STR セミナー 2026

部分文字列方程式と双方向スキームの相互変換

柴田 紘希（九州大学）

坂内 英夫（東京科学大学）

発表スライド

研究の背景（概要）

- **LZ 分解** [Ziv and Lempel, 1977]: `zip` にも使われている古典的・実用上高性能な圧縮手法
- **双方向スキーム (BMS)** [Storer and Szymanski, 1982]: LZ 分解の一般化で、理論的には LZ 分解よりも真に強力
- **部分文字列方程式 (SES)** [Shibata and Bannai, 2026]: 新たな圧縮手法で、双方向スキームと比べて**少なくとも同程度以上に強力**

研究の背景（概要）

- **LZ 分解** [Ziv and Lempel, 1977]: `zip` にも使われている古典的・実用上高性能な圧縮手法
- **双方向スキーム (BMS)** [Storer and Szymanski, 1982]: LZ 分解の一般化で、理論的には LZ 分解よりも真に強力
- **部分文字列方程式 (SES)** [Shibata and Bannai, 2026]: 新たな圧縮手法で、双方向スキームと比べて**少なくとも同程度以上に強力**

研究の背景（概要）

- **LZ 分解** [Ziv and Lempel, 1977]: `zip` にも使われている古典的・実用上高性能な圧縮手法
- **双方向スキーム (BMS)** [Storer and Szymanski, 1982]: LZ 分解の一般化で、理論的には LZ 分解よりも真に強力
- **部分文字列方程式 (SES)** [Shibata and Bannai, 2026]: 新たな圧縮手法で、双方向スキームと比べて**少なくとも同程度以上に強力**

研究の背景（概要）

- **LZ 分解** [Ziv and Lempel, 1977]: `zip` にも使われている古典的・実用上高性能な圧縮手法
- **双方向スキーム (BMS)** [Storer and Szymanski, 1982]: LZ 分解の一般化で、理論的には LZ 分解よりも真に強力
- **部分文字列方程式 (SES)** [Shibata and Bannai, 2026]: 新たな圧縮手法で、双方向スキームと比べて**少なくとも同程度以上に強力**

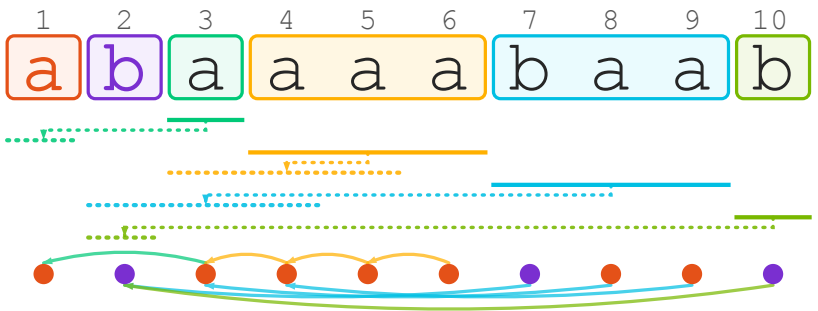
research question

双方向スキームと部分文字列方程式の理論的な圧縮能力は**同等**なのか？ それとも**部分文字列方程式の方が真に強力**なのか？

LZ 分解

LZ 分解 [Ziv and Lempel, 1977]: 文字列を以下のフレーズ列に分解して表す圧縮手法

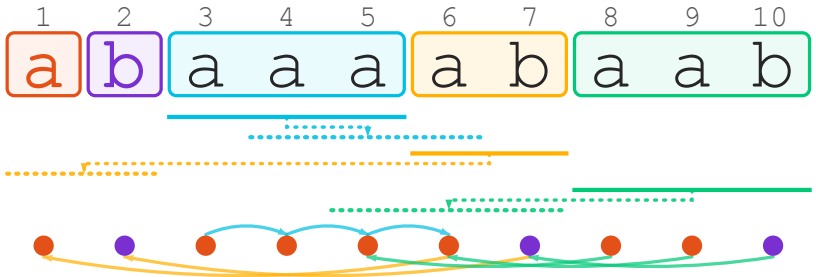
1. **char phrase:** 単一の文字を表す
2. **copy phrase:** より左側に現れる同じ部分文字列の参照を表す



双方向スキーム (BMS)

双方向スキーム (BMS) [Storer and Szymanski, 1982]: copy phrase が**自身より右側の文字列も参照できるような LZ 分解の一般化**

- ループができてしまうような参照関係は許さない

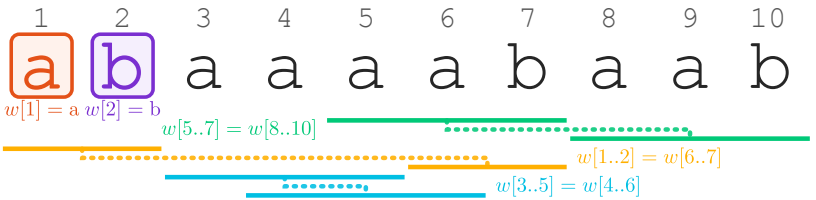


部分文字列方程式 (SES)

