Paper Reading:

“Efficient Divide-and-Conquer Parsing of Practical Context-Free Languages”

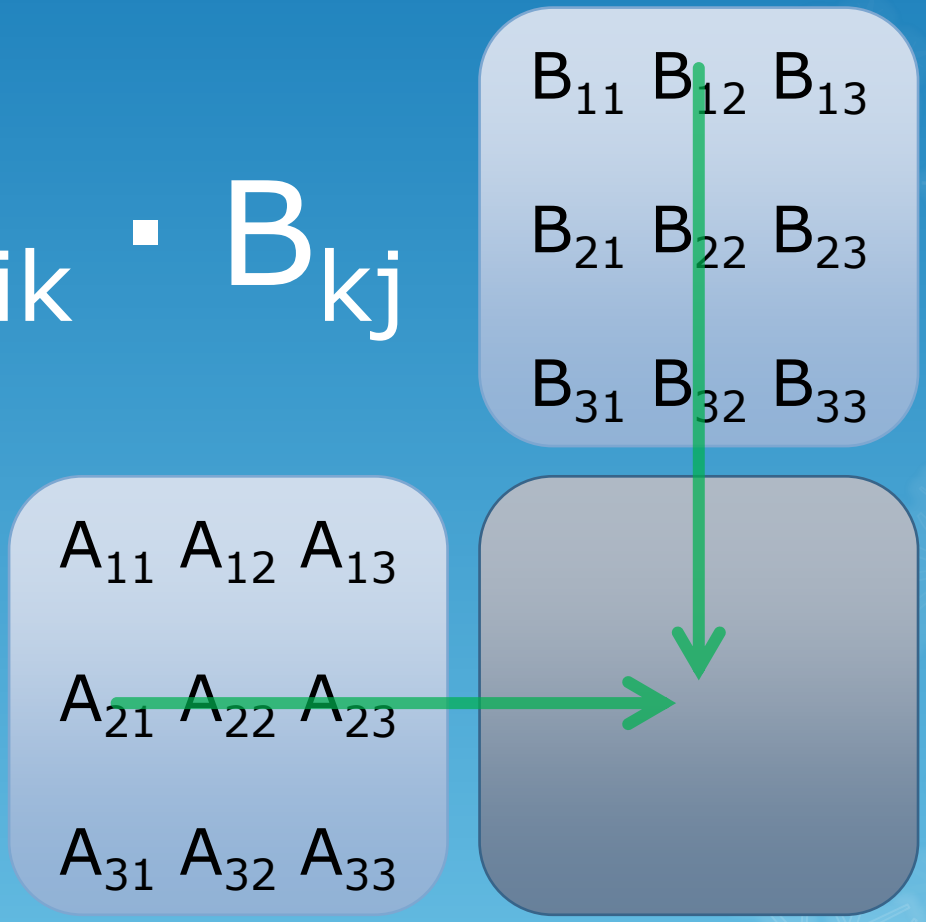
Paper from ICFP’13

発表者： kinaba

前提知識１：行列の掛け算

N×N 行列の掛け算

$$C_{ij} = \Sigma A_{ik} \cdot B_{kj}$$



$O(N^3)$

回の加算乗算

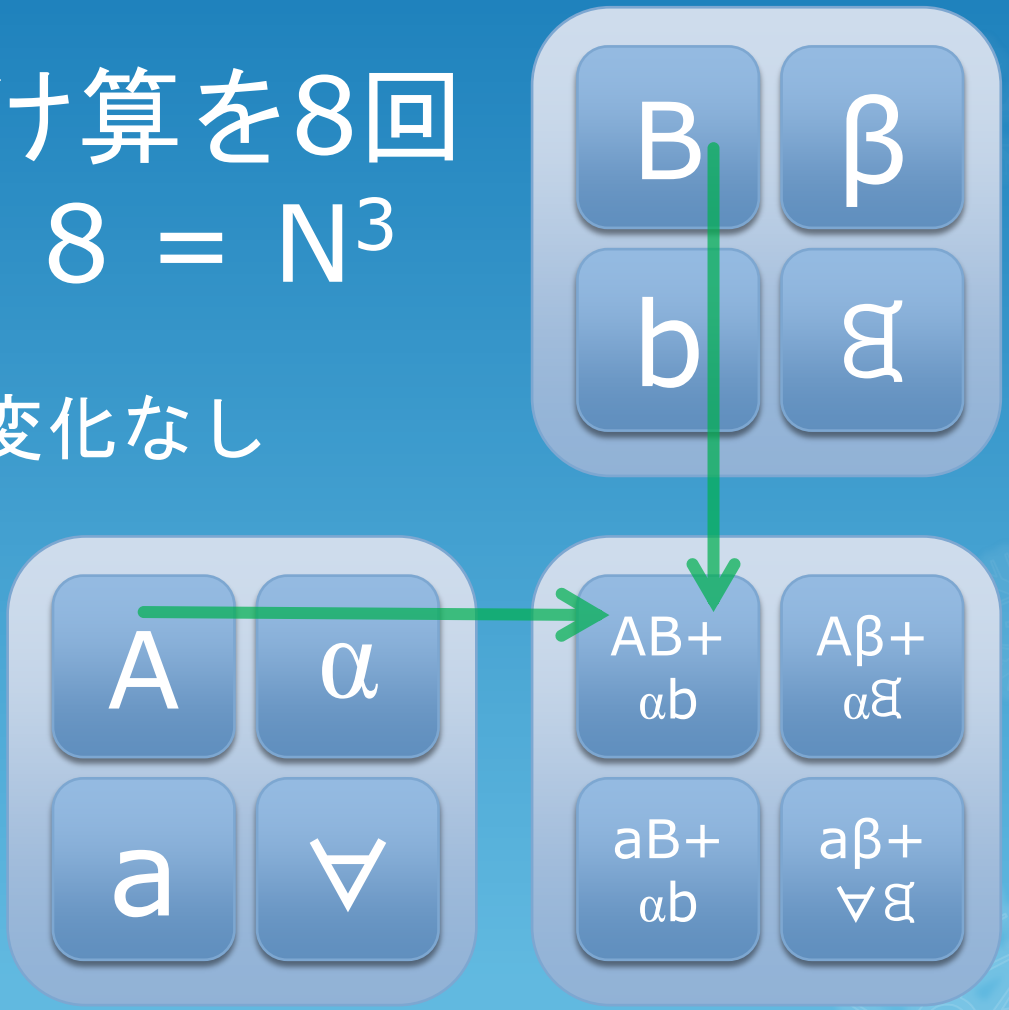
再帰的に掛け算

N/2 × N/2 の
行列 4 つに分解して
行列掛け算、
と考えてもよい

再帰的に掛け算

N/2 の掛け算を8回
$(N/2)^3 \times 8 = N^3$

このままでは変化なし



[Strassen 69]

式変形すると
7回でできる!!!
⇒ wikipedia
⇒ $O(N^{\log_2(7)})$ 乗算

[Williams 11] 等々

もっと頑張って式変形と解析すると、もっと減る!!!!!!!

Lemma 17.

$$V_{2410} \geq 2\left(\frac{(V_{026}V_{224}V_{035})^{3/2}}{V_{125}^{3/2}} + \frac{V_{116}^{3/2}V_{134}^{3/2}}{2}\right)^{1/3}\left(\frac{V_{044}^{3}V_{026}^{3/2}V_{125}^{3/2}}{(V_{224}^{3/2}V_{035}^{3/2})} + (V_{116}V_{134})^{3/2} + \frac{(V_{026}V_{224}V_{125})^{3/2}}{(2V_{035}^{3/2})}\right)^{1/3} \times$$

$$\left(\frac{V_{224}^{3}}{V_{044}^{3}V_{026}^{3}} + \frac{(V_{017}^{2}V_{233}^{2}V_{125})^{3/2}}{(V_{026}V_{224}V_{035}V_{116}V_{134})^{3/2}} + 1 + \frac{(V_{035}V_{125}^{3})^{3/2}}{2(V_{026}V_{224}V_{116}V_{134})^{3/2}}\right)^{1/3}\left(\frac{V_{224}V_{035}V_{134}}{(V_{233}V_{044}V_{125})}\right)^{f}.$$

