

Modul *Molekulare Algorithmen*

Projektvortrag

DNA-Algorithmus für das Partitionsproblem

Lennart Dabelow

06.07.2015

Übersicht

- Problemstellung
- DNA-Algorithmus zur Lösung
- Bewertung des Algorithmus

Das Partitionsproblem

- Ausgangspunkt: Multimenge $S = \{s_1, \dots, s_n\}$ mit $s_i \in \mathbf{N}$ („Gewichte“)
- Problemstellung: Können die Gewichte ausbalanciert werden?

Konkret:

Gibt es eine Menge $T \subset \{1, \dots, n\}$, sodass $\sum_{k \in T} s_k = \sum_{k \in \bar{T}} s_k$?

- Zusammenhang mit Rucksack-Problem:

Zielgewicht $b = \frac{1}{2} \sum_{k \in \{1, \dots, n\}} s_k$

Grundidee

- $1 \triangleq \mathbf{1}$, z.B. AC
- $s_i \triangleq \underbrace{\mathbf{1} \mathbf{1} \dots \mathbf{1}}_{s_i}$, z.B. $4 \triangleq \text{ACACACAC}$
- Kennzeichne abgeschlossene Sequenz (Teilmenge der s_i) durch Startsymbol \mathbf{S} (z.B. GG) und Endsymbol \mathbf{E} (z.B. TT)
- $T \triangleq \mathbf{S} \underbrace{\mathbf{1} \dots \mathbf{1}}_{s_{t_1}} \dots \underbrace{\mathbf{1} \dots \mathbf{1}}_{s_{t_{|T|}}} \mathbf{E}$, z.B. $\{2, 3\} \triangleq \text{GGACACACACTT}$
mit $s_2 = 1, s_3 = 4$
- Erzeugung aller möglichen Partitionen der s_i
- Vergleich mit Referenzsequenz der benötigten Länge b

Lösungsschritte

- 1** Synthese der Sequenzen für die Gewichte s_i sowie Start- und Endsymbol als (um eine Base verschobene) Doppelstränge

$$S = \begin{array}{l} GG \\ CCT \end{array} \quad E = \begin{array}{l} ATT \\ AA \end{array} \quad 1 = \begin{array}{l} AC \\ GT \end{array} \quad 2 = \begin{array}{l} ACAC \\ GTGT \end{array} \quad 3 = \begin{array}{l} ACACAC \\ GTGTGT \end{array} \quad \dots$$

- 2a** Zusammensetzen aller s_i -Sequenzen zu einer Gesamtgewichtssequenz

$$\sum_{k \in \{1, \dots, n\}} s_k = \begin{array}{l} GGACACACACACACACACATT \\ CCTGTGTGTGTGTGTGTGTAA \end{array}$$

$\underbrace{\hspace{1.5em}}_S \quad \underbrace{\hspace{2.5em}}_{s_1 = 3} \quad \underbrace{\hspace{3.5em}}_{s_2 = 5} \quad \underbrace{\hspace{2.5em}}_{s_3 = 2} \quad E$

- 2b** Erzeugung aller möglichen Partitionen der s_i

$$\text{z.B. } \{s_2, s_3\} = \begin{array}{l} GGACACACACACACATT \\ CCTGTGTGTGTGTGTAA \end{array}$$

$\underbrace{\hspace{1.5em}}_S \quad \underbrace{\hspace{4.5em}}_{s_2 = 7} \quad \underbrace{\hspace{2.5em}}_{s_3 = 2} \quad E$

Lösungsschritte (2)

- 3** Erzeugung der Referenzsequenz der Länge b aus der Gesamtsequenz
mit Länge $2b$

$$M = \begin{array}{c} \text{GGACACACACACATT} \\ \text{CCTGTGTGTGTGTAA} \end{array}$$

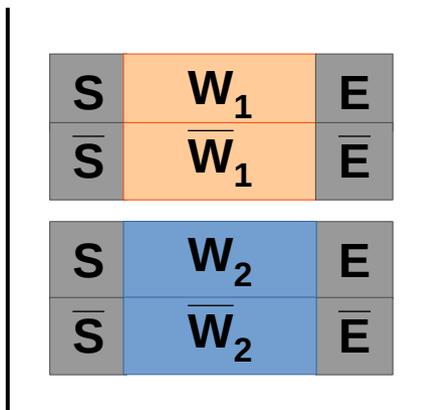
$\underbrace{\hspace{1.5cm}}_S \quad \underbrace{\hspace{3.5cm}}_5 \quad \underbrace{\hspace{1.5cm}}_E$

- 4** Vergleich auf Übereinstimmung der Partitionen mit Referenzsequenz
z.B. $\{s_2, s_3\} \neq M$

