambda\lambda(\lambda((\lambda 0\ \lambda\lambda(1\ 0))\ 0)\ 0)$$

- A random closed linear term of size 23:

$$\lambda\lambda\lambda\lambda((\lambda(0\ \lambda 0)\ (3\ (0\ 2)))\ 1)$$

1 Affine, Linear, Closed

2 De Bruijn indices

3 Swiss Cheese

4 Counting closed linear terms

5 Counting closed affine terms

6 Generating functions

7 Effective computations

8 Generating terms

9 Other results

Counting terms with variable size 0

For variable size 0, one replaces

$$\sum_{i=0}^{p-1} (m_i + 1)\ l^{\nu}_{n-i,\mathbf{m}\uparrow i}$$

by

$$\sum_{i=0}^{p} (m_i + 1)\ l^{0}_{n,\mathbf{m}\uparrow i}$$

Counting terms with variable size 0

For variable size 0, one replaces

$$\sum_{i=0}^{p-1} (m_i + 1)\ l^{\nu}_{n-i,\mathbf{m}\uparrow i}$$

by

$$\sum_{i=0}^{p} (m_i + 1)\ l^{0}_{n,\mathbf{m}\uparrow i}$$

For variable size 1:

$$\sum_{i=0}^{p} (m_i + 1)\ l^{1}_{n-1,\mathbf{m}\uparrow i}$$

Counting β-normal forms

$$
\begin{aligned}
anf^{\nu}_{0,\mathbf{m}} &= ane^{\nu}_{0,\mathbf{m}} \\
anf^{\nu}_{n+1,\mathbf{m}} &= ane^{\nu}_{n+1,\mathbf{m}} + anf^{\nu}\lambda w_{n+1,m} + anf^{\nu}\lambda n_{n+1,m}
\end{aligned}
$$

Counting β-normal forms

$$
\begin{aligned}
anf^{\nu}_{0,\mathbf{m}} &= ane^{\nu}_{0,\mathbf{m}}\\
anf^{\nu}_{n+1,\mathbf{m}} &= ane^{\nu}_{n+1,\mathbf{m}} + anf^{\nu}\lambda w_{n+1,m} + anf^{\nu}\lambda n_{n+1,m}
\end{aligned}
$$

$$
\begin{aligned}
ane^{\nu}_{0,\mathbf{m}} &= m_0 = 1 \wedge \bigwedge_{j=1}^{p} m_j = 0\\
ane^{\nu}_{n+1,\mathbf{m}} &= \sum_{\mathbf{q}\oplus\mathbf{r} = 0:\mathbf{m}} \sum_{k=0}^{n} ane^{\nu}_{k,\mathbf{q}}\, anf^{\nu}_{n-k,\mathbf{r}}
\end{aligned}
$$

Counting β-normal forms

$$\begin{aligned} anf^{\nu}_{0,\mathbf{m}} &= ane^{\nu}_{0,\mathbf{m}} \\ anf^{\nu}_{n+1,\mathbf{m}} &= ane^{\nu}_{n+1,\mathbf{m}} + anf^{\nu}\lambda w_{n+1,m} + anf^{\nu}\lambda n_{n+1,m} \end{aligned}$$

$$\begin{aligned} ane^{\nu}_{0,\mathbf{m}} &= m_0 = 1 \wedge \bigwedge_{j=1}^{p} m_j = 0 \\ ane^{\nu}_{n+1,\mathbf{m}} &= \sum_{\mathbf{q}\oplus\mathbf{r} = 0:\mathbf{m}} \sum_{k=0}^{n} ane^{\nu}_{k,\mathbf{q}}\, anf^{\nu}_{n-k,\mathbf{r}} \end{aligned}$$

$$anf^{\nu}\lambda w_{n+1,m} = \sum_{i=0}^{n} (m_i + 1)\; anf^{\nu}_{n-i,\mathbf{m}\uparrow i}$$

$$anf^{\nu}\lambda n_{n+1,m} = anf^{\nu}_{n,m}$$

Further works

- (Dan Dougherty) A generic framework for
  - plain terms and
  - linear and affine terms.
- To generate random terms further than 19 and 23.

Questions ?

Generating functions of m-open λ-terms with natural size.

$$P_m(z) = \frac{z(1-z^m)}{1-z} + zP_m^2(z) + zP_{m+1}(z)$$

m replaces the characteristics.