$w = .008959174543374085484041135837907754 8, x = .419906566426457247737023400662697 04 \cdot 10^{-3},$

Using Maple, we obtain the bound

$$\omega \leq 2.372658.$$

ω

現時点の
人類の知識:

$\omega \leqq 2.372658$

行列乗算は

$O(N^{\omega})$

でできる

漸近計算量は速いが
あまりに複雑なので
実際に動かすと
$O(N^3)$ の普通の乗算を
きちんと書いた方が速い。

使えるのはせいぜい Strassen まで。
ωを減らすのは理論的興味。
…と考えられている模様

前提知識２：文脈自由文法

0 と 1 からなる奇数長の回文の文法

$S \rightarrow 0\ S\ 0$
$S \rightarrow 1\ S\ 1$
$S \rightarrow 0$
$S \rightarrow 1$

C言語風の何かの文法

文 → if (*式*) *ブロック*
文 → if (*式*) *ブロック* else *ブロック*
文 → *式* ;
ブロック → *文*
ブロック → { *文列*
文列 → *文 文列*
文列 → }

文脈自由文法

非終端記号 を、文法にしたがって
書き換えっていって作れるのが
その”言語”の”文”

という”文法”の定義

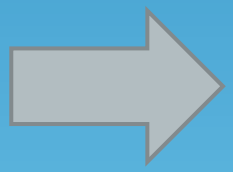
$$S \rightarrow 0\,S\,0$$
$$S \rightarrow 1\,S\,1$$
$$S \rightarrow 0$$
$$S \rightarrow 1$$

Chomsky 標準形 (CNF)

書き換え先の右辺が記号２つか１つな文法のことを “Chomsky標準形” であるという。# やや不正確な定義

この発表ではCNFだけ
考えます

$S \to 0\ S\ 0$
$S \to 1\ S\ 1$
$S \to 0$
$S \to 1$



$S \to 0\ T$
$T \to S\ 0$
$S \to 1\ U$
$U \to S\ 1$
$S \to 0$
$S \to 1$

CYK法
[Cocke et al. 70][Younger 67][Kasami 65]

文脈自由な文法に対して

$S \rightarrow 0\ T$
$T \rightarrow S\ 0$
$S \rightarrow 1\ U$
$U \rightarrow S\ 1$
$S \rightarrow 0$
$S \rightarrow 1$

"10101"

長さ N の文字列

の構文解析は $O(N^3)$ 時間でできる

CYK法

[Cocke et al. 70][Younger 67][Kasami 65]

"10101"

$S \rightarrow 0\ T$
$T \rightarrow S\ 0$
$S \rightarrow 1\ U$
$U \rightarrow S\ 1$
$S \rightarrow 0$
$S \rightarrow 1$

1				
-	**0**			
-	-	**1**		
-	-	-	**0**	
-	-	-	-	**1**

CYK法

[Cocke et al. 70][Younger 67][Kasami 65]

"10101"

$S \rightarrow 0\ T$
$T \rightarrow S\ 0$
$S \rightarrow 1\ U$
$U \rightarrow S\ 1$
$S \rightarrow 0$
$S \rightarrow 1$

1,S				
-	0,S			
-	-	1,S		
-	-	-	0,S	
-	-	-	-	1,S

CYK法
[Cocke et al. 70][Younger 67][Kasami 65]

"10101"

$S \rightarrow 0\ T$
$T \rightarrow S\ 0$
$S \rightarrow 1\ U$
$U \rightarrow S\ 1$
$S \rightarrow 0$
$S \rightarrow 1$

1,S	T			
-	0,S			
-	-	1,S		
-	-	-	0,S	
-	-	-	-	1,S

$M_{12} = M_{11} \star M_{22}$

"1文字目から2文字目まで"
を生成するには
"1文字目から1文字目" を作り
"2文字目から2文字目"を作る

CYK法

[Cocke et al. 70][Younger 67][Kasami 65]

"10101"