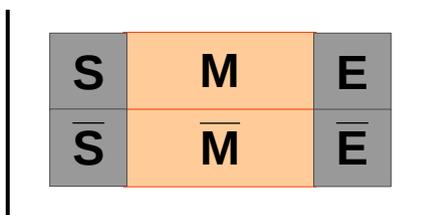
Sequenzvergleich

Eingabe

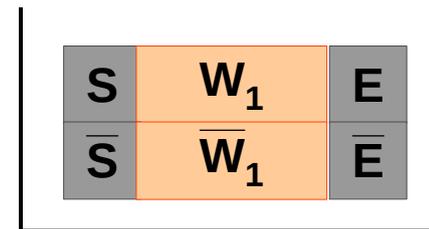
- Probe



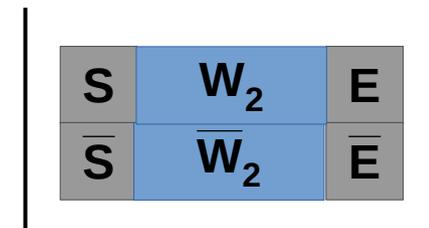
- Muster



Ausgabe



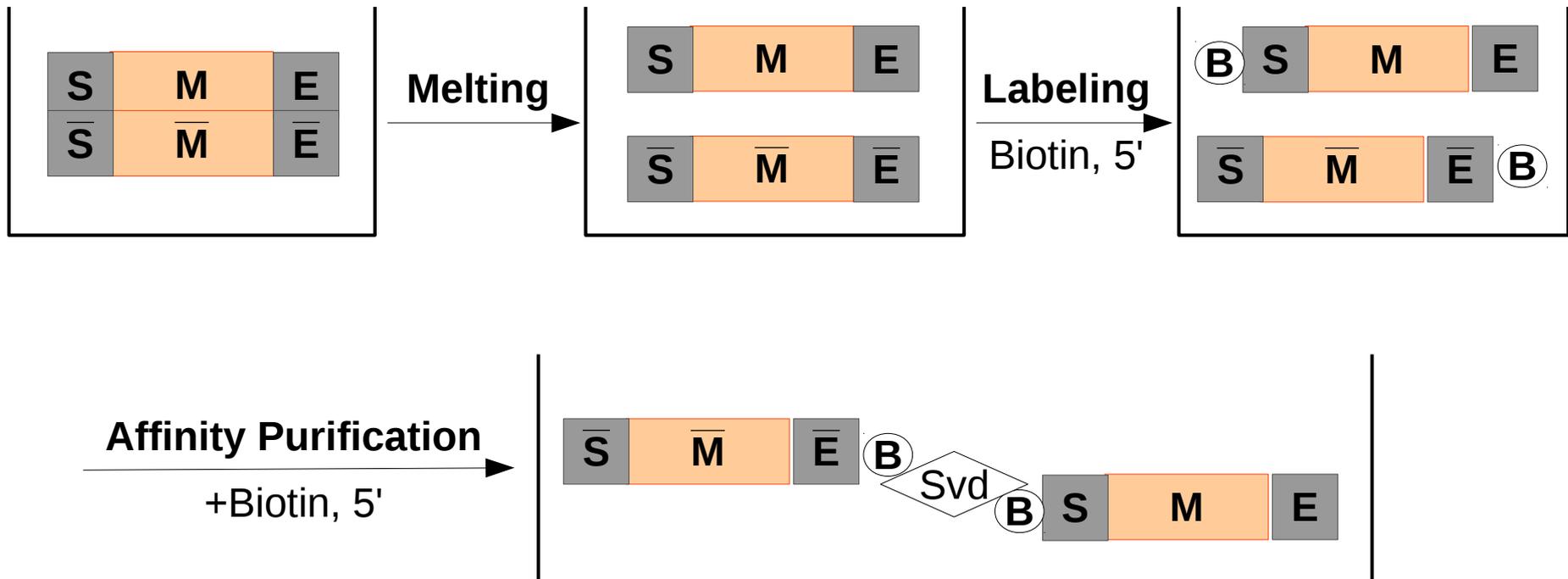
$$W_1 = M$$



$$W_1 \neq M$$

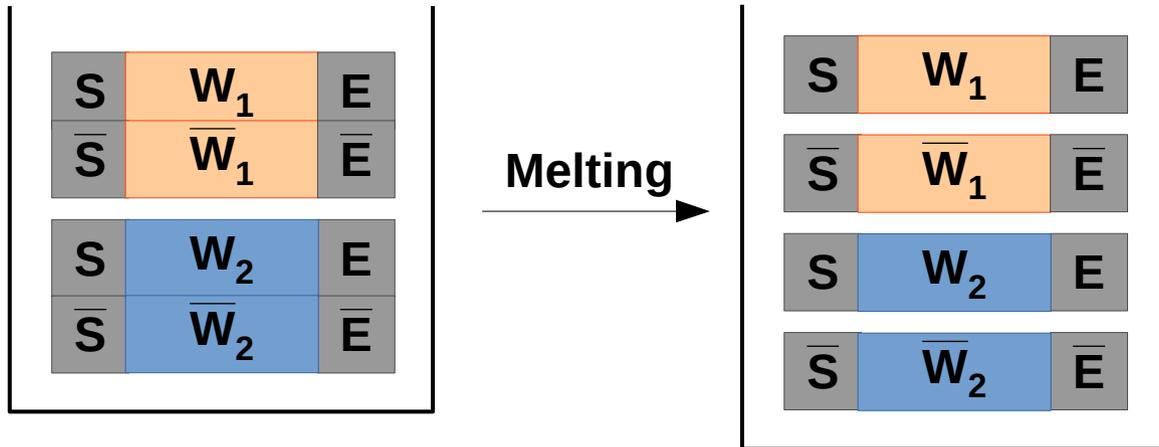
Sequenzvergleich (2)

