

# SHORTEST COMMON SUPERSTRING

**Eingabe:** eine Menge  $U$  von  $n$  Strings;

**Ausgabe:** ein String  $S$  minimaler Länge der alle Strings in  $U$  als Teilstrings enthält;

(Ein String  $u$  ist ein endliches Wort über einem endlichen Alphabet  $\Sigma$ .)

Es ist annehmbar dass kein String in  $U$  Teilstring eines anderen ist.

Anwendung:

*Sequenzierung:*

die Reihenfolge von Basenpaaren in einem DNA-Molekül zu bestimmen

# Einführung

Def. 1: Für Strings  $u$  und  $v$  ist  $\text{overlap}(u, v)$  die *Länge* des längsten Suffix von  $u$  der auch Präfix von  $v$  ist.

Def. 2:  $\text{Präfix}(u, v)$  ist der nicht von  $v$  überdeckte Präfix in  $u$ .

Def. 3: Eine Bijektion  $\pi : \{1, 2, \dots, n\} \rightarrow U$  definiert eine Reihenfolge der Strings in  $U$ ,

$$\pi(1), \pi(2), \dots, \pi(n)$$

und damit den kürzesten Superstring  $S(\pi)$

*für diese Reihenfolge:*

$$S(\pi) = \text{Präfix}(\pi(1), \pi(2)) \cdot \dots \cdot \text{Präfix}(\pi(n-1), \pi(n)) \cdot \pi(n)$$

$S(\pi)$  ist der durch  $\pi$  induzierte Superstring von  $U$

# Beobachtungen

Behauptung 1. Wenn  $S^*$  ein kürzester Superstring ist, dann existiert eine Reihenfolge  $\pi$  mit  $S^* = S(\pi)$ .

Behauptung 2.

$$|S(\pi)| = \sum_{u \in U} |u| - \sum_{i=1}^{n-1} \text{overlap}(\pi(i), \pi(i+1))$$

Also: wir brauchen eine Reihenfolge der Strings in  $U$  die die Summe der Overlaps maximiert!

# 1. Greedy-Superstring Algorithmus

Sei  $U$  die Menge der Strings

REPEAT

- bestimme zwei Strings  $u \neq v$  mit maximalem  $\text{overlap}(u, v)$ ;
- 'verschmelze'  $u$  und  $v$  zu einem String

UNTIL  $|U| = 1$ ;

Gib den einzigen String in  $U$  als Superstring aus.

Theorem: Der Greedy-Superstring algorithmus ist 4-approximativ.  
(Ohne Beweis.)

Vermutung: Er ist sogar 2-approximativ.

# Overlap-Graph

- sei  $G(U, E)$  ein *vollständiger gerichteter* Graph über den Strings in  $U$  als Knoten; mit  $\text{overlap}(u, v)$ ; als Kantengewichten
- eine Reihenfolge der Strings mit maximaler Summe der Overlaps

=

ein Hamiltonscher Pfad mit maximalem Gesamtgewicht seiner Kanten

- leider ist max-HAMILTONSCHER-PFAD NP-schwer ...
- ... man verwendet Approximationsalgorithmen

## 2. Noch ein Algorithmus (Vorbereitung)

- (leider ist max-HAMILTONSCHER-PFAD NP-schwer)
- suche (*gerichtete*) *Kreis-Zerlegung* (*directed cycle cover*) mit maximalem Gewicht

### Definition:

Die Kantenmenge  $E' \subseteq E$  ist eine Kreis-Zerlegung



$E'$  besteht aus disjunkten Kreisen die gemeinsam jeden Knoten einmal überdecken



jeder Knoten hat eine eingehende und eine ausgehende Kante

# Fragen

1. Wie berechnet man eine maximale *Kreis-Zerlegung*?
2. Wie benutzt man die Kreis-Zerlegung für SHORTEST COMMON SUPERSTRING?
3. Approximationsfaktor?

# 1. Ein Greedy Algorithmus für maximale Kreis-Zerlegung

Eingabe: ein Overlap-Graph

setze  $E' = \emptyset$  (initialisiere Kreis-Zerlegung)

REPEAT

- bestimme Kante  $(u, v) \in E$  mit maximalem Gewicht;  
setze  $E := E \setminus \{(u, v)\}$ , und  $E' := E' \cup \{(u, v)\}$
- entferne aus  $E$  alle aus  $u$  ausgehende und in  $v$  eingehende Kanten

UNTIL  $E = \emptyset$

Theorem: Der Algorithmus berechnet eine maximale Kreis-Zerlegung im Overlap-Graph.