$M_{13} = M_{11} \star M_{23}$
$\dagger M_{12} \star M_{33}$

S → 0 **T**
T → **S** 0
S → 1 **U**
U → **S** 1
S → 0
S → 1

1,S	T	S		
-	0,S	U		
-	-	1,S	T	
-	-	-	0,S	U
-	-	-	-	1,S

CYK法

[Cocke et al. 70][Younger 67][Kasami 65]

"10101"

$S \rightarrow 0\ T$
$T \rightarrow S\ 0$
$S \rightarrow 1\ U$
$U \rightarrow S\ 1$
$S \rightarrow 0$
$S \rightarrow 1$

1,S	T	S	T	S
-	0,S	U	S	U
-	-	1,S	T	S
-	-	-	0,S	U
-	-	-	-	1,S

本題

[Valiant 75]

General Context-Free Recognition in Less than Cubic Time

LESLIE G. VALIANT*

定理:
CFG構文解析は $O(N^{\omega})$ 時間で可能

行列の掛け算と
同じくらいの

おさらい

行列の掛け算と
同じくらいの

CFG構文解析は $O(N^{\omega})$ 時間で可能

$O(N^3)$ 未満の行列乗算法は、複雑で
実用には適さない

おさらい

CFG構文解析は 行列の掛け算と同じくらいの $O(N^{\omega})$ 時間で可能

$O(N^3)$ 未満の行列乗算法は、複雑で実用には適さない

⇒

行列乗算を使うValiantの構文解析法も実用するなら$O(N^3)$でCYKと同じ。
この定理は理論的な興味にすぎない…？

読んだ論文

Efficient Divide-and-Conquer Parsing of Practical Context-Free Languages

Long Version

Jean-Philippe Bernardy Koen Claessen

We present a divide-and-conquer algorithm for parsing context-free languages efficiently. Our algorithm is an instance of Valiant's (1975), who reduced the problem of parsing to matrix multiplica-

読んだ論文

- “Efficient Divide-and-Conquer Parsing of Practical Context-Free Languages”
- To appear in ICFP’13
- Bernardy & Claessen
  - Haskell で書かれ Haskell でスクリプトが書けるエディタ “Yi” を作っている人

- Valiant 法を新しい視点から見直す
- Valiant 法が “速く” 動く文法について考える
  - 実際のプログラミング言語の文法はそうであると主張する
- あと普通に Valiant 法の説明が Valiant の原論文より100倍読みやすい

Valiant 法を新しい視点から見直す

“文脈自由文法の構文解析は
行列乗算を使って実装できる”