部分文字列方程式 (SES) [Shibata and Bannai, 2026]: 文字列を以下の形の制約の集合によって記述する表現手法

1. Equality 制約 (i, j, ℓ): $w[i..i+\ell-1] = w[j..j+\ell-1]$
2. Assignment 制約 (k, c): $w[k] = c$

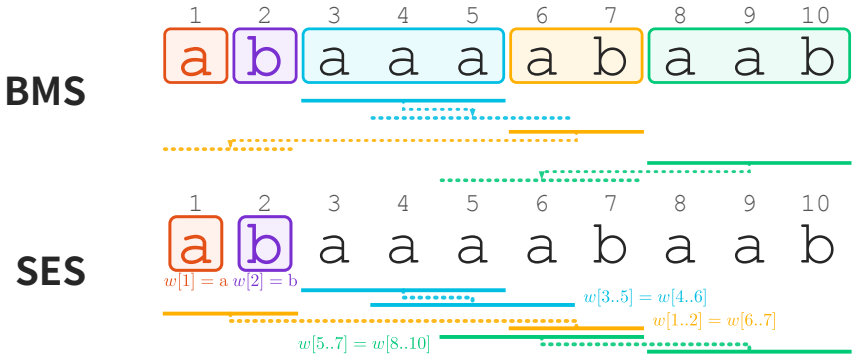
w **を表す SES**: 全制約を満たす文字列が w のみである SES



BMS との関係

観察 1: BMS と SES の関係

w に対する phrase 数 t の BMS が与えられたとき、
w を表す制約数 t の SES を構成できる。



BMS/SESに関する疑問

事実: SESの最小サイズ $s(w)$ は、常にBMSの最小サイズ $b(w)$ 以下

直感: SESはBMSより**シンプルな構造**であるため、もっと強いことが言えるかも？

- フレーズの重なりを許す
- 参照方向の向きづけが存在しない

BMS/SESに関する疑問

事実: SESの最小サイズ $s(w)$ は、常にBMSの最小サイズ $b(w)$ 以下

直感: SESはBMSより**シンプルな構造**であるため、もっと強いことが言えるかも？

- フレーズの重なりを許す
- 参照方向の向きづけが存在しない

BMS/SES に関する疑問

事実: SES の最小サイズ $s(w)$ は、常に BMS の最小サイズ $b(w)$ 以下

直感: SES は BMS より**シンプルな構造**であるため、もっと強いことが言えるかも？

- フレーズの重なりを許す
- 参照方向の向きづけが存在しない

research question

BMS/SES の**サイズ関係**はどうなっているのか？

今回の問題と結果

今回扱う問題

最小 BMS サイズと最小 SES サイズの関係は以下のどちらか？

- どんな文字列に対しても $s(w) \leq b(w) \leq \alpha s(w)$ が成り立ち、高々定数倍の差しかない
- 実はオーダーレベルの差があり、$s(w) \in o(b(w))$ となるような文字列（の系列）が存在する

$b(w)$: 文字列 w の最小 BMS サイズ, $s(w)$: 文字列 w の最小 BMS サイズ, $\alpha \geq 1$ は任意の定数

今回の問題と結果

今回扱う問題

最小 BMS サイズと最小 SES サイズの関係は以下のどちらか？

- どんな文字列に対しても $s(w) \leq b(w) \leq \alpha s(w)$ が成り立ち、高々定数倍の差しかない
- 実はオーダーレベルの差があり、$s(w) \in o(b(w))$ となるような文字列（の系列）が存在する

$b(w)$: 文字列 w の最小 BMS サイズ, $s(w)$: 文字列 w の最小 BMS サイズ, $\alpha \geq 1$ は任意の定数

研究の成果

任意の文字列 w に対して、$s(w) \leq b(w) \leq 4s(w)$ が成り立つ。

証明手法のアウトライン

研究の成果

任意の文字列 w に対して、$s(w) \leq b(w) \leq 4s(w)$ が成り立つ。

$s(w) \leq b(w)$: 前述の議論より成り立つ
$b(w) \leq 4s(w)$: **SES → BMS の変換手法を与える**ことで示す

変換手法の流れ:

1. SES を forest SES に変換する
2. forest SES から、一致関係を表現する置換 π を作る
3. 置換 π から BMS を構成する

証明手法のアウトライン

研究の成果

任意の文字列 w に対して、$s(w) \leq b(w) \leq 4s(w)$ が成り立つ。

$s(w) \leq b(w)$: 前述の議論より成り立つ
$b(w) \leq 4s(w)$: **SES → BMS の変換手法を与える**ことで示す

変換手法の流れ:

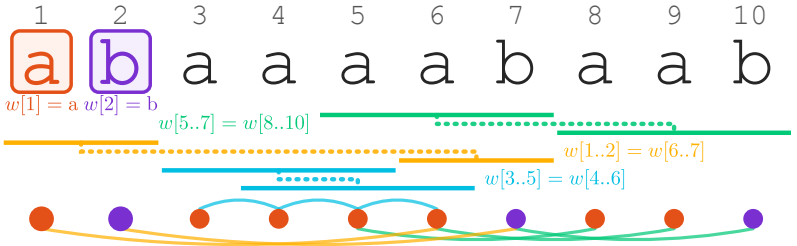
1. **SES を forest SES に変換する**
2. forest SES から、一致関係を表現する置換 π を作る
3. 置換 π から BMS を構成する

SES graph

定義 1: SES graph

文字列 w を表す SES $\mathcal{S}$ において、頂点集合が $\{1, \ldots, n\}$ であり、各 Equality 制約 (i, j, ℓ) について、$i + k$ と $j + k$ $(0 \leq k < \ell)$ の間に無向辺が存在するようなグラフを、$\mathcal{S}$ の **SES graph** と呼ぶ。