Prozess



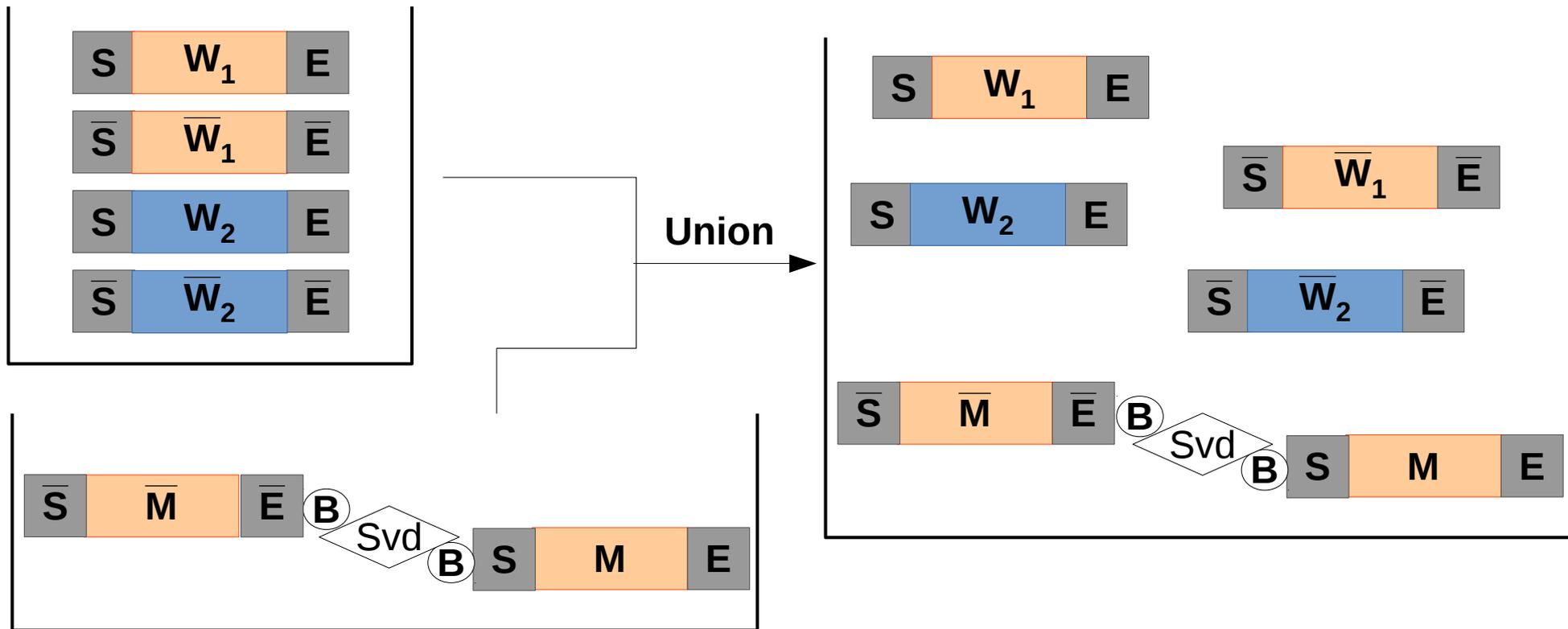
Sequenzvergleich (2)

Prozess (2)



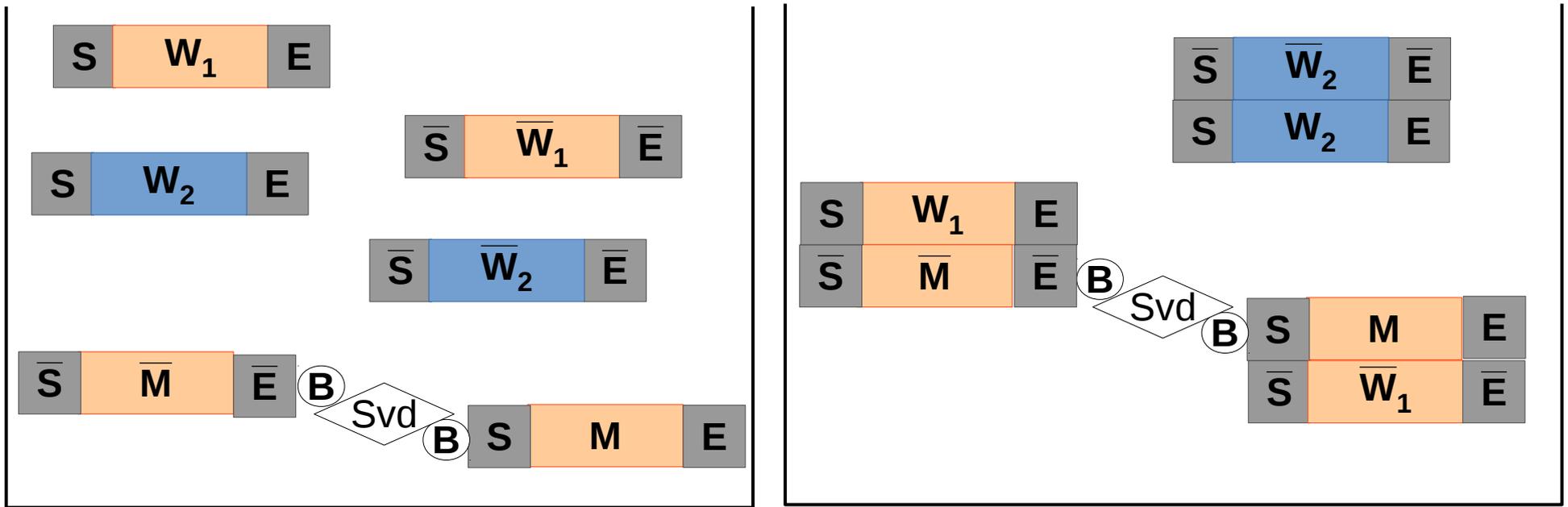
Sequenzvergleich (2)