- △ 理論上 $O(N^{\omega})$ 時間でできる
- ○ 乗算が高速にできるような
  特殊な行列しか使わないなら
  実用上もっと速くできる

疎行列！

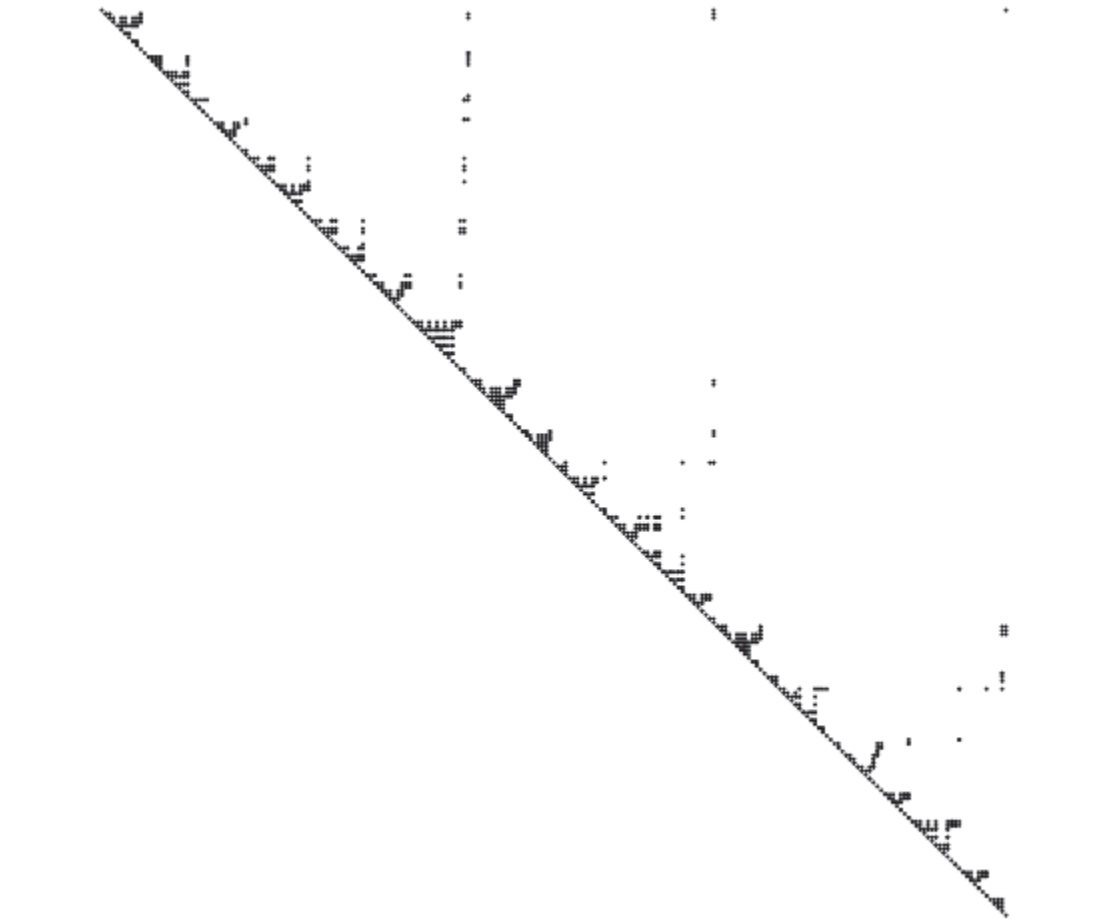


Figure 4. The chart corresponding to a fragment of a C program.

Valiant 法の説明 (Why行列)

CYK 法を思い出そう

$$M_{ij} = \bigcup (M_{ik} \star M_{kj})$$

行列の
掛け算の形

“i 文字目から j文字目まで”
が生成可能か調べる
“i 文字目から k 文字目” を調べ
“k 文字目から j 文字目”を調べ
それを作れる文法規則があるか
調べる

1,S	T	S		
-	0,S	U		
-	-	1,S	T	
-	-	-	0,S	U
-	-	-	-	1,S

Valiant 法の説明(Why行列)

CYK 法のナナメ一列を埋める操作は

- 要素　⇒　非終端記号の集合
- 要素の足し算（ ∪ ）　⇒　集合の合併
- 要素の掛け算（ ☆ ）⇒
  $X☆Y = \{N \mid N \rightarrow ML, M \in X, L \in Y\}$

の行列乗算！！！！！！

1,S	T	S		
-	0,S	U		
-	-	1,S	T	
-	-	-	0,S	U
-	-	-	-	1,S

Valiant 法の説明

- CYK 法のナナメ一列を埋める操作は行列乗算

  ⇒ ナナメ N 列を埋める操作は $O(N^{\omega+1})$!!?!?!!!?!?!?!

- CYK 法の表全部を埋める操作は、
  乗算結果が変わらなくなるまで掛け算を繰り返す操作

  ⇒ “Transitive Closure”
  ⇒ **定理： 上三角行列のTCは$O(N^{\omega})$時間で計算できる**

1,S	T	S		
-	0,S	U		
		1,S	T	
		-	0,S	U
		-	-	1,S

Definition 8 (Transitive closure). *If it exists, the transitive closure of a matrix W, written W^+, is the matrix C such that*

$$C = C \cdot C + W$$

Valiant 法の説明 (TC計算)

Definition 8 (Transitive closure). *If it exists, the transitive closure of a matrix W, written W^+, is the matrix C such that*

$$C = C \cdot C + W$$

“C=C・C+W”
⇒
掛け算しても
変わらない
不動点

Valiant 法の説明

Strassen乗算を思いだそう

N/2 × N/2 行列に分解して
再帰的に計算

Valiant 法の説明

Definition 8 (Transitive closure). *If it exists, the transitive closure of a matrix W, written W^+, is the matrix C such that*

$$C = C \cdot C + W$$

TCも N/2 × N/2 行列に分解して再帰計算

$$W = \begin{bmatrix} A & X \\ 0 & B \end{bmatrix} \qquad C = \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix}$$