観察: SES graph で同じ連結成分に属する要素は同じ文字を表す

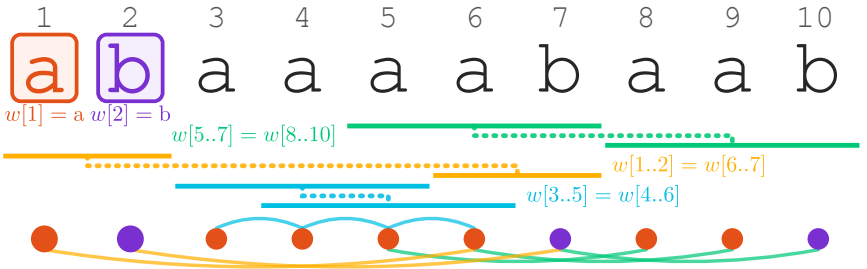


forest SES

定義 2: forest SES

SES graph が森をなすような SES を **forest SES** と呼ぶ。

直感:「文字列復元のために余分な一致関係がない」SES

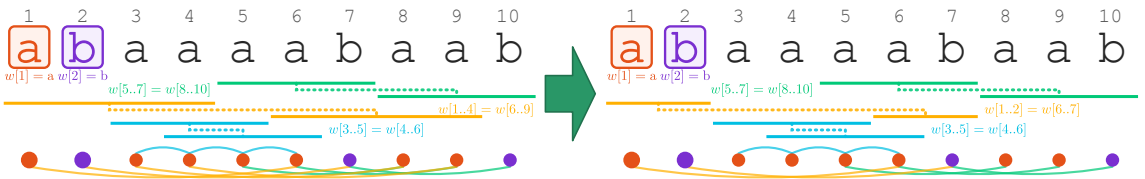


SES の forest SES 化

補題 1: SES の forest SES 化

任意の SES は、同じ文字列を表す forest SES に制約数を増やさずに変換できる。

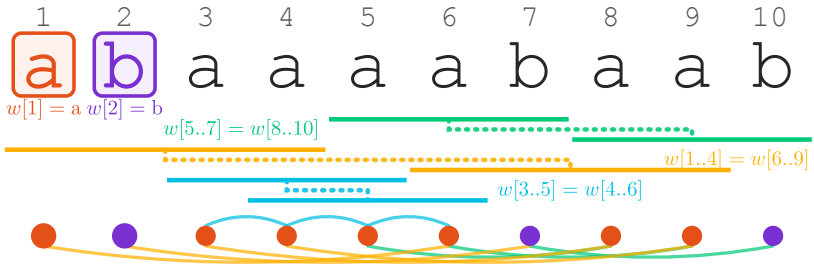
SES graph の**閉路削減アルゴリズム**を繰り返し用いて変換する



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

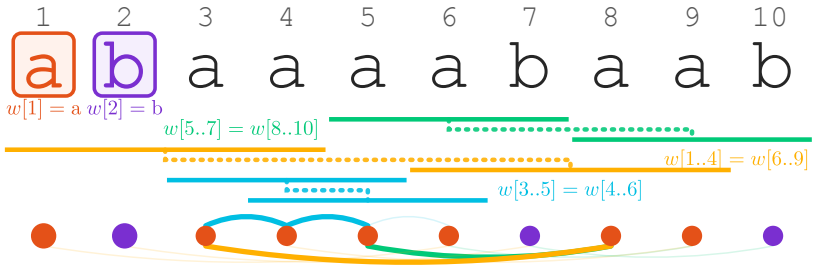
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

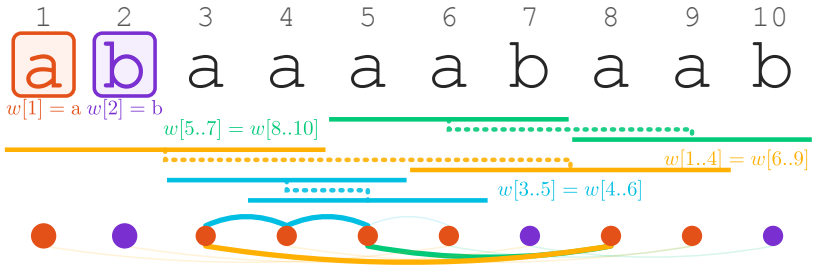
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

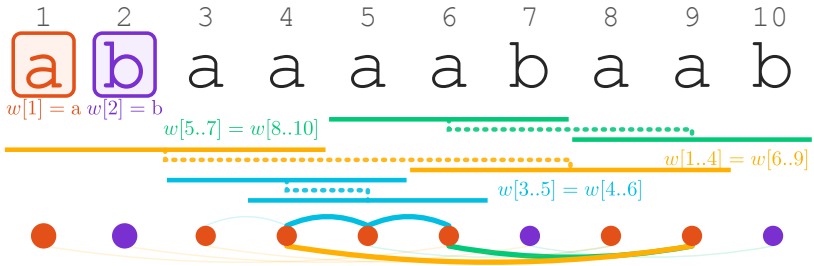
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

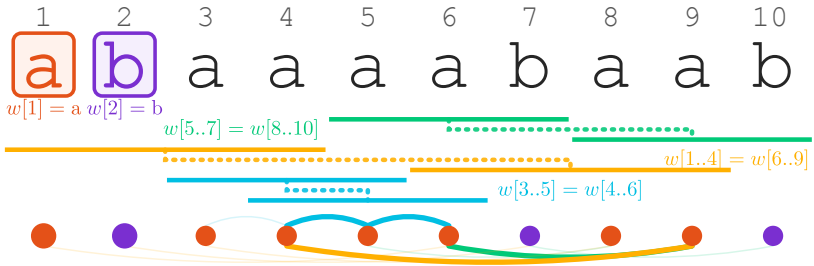
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

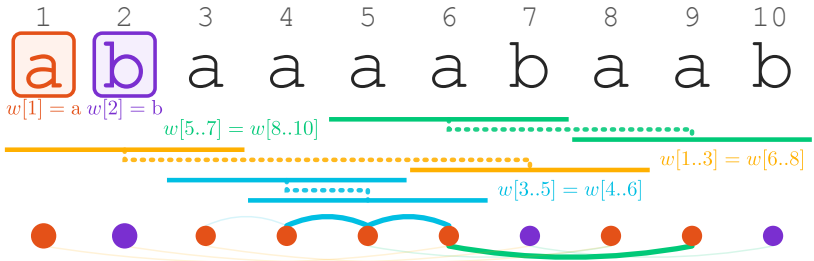
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

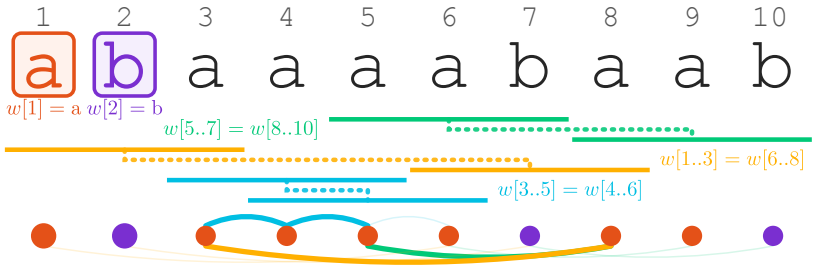
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

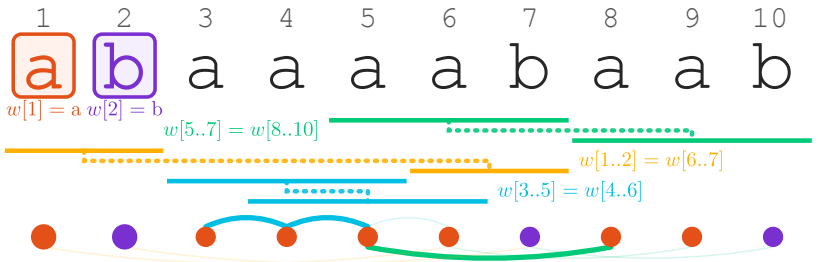
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

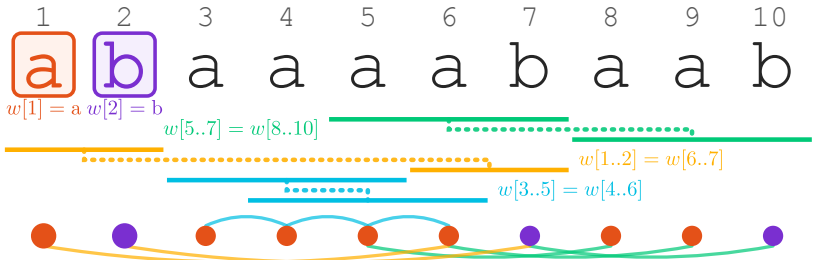
- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