Prozess (3)



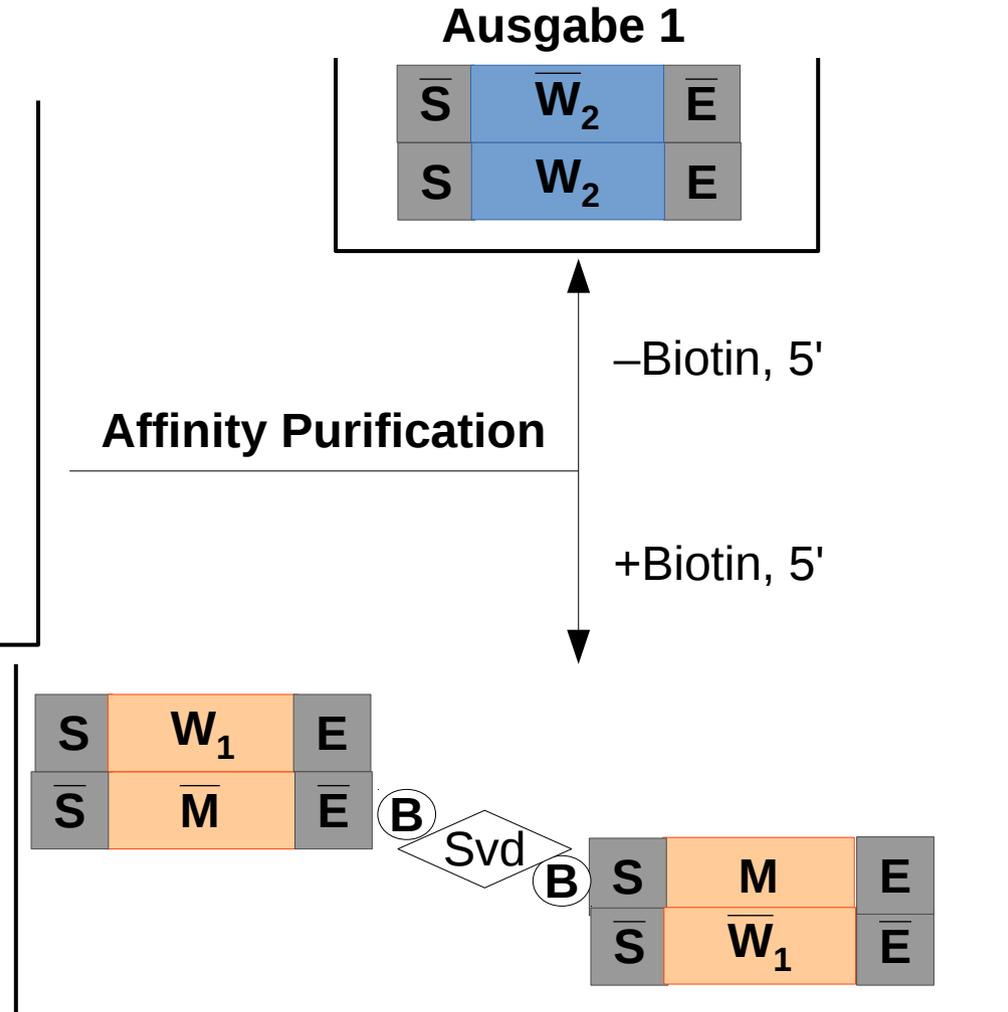
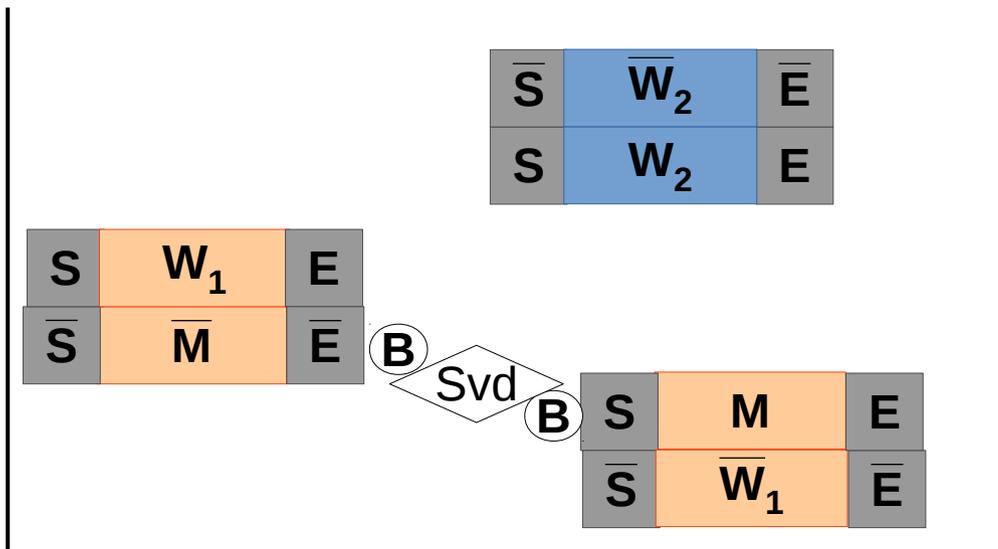
Sequenzvergleich (2)