**NUR IM OVERLAP-GRAPH OPTIMAL !!**



Der Beweis dass dieser Greedy Algorithmus optimal ist benutzt das folgende

Lemma:

Falls

$$\text{overlap}(u, v) \geq \text{overlap}(u, v^*)$$

und

$$\text{overlap}(u, v) \geq \text{overlap}(u^*, v)$$

dann gilt

$$\text{overlap}(u, v) + \text{overlap}(u^*, v^*) \geq \text{overlap}(u^*, v) + \text{overlap}(u, v^*)$$

## 2. Ein Kreis im Overlap-Graph

Ein gerichteter Kreis  $C(u_1, u_2, \dots, u_m, u_1)$  im Overlap-Graph entspricht einem String

$$v = \text{Praefix}(u_1, u_2) \cdot \text{Praefix}(u_2, u_3) \cdot \dots \cdot \text{Praefix}(u_m, u_1).$$

Beobachtung 1:  $v^\infty$  bezeichne den unendlich oft mit sich selbst konkatenierten String  $v$ .

All die  $u_1, u_2, \dots, u_m$  sind Teilstrings von  $v^\infty$ .

$v$  wird auch *Zyklus* genannt,

und alle  $u_i$  (länger als  $v$ ) sind zyklisch mit Periode  $v$

## Die Kreise einer Kreis-Zerlegung

Der Zyklus  $v$  von einem Kreis  $C(u_1, u_2, \dots, u_m, u_1)$  hat Länge

$$|v| = \sum_{i=1}^m |\text{Praefix}(u_i, u_{i+1})| = \sum_{i=1}^m |u_i| - \sum_{i=1}^m \text{overlap}(u_i, u_{i+1}) \quad \text{mod } m.$$

wobei  $m + 1 = 1 \pmod m$

Beobachtung 2: Seien  $C_1, \dots, C_l$  die Kreise einer Kreis-Zerlegung  $E'$ , und die Strings  $v_1, v_2, \dots, v_l$  ihre Zyklen. Dann gilt

$$\sum_{k=1}^l |v_k| = \sum_{i=1}^n |u_i| - \sum_{(u,v) \in E'} \text{overlap}(u, v) = \sum_{i=1}^n |u_i| - \text{Gewicht der } E'.$$

# Zyklus-Überdeckung

Definition: Die Zyklen aller Kreise  $v_1, v_2, \dots, v_\ell$  ergibt eine Zyklus-Überdeckung von  $U$  weil jeder  $u \in U$  von einem  $v_k$  zyklisch überdeckt wird.

Beobachtung 3.:  $v_1, v_2, \dots, v_\ell$  ist eine *minimale* Zyklus-Überdeckung  
(d.h.  $\sum_k |v_k|$  minimal)



$E'$  eine Kreis-Zerlegung mit maximalem Gewicht.

Theorem: Sei  $S^* = S(\pi)$  ein *Shortest Common Superstring* von  $U$ , dann gilt

$$|S^*| \geq \sum_{k=1}^{\ell} |v_k|$$

# Algorithmus für SHORTEST COMMON SUPERSTRING

1. Bestimme eine Kreis-Zerlegung mit maximalem Gewicht  $C_1, C_2, \dots, C_\ell$  im Overlap-Graph
2. sei  $v_k$  der Zyklus von Kreis  $C_k = (u_1^k, u_2^k, \dots, u_{m_k}^k)$
3. Sei  $S_k$  der Zyklus  $v_k$  *aufgebrochen*,  
so dass  $S_k$  jeden String in  $C_k$  enthält;  
dann gilt  $|S_k| \leq |u_1^k| + |v_k|$ .
4. Gib die Konkatenation aller  $S_k$  als Superstring aus:

$$S = S_1 \cdot S_2 \cdot \dots \cdot S_\ell$$

### 3. Approximationsfaktor

Theorem 1: Der obige Algorithmus für SHORTEST COMMON SUPERSTRING ist 4-approximativ.

Theorem 2: Bei 'nicht-periodischen Strings', also wenn in jedem Kreis  $C_k$  einen String  $u_i^k$  gibt mit  $|u_i^k| \leq |v_k|$  ist er 2-approximativ.

## Approximationsfaktor Analyse

Sei  $S'$  ein kürzester Superstring aller ersten Strings in jedem Kreis:

$$u_1^1, u_1^2, \dots, u_1^k, \dots, u_1^\ell$$

Dann gilt

$$\begin{aligned} |S^*| &\geq |S'| = \sum_{k=1}^{\ell} |u_1^k| - \sum_{k=1}^{\ell-1} \text{overlap}(u_1^k, u_1^{k+1}) \geq \\ &\sum_{k=1}^{\ell} |u_1^k| - \sum_{k=1}^{\ell-1} (|v_k| + |v_{k+1}|) \geq \sum_{k=1}^{\ell} |u_1^k| - 2 \cdot \sum_{k=1}^{\ell} |v_k| \\ \Rightarrow \quad &\sum_{k=1}^{\ell} |u_1^k| \leq |S^*| + 2 \cdot \sum_{k=1}^{\ell} |v_k| \leq 3 \cdot |S^*| \end{aligned}$$