SES graph の閉路削減アルゴリズム

SES graph の閉路を任意に 1 つとり、以下の操作を行う

- **Equality 制約の終端に対応する辺**がある場合:
  そのような Equality 制約の長さを 1 だけ縮め、
  SES graph から辺を 1 つ削除
- 全ての辺が **Equality 制約の終端以外に対応する**場合:
  各辺を右に 1 つずらした閉路を考えて再度処理



証明手法のアウトライン

研究の成果

任意の文字列 w に対して、$s(w) \leq b(w) \leq 4s(w)$ が成り立つ。

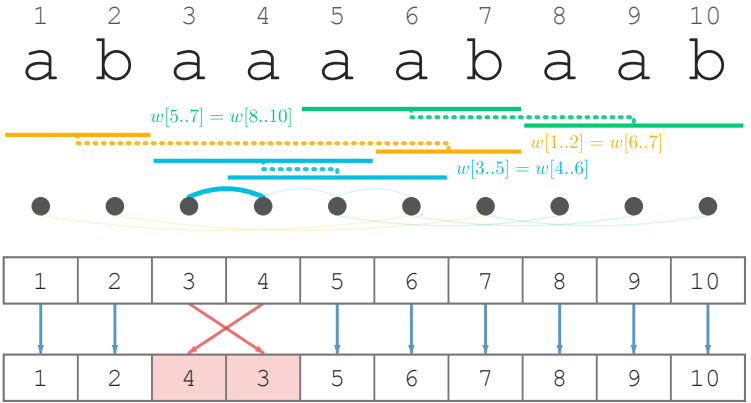
$s(w) \leq b(w)$: 前述の議論より成り立つ
$b(w) \leq 4s(w)$: **SES → BMS の変換手法を与える**ことで示す

変換手法の流れ:

1. SES を forest SES に変換する
2. **forest SES から、文字同士の一致関係を表現する置換 π を作る**
3. 置換 π から BMS を構成する

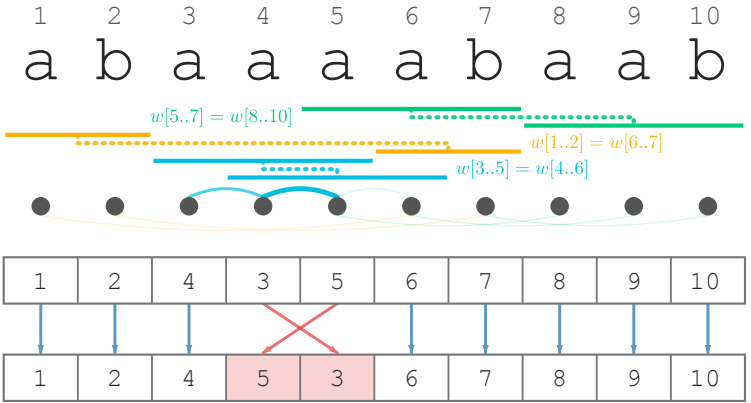
置換の合成

forest SES の辺 (i, j) に対し、i, j を入れ替える置換（互換）を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



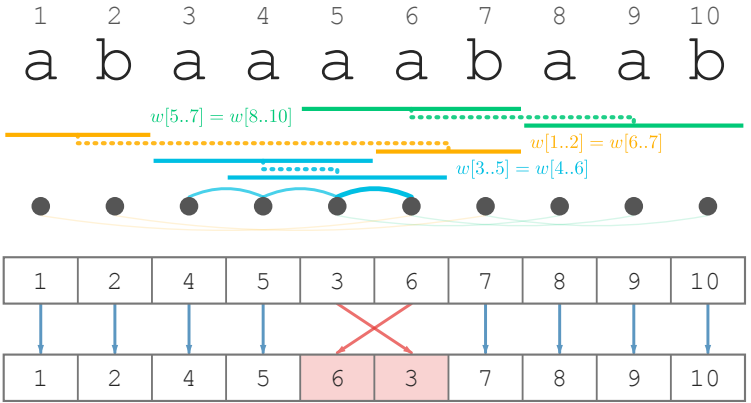
置換の合成

forest SES の辺 (i, j) に対し、i, j を入れ替える**置換（互換）**を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



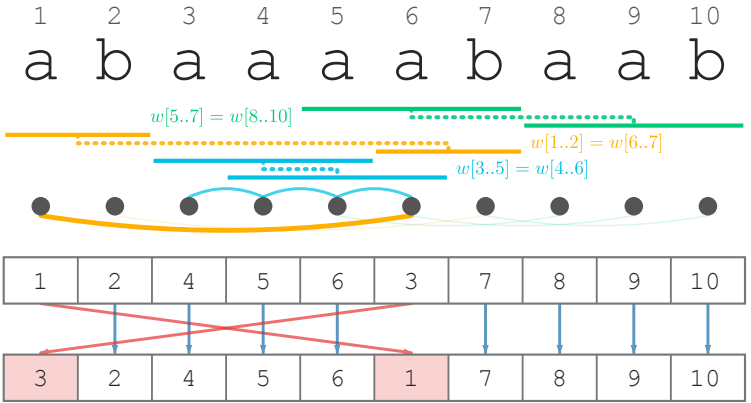
置換の合成

forest SES の辺 (i, j) に対し、i, j を入れ替える**置換（互換）**を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



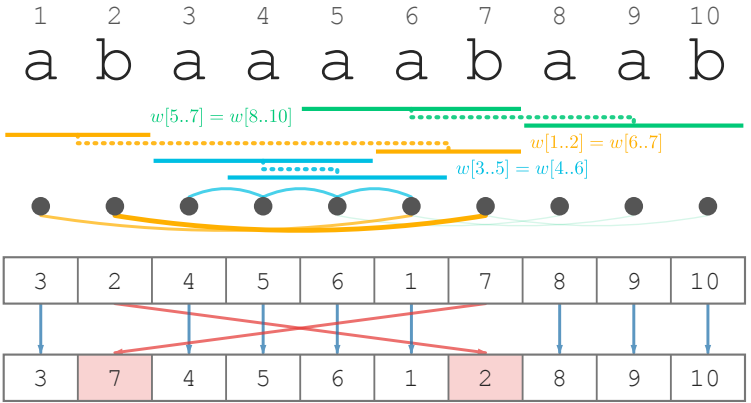
置換の合成

forest SES の辺 (i, j) に対し、i, j を入れ替える**置換（互換）**を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