Prozess (4)



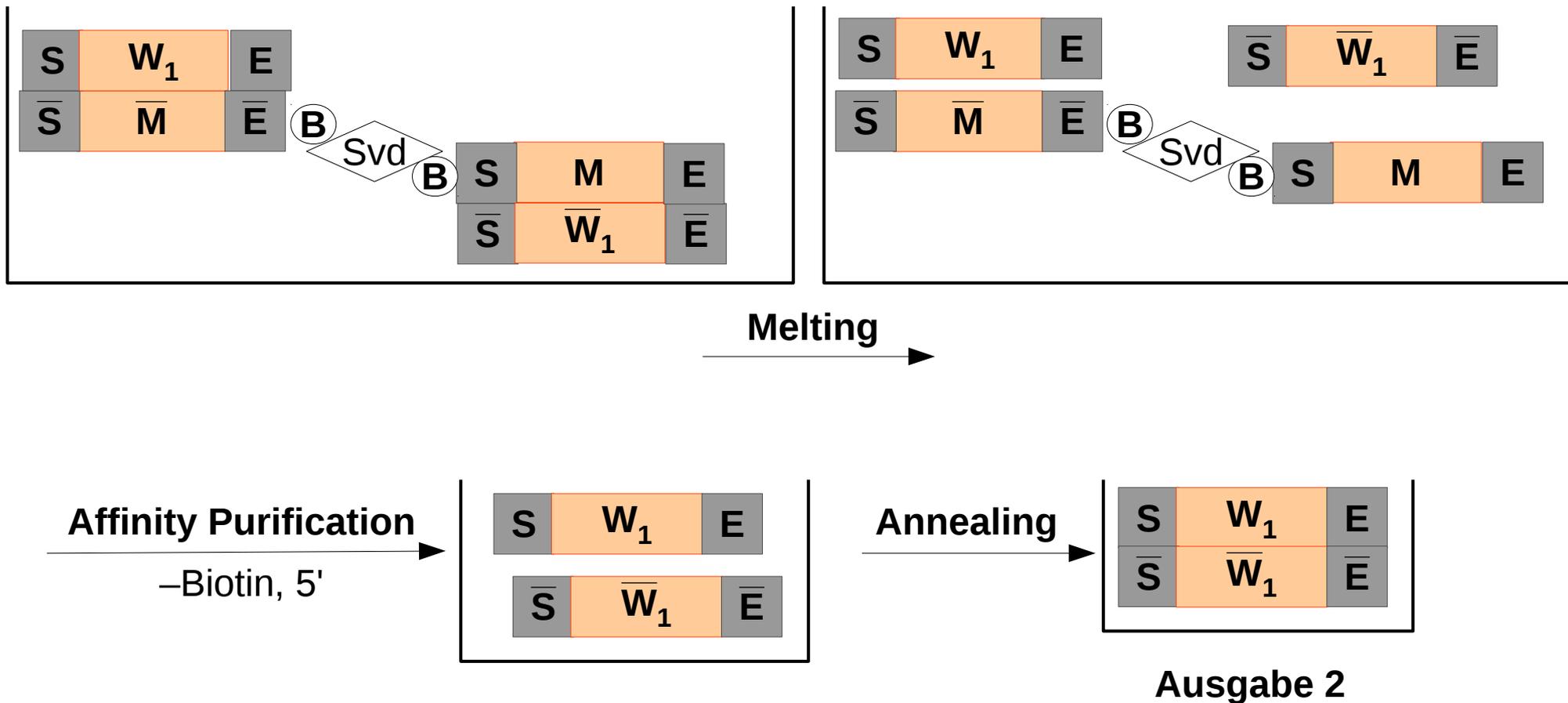
Sequenzvergleich (2)