Then the condition that C is the transitive closure of W becomes

$$\begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} = \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} \cdot \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} + \begin{bmatrix} A & X \\ 0 & B \end{bmatrix}$$

Valiant 法の説明

$$W = \begin{bmatrix} A & X \\ 0 & B \end{bmatrix} \qquad C = \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix}$$

Then the condition that C is the transitive closure of W becomes

$$\begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} = \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} \cdot \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} + \begin{bmatrix} A & X \\ 0 & B \end{bmatrix}$$

$$A' = A'A' + A$$
$$X' = A'X' + X'B' + X$$
$$B' = B'B' + B$$

A′ と B′ は
A と B の TC。
再帰計算で求まる。

X′ は？

Valiant 法の説明

$$W = \begin{bmatrix} A & X \\ 0 & B \end{bmatrix} \qquad C = \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix}$$

Then the condition that C is the transitive closure of W becomes

$$\begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} = \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} \cdot \begin{bmatrix} A' & X' \\ 0 & B' \end{bmatrix} + \begin{bmatrix} A & X \\ 0 & B \end{bmatrix}$$

$$A' = A'A' + A$$
$$X' = A'X' + X'B' + X$$
$$B' = B'B' + B$$

A,B,Xが与えられた時

$$Y = AY + YB + X$$

を満たすYを求める関数を

$$V(A, X, B)$$

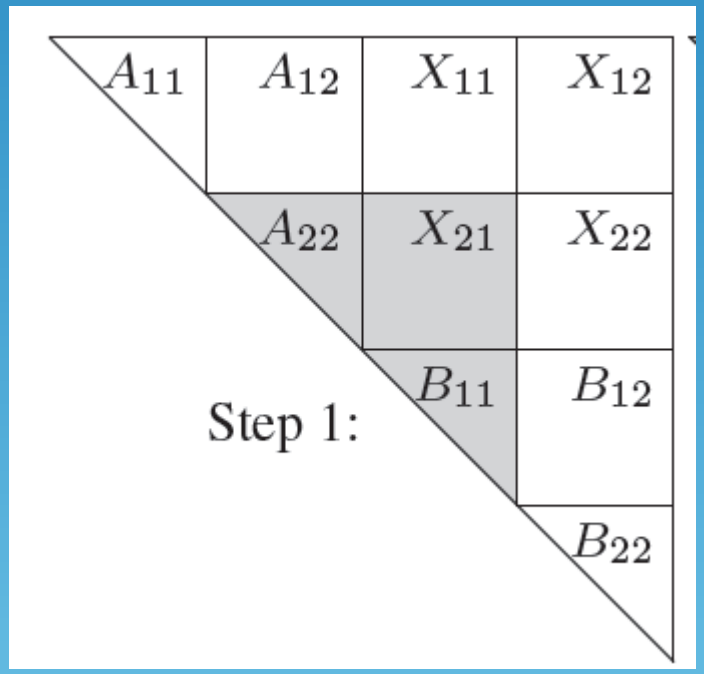
とする。X′=V(A′,X,B′)

A′ と B′ は
A と B の TC。
再帰計算で求まる。

X′ は？

Valiant 法の説明

$$V\left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}\right) = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$



さらに
４つに
分解

$$V\left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}\right) = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \cdot \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} + \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix} + \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$$

$$\begin{aligned}
Y_{11} &= A_{11}Y_{11} + A_{12}Y_{21} + Y_{11}B_{11} + 0 + X_{11} \\
Y_{12} &= A_{11}Y_{12} + A_{12}Y_{22} + Y_{11}B_{12} + Y_{12}B_{22} + X_{12} \\
Y_{21} &= 0 + A_{22}Y_{21} + Y_{21}B_{11} + 0 + X_{21} \\
Y_{22} &= 0 + A_{22}Y_{22} + Y_{21}B_{12} + Y_{22}B_{22} + X_{22}
\end{aligned}$$