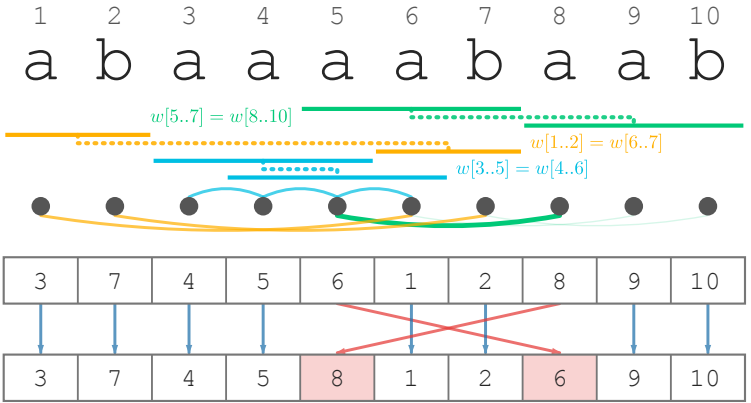
置換の合成

forest SES の辺 (i, j) に対し、i, j を入れ替える**置換（互換）**を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



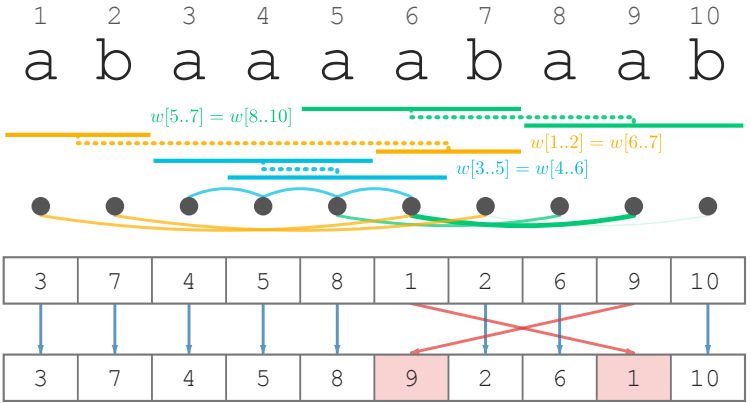
置換の合成

forest SES の辺 (i, j) に対し、i, j を入れ替える**置換（互換）**を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



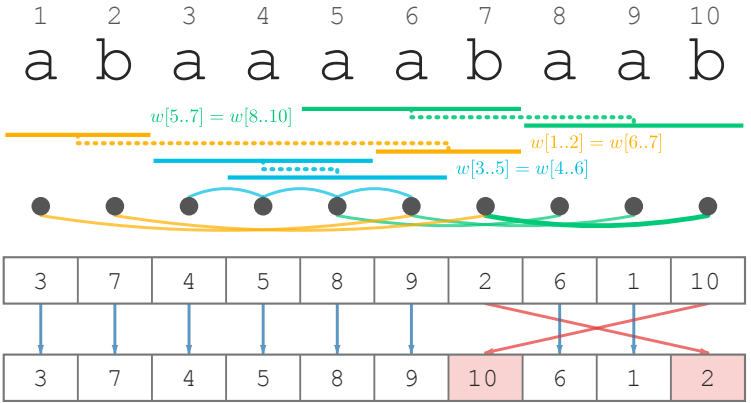
置換の合成

forest SES の辺 (i, j) に対し、i, j を入れ替える**置換（互換）**を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



置換の合成

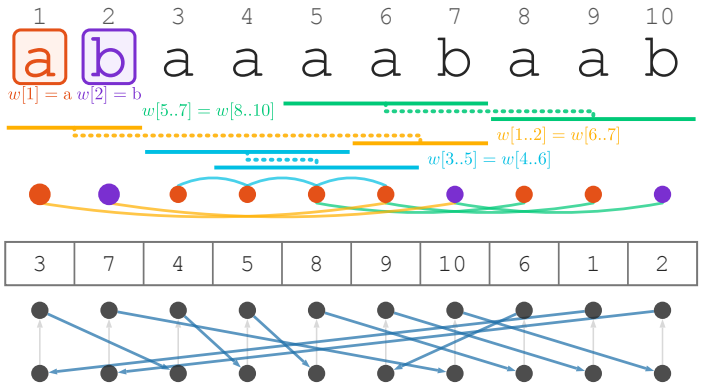
forest SES の辺 (i, j) に対し、i, j を入れ替える**置換（互換）**を考える
forest SES の各辺に対応する置換を順番に合成していき、
forest SES に対応する置換 π を構成する



合成によって得られた置換の性質

補題 2: 合成置換と SES graph の関係性

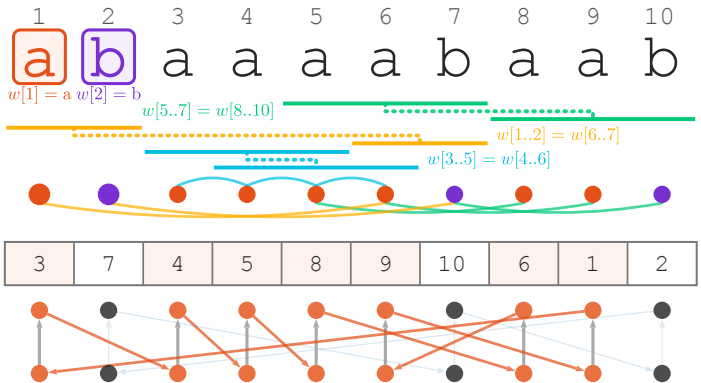
forest SES に対応する置換の軌道（下図のグラフの各サイクル）の集合は、SES graph の連結成分の集合と一致する。



合成によって得られた置換の性質

補題 2: 合成置換と SES graph の関係性

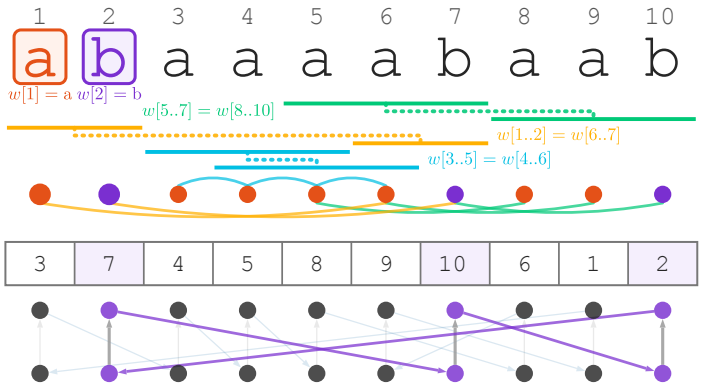
forest SES に対応する置換の軌道（下図のグラフの各サイクル）の集合は、SES graph の連結成分の集合と一致する。