Prozess (5)



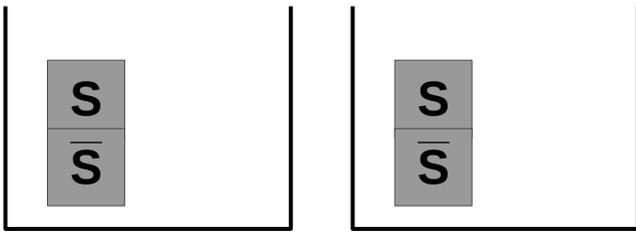
Sequenzvergleich (2)

Prozess (6)



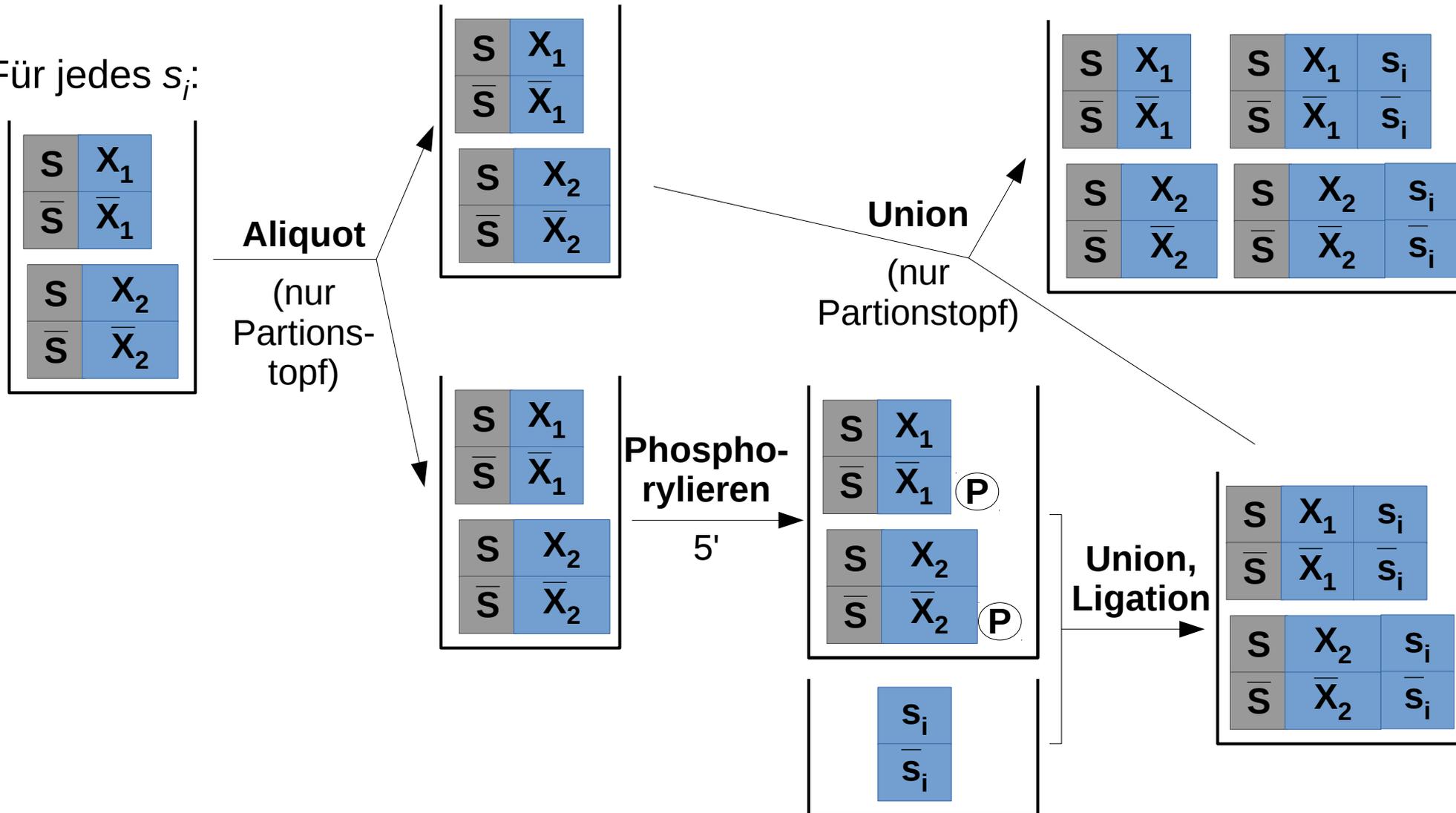
Erzeugung der Partitionen

- zwei Töpfe: Partitionen und Gesamtsequenz
- Start:



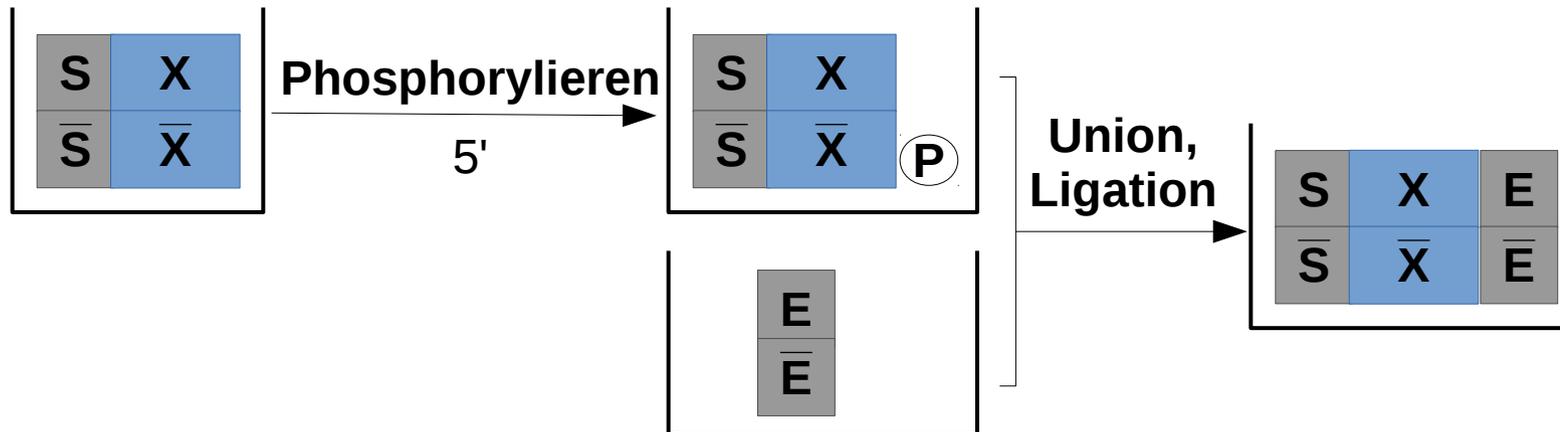
Erzeugung der Partitionen (2)

- Für jedes s_i :



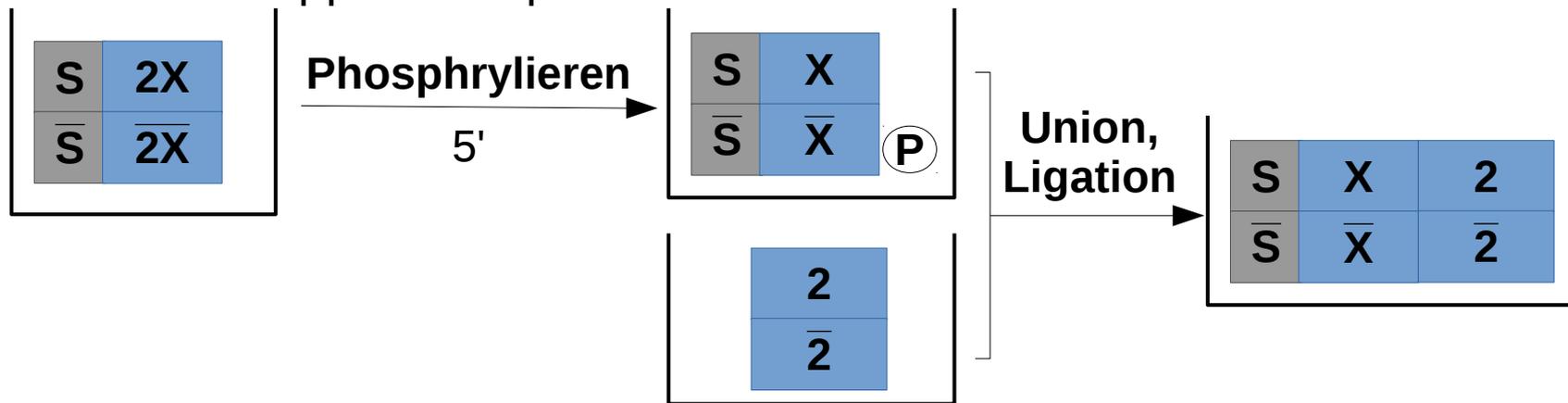
Erzeugung der Partitionen (3)

- Ende (nur Partitionstopf):



Erzeugung der Referenzsequenz

- zwei Bausteine: **1** und **2** = **1 1**
- zwei Töpfe: einfach und doppelt
- Iteration für doppelten Topf:



- Iteration für einfachen Topf: analog mit einfachem Baustein



Erzeugung der Referenzsequenz (2)

- nach jedem Iterationsschritt:
vergleiche doppelten Topf mit Gesamtlängensequenz

$$\begin{array}{|c|c|} \hline \mathbf{S} & \mathbf{2X} \\ \hline \overline{\mathbf{S}} & \overline{\mathbf{2X}} \\ \hline \end{array} \stackrel{?}{=} \begin{array}{|c|c|} \hline \mathbf{S} & \mathbf{2b} \\ \hline \overline{\mathbf{S}} & \overline{\mathbf{2b}} \\ \hline \end{array}$$