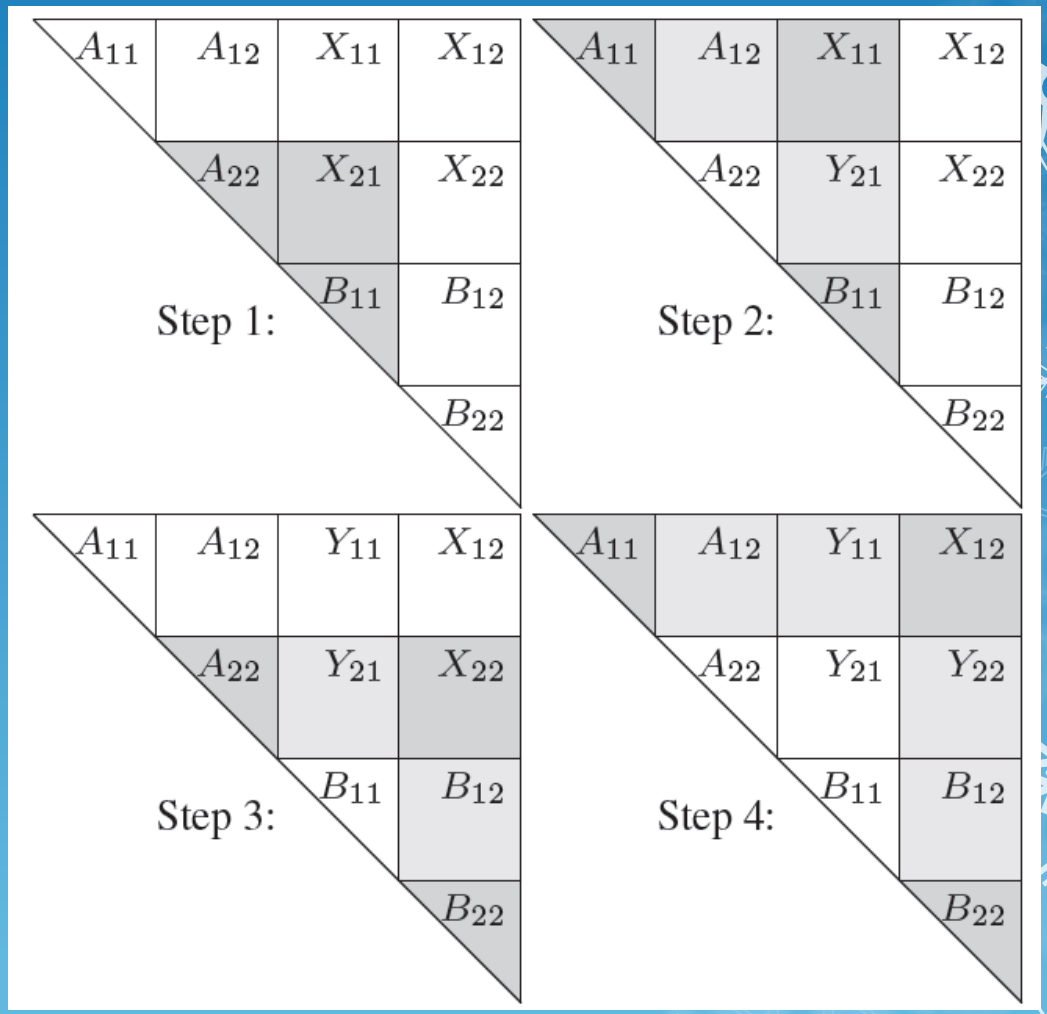
$Y = AY + YB + X$ を満たす Y = $V(A, X, B)$

$$
\begin{aligned}
Y_{11} &= A_{11}Y_{11} + A_{12}Y_{21} + Y_{11}B_{11} + 0 + X_{11} \\
Y_{12} &= A_{11}Y_{12} + A_{12}Y_{22} + Y_{11}B_{12} + Y_{12}B_{22} + X_{12} \\
Y_{21} &= 0 + A_{22}Y_{21} + Y_{21}B_{11} + 0 + X_{21} \\
Y_{22} &= 0 + A_{22}Y_{22} + Y_{21}B_{12} + Y_{22}B_{22} + X_{22}
\end{aligned}
$$