合成によって得られた置換の性質

補題 2: 合成置換と SES graph の関係性

forest SES に対応する置換の軌道（下図のグラフの各サイクル）の集合は、SES graph の連結成分の集合と一致する。



証明手法のアウトライン

研究の成果

任意の文字列 w に対して、$s(w) \leq b(w) \leq 4s(w)$ が成り立つ。

$s(w) \leq b(w)$: 前述の議論より成り立つ
$b(w) \leq 4s(w)$: **SES → BMS の変換手法を与える**ことで示す

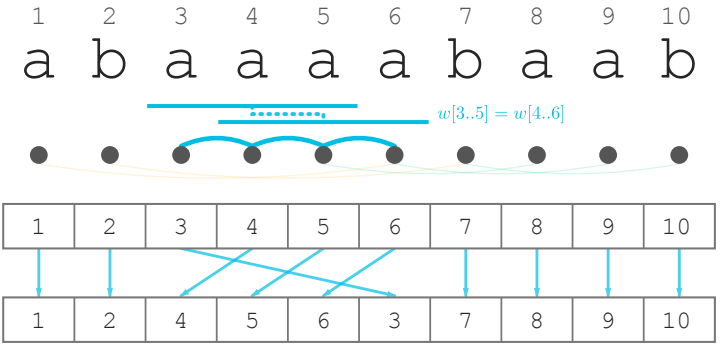
変換手法の流れ:

1. SES を forest SES に変換する
2. forest SES から、一致関係を表現する置換 π を作る
3. **置換 π から BMS を構成する**

Equality制約に対応する合成置換

ある Equality 制約 e について、**制約に対応する置換のみを合成した置換 π_e を考える**

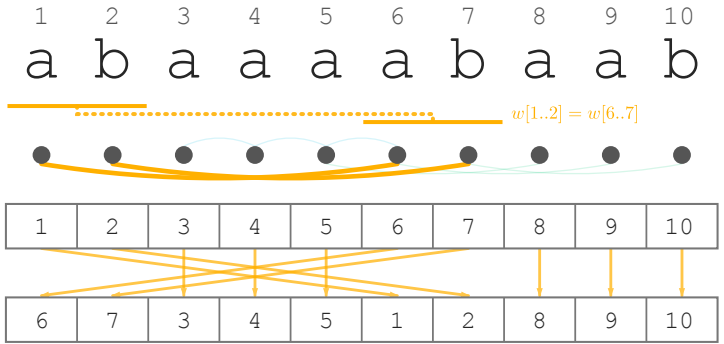
- forest SES 全体に対応する置換 π は、全ての Equality 制約に対応する π_e の合成としても表せる



Equality制約に対応する合成置換

ある Equality 制約 e について、**制約に対応する置換のみを合成した置換 π_e を考える**

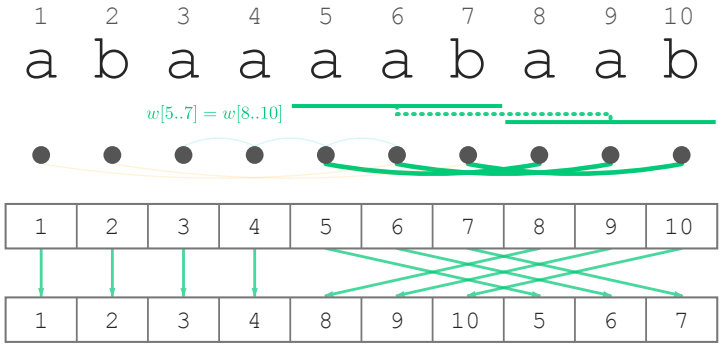
- forest SES 全体に対応する置換 π は、全ての Equality 制約に対応する π_e の合成としても表せる