- falls ja: einfacher Topf hat Länge b

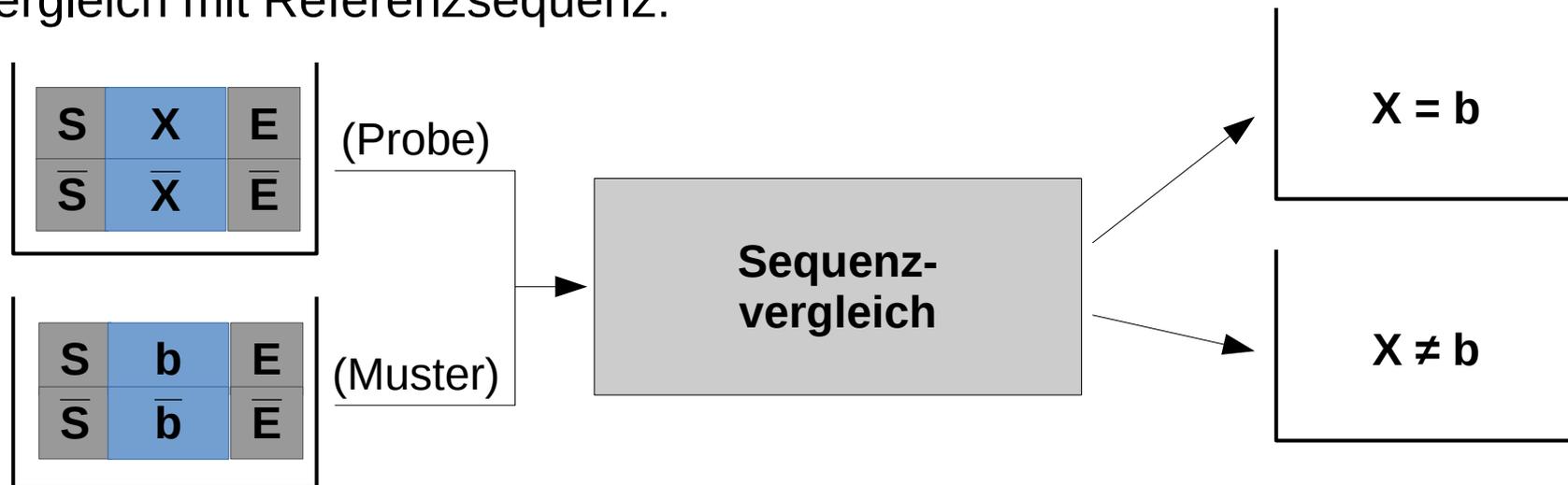
$$\Rightarrow \begin{array}{|c|c|} \hline \mathbf{S} & \mathbf{X} \\ \hline \overline{\mathbf{S}} & \overline{\mathbf{X}} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \mathbf{S} & \mathbf{b} \\ \hline \overline{\mathbf{S}} & \overline{\mathbf{b}} \\ \hline \end{array}$$

- terminiere einfachen Topf mit Endsymbol

$$\mathbf{M} = \begin{array}{|c|c|c|} \hline \mathbf{S} & \mathbf{b} & \mathbf{E} \\ \hline \overline{\mathbf{S}} & \overline{\mathbf{b}} & \overline{\mathbf{E}} \\ \hline \end{array}$$

Vergleich der Partitionen mit Referenzsequenz

- Probe enthält alle möglichen 2^n Teilmengen der s_i
- Ausbalancieren genau dann möglich, wenn Partition der Länge b (und damit auch ihr Komplement) enthalten sind
- Vergleich mit Referenzsequenz:



- Entscheidungsproblem: **ja**, falls (X=b)-Topf nicht leer, sonst **nein**

Komplexität der Einzelschritte

- 1** Synthese der Sequenzen für die Gewichte s_i sowie Start- und Endsymbol
Zeit: n
- 2a** Zusammensetzen aller s_i -Sequenzen zu einer Gesamtgewichtssequenz
Zeit: n Speicher: n
- 2b** Erzeugung aller möglichen Partitionen der s_i
Zeit: n Speicher: 2^n
- 3** Erzeugung der Referenzsequenz
Zeit: n Speicher: n
- 4** Vergleich auf Übereinstimmung der Partitionen mit Referenzsequenz
Zeit: 1

Probleme/Verbesserungsvorschläge

- Lösungspartition im Falle positiver Antwort nicht rekonstruierbar
- Sequenzlänge und damit maximales Gewicht technisch beschränkt
- Erstellen der Referenzsequenz aufgrund des Sequenzvergleichs in jeder Iteration aufwendig
 - falls Berechnung der Referenzlänge „außerhalb von DNA“ zulässig, kann Referenzsequenz vorab als Eingabeparameter bereitgestellt werden (Rucksack-Problem)
- sofortiges Aussortieren zu langer Partitionen in jedem Schritt der Erzeugung
 - Reduktion des Speicherbedarfs

Fragen?

Literatur

- Hinze, T., *Molekularbiologische Operationen für das DNA-Computing*. Technische Universität Dresden, 1999.
- Hinze, T., *Rechnen mit DNA*. Oldenbourg Wissenschaftsverlag, 2004.
- Tsai, S. und Chen, W.-Y., *Molecular Solutions For The Set-Partition Problem On Dna-Based Computing*. IJCSIT **5** (6), 2004.