$$
\begin{aligned}
Y_{21} &= V(A_{22}, X_{21}, B_{11}) \\
Y_{11} &= V(A_{11}, X_{11} + A_{12}Y_{21}, B_{11}) \\
Y_{22} &= V(A_{22}, X_{22} + Y_{21}B_{12}, B_{22}) \\
Y_{12} &= V(A_{11}, X_{12} + A_{12}Y_{22} + Y_{11}B_{12}, B_{22})
\end{aligned}
$$

$$
\begin{aligned}
Y_{21} &= V(A_{22}, X_{21} &&, B_{11}) \\
Y_{11} &= V(A_{11}, X_{11} + A_{12}Y_{21} &&, B_{11}) \\
Y_{22} &= V(A_{22}, X_{22} + Y_{21}B_{12} &&, B_{22}) \\
Y_{12} &= V(A_{11}, X_{12} + A_{12}Y_{22} + Y_{11}B_{12}, B_{22})
\end{aligned}
$$

この式変形の
読み方



Valiant 法の計算時間

$$
\begin{aligned}
A' &= A'A' + A \\
X' &= A'X' + X'B' + X \\
B' &= B'B' + B
\end{aligned}
$$

$$
\begin{aligned}
Y_{21} &= V(A_{22}, X_{21} &&, B_{11}) \\
Y_{11} &= V(A_{11}, X_{11} + A_{12}Y_{21} &&, B_{11}) \\
Y_{22} &= V(A_{22}, X_{22} + Y_{21}B_{12} &&, B_{22}) \\
Y_{12} &= V(A_{11}, X_{12} + A_{12}Y_{22} + Y_{11}B_{12}, &&B_{22})
\end{aligned}
$$

T(N) = N×N行列のTC計算
V(N) = N×N行列の V の計算
M(N) = N×N行列の乗算

$$T(N) = 2\,T(N/2) + V(N)$$
$$V(N) = 4\,V(N/2) + 4\,M(N/4)$$

よって $T(N) \leqq O(N^{\omega})$ (???)

Haskell での実装

```
import Prelude (Eq (..))
class RingLike a where
  zero :: a
  (+) :: a → a → a
  (·) :: a → a → a

data M a = Q (M a) (M a) (M a) (M a) | Z | One a

q Z Z Z Z = Z
q a b c d = Q a b c d

one x = if x ≡ zero then Z else One x

instance (Eq a, RingLike a) ⇒ RingLike (M a) where
  zero = Z

  Z + x = x
  x + Z = x
  One x + One y = one (x + y)
  Q a11 a12 a21 a22 + Q b11 b12 b21 b22
    = q (a11 + b11) (a12 + b12)
        (a21 + b21) (a22 + b22)

  Z · x = Z
  x · Z = Z
  One x · One y = one (x · y)
  Q a11 a12 a21 a22 · Q b11 b12 b21 b22
    = q (a11 · b11 + a12 · b21) (a11 · b12 + a12 · b22)
        (a21 · b11 + a22 · b21) (a21 · b21 + a22 · b22)

v :: (Eq a, RingLike a) ⇒ M a → M a → M a → M a
v a                      Z                      b = Z
v Z                      (One x)                Z = One x
v (Q a11 a12 Z a22)      (Q x11 x12 x21 x22)    (Q b11 b12 Z b22)
  = q y11 y12 y21 y22
    where y21 = v a22  x21                          b11
          y11 = v a11 (x11 + a12 · y21            ) b11
          y22 = v a22 (x22 +           y21 · b12) b22
          y12 = v a11 (x12 + a12 · y22 + y11 · b12) b22
```

どのような文法で、疎行列になるか？

Definition 9 (Assumption). *There exists a constant α such that, for any input, the distribution of non-zero elements in the chart C corresponding to it is bounded as follows. For any square subchart A of C above the diagonal,*

$$\#A \leq \left\lceil \alpha \sum_{(i,j) \in \mathrm{dom}(A)} \frac{1}{(j-i)^2} \right\rceil$$

where $\#A$ is the number of non-zero elements in matrix A.

スライドはここで終わっている

（スライド作っている時間がありませんでした...）