Equality制約に対応する合成置換

ある Equality 制約 e について、**制約に対応する置換のみを合成した置換 π_e を考える**

- forest SES 全体に対応する置換 π は、全ての Equality 制約に対応する π_e の合成としても表せる



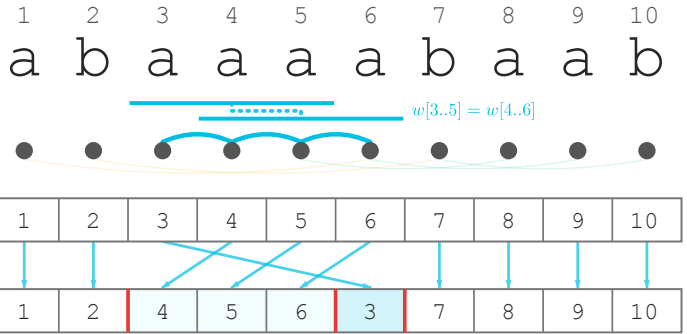
境界要素

置換 π について、**1 つ右との差が 1 でない**要素（**境界要素**）の個数を $r(\pi)$ とおく

補題 3: Equality 制約に対応する置換の境界要素数

任意の Equality 制約 e に対応する置換 π_e について、$r(\pi_e) \leq 4$ である。

主張: 各等式に対応する置換について、境界要素の数は少ない



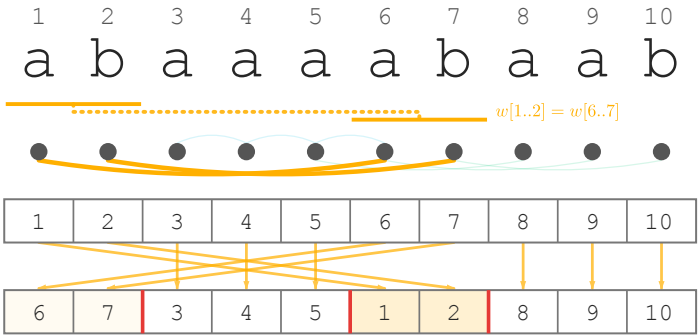
境界要素

置換 π について、**1 つ右との差が 1 でない**要素（**境界要素**）の個数を $r(\pi)$ とおく

補題 3: Equality 制約に対応する置換の境界要素数

任意の Equality 制約 e に対応する置換 π_e について、$r(\pi_e) \leq 4$ である。

主張: 各等式に対応する置換について、境界要素の数は少ない



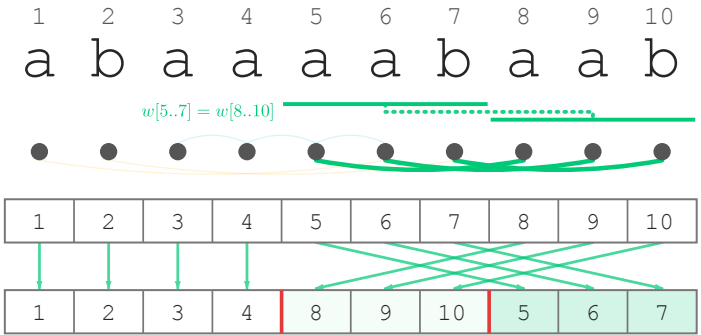
境界要素

置換 π について、**1 つ右との差が 1 でない**要素（**境界要素**）の個数を $r(\pi)$ とおく

補題 3: Equality 制約に対応する置換の境界要素数

任意の Equality 制約 e に対応する置換 π_e について、$r(\pi_e) \leq 4$ である。

主張: 各等式に対応する置換について、境界要素の数は少ない

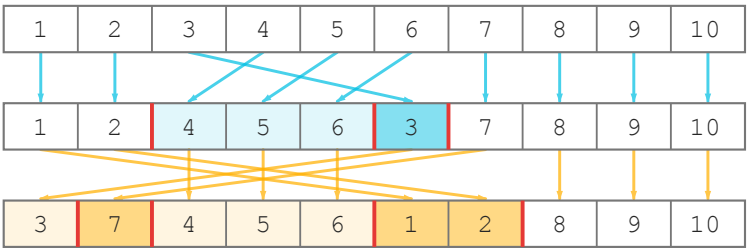


合成による境界要素数の変化

補題 4: 置換の合成による境界要素数の変化

任意の置換 π, ρ について $r(\pi \odot \rho) \leq r(\pi) + r(\rho)$ である。

主張: 合成置換の境界要素の数は、元の置換の境界要素数の合計以下

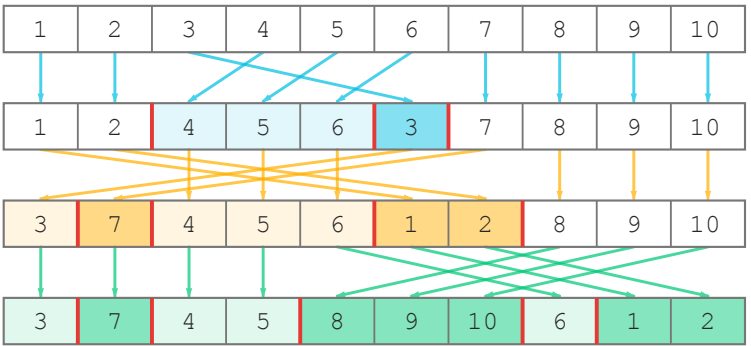


forest SES に対応する置換の境界要素数

系 1: 置換 π の境界要素数

k 個の Equality 制約をもつ SES に対応する置換 π について、$r(\pi) \leq 4k$ である。

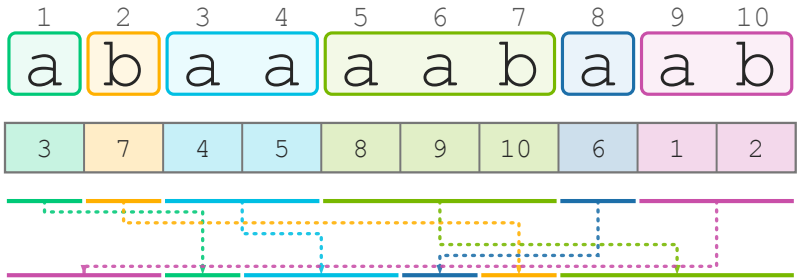
→ 置換 π の**隣接要素の差が 1 の区間**を 1 つの phrase として表せれば良い！



forest SES の合成置換からの SES の構成 (1)

隣接差分が 1 の区間を 1 つの copy phrase とし、BMS の参照関係において位置 i が位置 $\pi(i)$ を参照するようにする

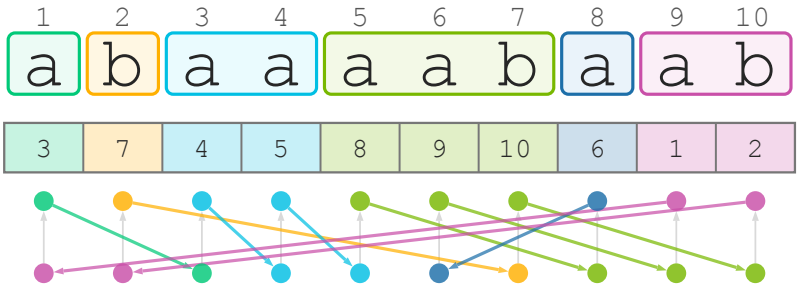
- text position 間の参照関係の**各サイクルが一つの文字に対応**
- copy phrase 数は **Equality 制約数の高々 4 倍**（前ページの議論より）



forest SES の合成置換からの SES の構成 (1)

隣接差分が 1 の区間を 1 つの copy phrase とし、BMS の参照関係において位置 i が位置 $\pi(i)$ を参照するようにする

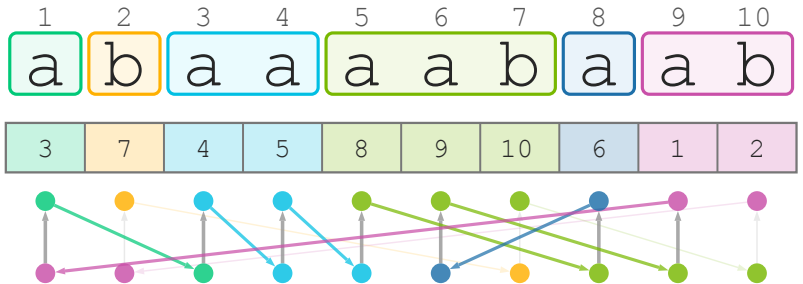
- text position 間の参照関係の**各サイクルが一つの文字に対応**
- copy phrase 数は **Equality 制約数の高々 4 倍**（前ページの議論より）



forest SES の合成置換からの SES の構成 (1)

隣接差分が 1 の区間を 1 つの copy phrase とし、BMS の参照関係において位置 i が位置 $\pi(i)$ を参照するようにする

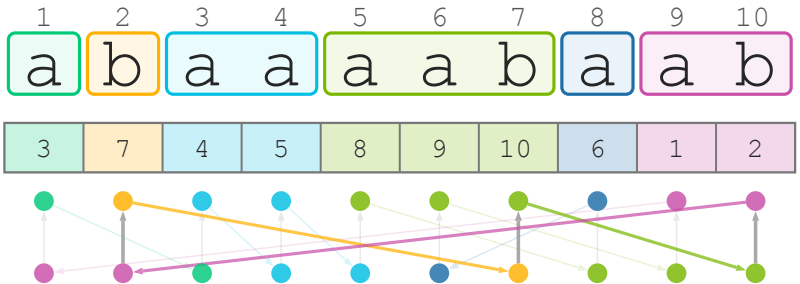
- text position 間の参照関係の**各サイクルが一つの文字**に対応
- copy phrase 数は **Equality 制約数の高々 4 倍**（前ページの議論より）



forest SES の合成置換からの SES の構成 (1)

隣接差分が 1 の区間を 1 つの copy phrase とし、BMS の参照関係において位置 i が位置 $\pi(i)$ を参照するようにする

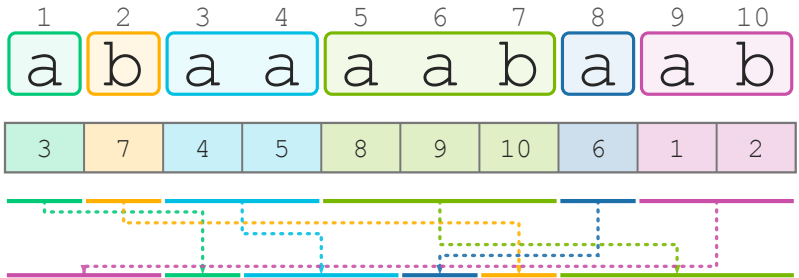
- text position 間の参照関係の**各サイクルが一つの文字**に対応
- copy phrase 数は **Equality 制約数の高々 4 倍**（前ページの議論より）



forest SES の合成置換からの SES の構成 (1)

隣接差分が 1 の区間を 1 つの copy phrase とし、BMS の参照関係において位置 i が位置 $\pi(i)$ を参照するようにする

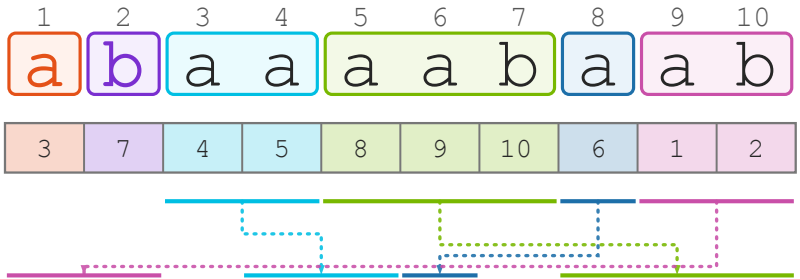
- text position 間の参照関係の**各サイクルが一つの文字**に対応
- copy phrase 数は **Equality 制約数の高々 4 倍**（前ページの議論より）



forest SES の合成置換からの SES の構成 (2)

先ほど作った BMS から、**各文字ごとに位置を 1 つ選び char phrase に置き換える**ことで、正しい BMS を構成できる

- この操作により、**参照関係に閉路がなくなる**
- この操作で増える phrase 数は、**Assignment 制約数の高々 2 倍**



サイズの解析

制約数 $s(w) = |\mathsf{Eq}| + |\mathsf{Ch}|$[1]の最小 SES からの構成を考える

1. 与えられた SES を forest SES に変換
   - SES の制約数は増えない
2. 各 Equality 制約の合成置換から copy phrase を構成
   - copy phrase の個数は高々 $4|\mathsf{Eq}|$
3. 各文字種ごとに char phrase を追加
   - 操作で増える phrase 数は高々 $2|\mathsf{Ch}|$

[1] $|\mathsf{Eq}|$: Equality 制約数, $|\mathsf{Ch}|$: Assignment 制約数

サイズの解析

制約数 $s(w) = |\mathrm{Eq}| + |\mathrm{Ch}|$[1]の最小 SES からの構成を考える

1. 与えられた SES を forest SES に変換
   - SES の制約数は増えない
2. 各 Equality 制約の合成置換から copy phrase を構成
   - copy phrase の個数は高々 $4|\mathrm{Eq}|$
3. 各文字種ごとに char phrase を追加
   - 操作で増える phrase 数は高々 $2|\mathrm{Ch}|$

[1] $|\mathrm{Eq}|$: Equality 制約数, $|\mathrm{Ch}|$: Assignment 制約数

サイズの解析

制約数 $s(w) = |\mathrm{Eq}| + |\mathrm{Ch}|$[1]の最小 SES からの構成を考える

1. 与えられた SES を forest SES に変換
   - SES の制約数は増えない
2. 各 Equality 制約の合成置換から copy phrase を構成
   - copy phrase の個数は高々 $4|\mathrm{Eq}|$
3. 各文字種ごとに char phrase を追加
   - 操作で増える phrase 数は高々 $2|\mathrm{Ch}|$

[1] $|\mathrm{Eq}|$: Equality 制約数, $|\mathrm{Ch}|$: Assignment 制約数

サイズの解析

制約数 $s(w) = |\mathrm{Eq}| + |\mathrm{Ch}|$[1]の最小 SES からの構成を考える

1. 与えられた SES を forest SES に変換
   - SES の制約数は増えない
2. 各 Equality 制約の合成置換から copy phrase を構成
   - copy phrase の個数は高々 $4|\mathrm{Eq}|$
3. 各文字種ごとに char phrase を追加
   - 操作で増える phrase 数は高々 $2|\mathrm{Ch}|$

[1] $|\mathrm{Eq}|$: Equality 制約数, $|\mathrm{Ch}|$: Assignment 制約数

サイズの解析

制約数 $s(w) = |\mathrm{Eq}| + |\mathrm{Ch}|$[1]の最小 SES からの構成を考える

1. 与えられた SES を forest SES に変換
   - SES の制約数は増えない
2. 各 Equality 制約の合成置換から copy phrase を構成
   - copy phrase の個数は高々 $4|\mathrm{Eq}|$
3. 各文字種ごとに char phrase を追加
   - 操作で増える phrase 数は高々 $2|\mathrm{Ch}|$

定理 1: 最小 SES サイズと最小 BMS サイズの関係

任意の文字列 w に対して、$s(w) \leq b(w) \leq 4s(w)$ が成り立つ。

[1] $|\mathrm{Eq}|$: Equality 制約数, $|\mathrm{Ch}|$: Assignment 制約数

まとめ

研究の成果

任意の文字列 w に対して、$s(w) \leq b(w) \leq 4s(w)$ が成り立つ。

今回の結果の嬉しさ

- 今回の結果より、**smallest suffixient set** のサイズ $\chi(w)$ について、$\chi(w) \in O(b(w))$ が即座にいえる
  - 先行研究で $s(w) \in O(\chi(w))$ を示していた
  - **episturmian word** に対する定数サイズ BMS の存在もいえる
- BMS を考えても SES を考えてもオーダーは同じなため、今後の最小 BMS サイズの解析が楽になるかも？