

Formale Sprachen: DNA Computing

Adleman-Lab: Prototyp eines DNA-Computers

Zur Verfügung stehen Reagenzgläser, denen man einzeln einen Namen T geben kann oder als Batterie (Feld) $(T_{i_1 \dots i_k})_{i_j \in [n_j], j \in [k]}$ für $k \in \mathbb{N}$. Dabei können die oberen Grenzen der einzelnen Indizes frei gewählt werden; ebenso ist die Dimension k frei wählbar. Anfangs sind alle Reagenzgläser leer.

Alle benannten Reagenzgläser können pro Arbeitsschritt (Berechnungsschritt) befüllt werden. Ein Arbeitsschritt wird durch einen Pfeil der Form \leftarrow notiert. Links stehen die Reagenzgläser, deren Füllung verändert werden soll; rechts steht, wodurch das geschieht.

Es gibt zwei prinzipielle Möglichkeiten, die Füllung zu ändern: Zufügen und Selektieren. Zufügen lassen sich einzelne DNA-Moleküle oder die Füllung von Reagenzgläsern. Werden mehrere Komponenten zugefügt, werden sie durch $+$ getrennt. Die ursprüngliche Füllung bleibt vorhanden. Von benannten Reagenzgläsern wird der Inhalt eingefüllt, den sie vor dem Arbeitsschritt haben.

Selektion aus Reagenzgläsern ist mit einem DNA-Einzelstrang s möglich, wobei $select(s, T)$ alle Moleküle von T enthält, in denen s vorkommt. Außerdem kann nach Länge der Moleküle selektiert werden; $select(p, T)$ enthält alle Moleküle aus T der Länge p . Mit einem $-$ davor kann man das Komplement selektieren. Also enthält $-select(s, T)$ alle Moleküle aus T , die s nicht als Teilsequenz haben, und $-select(p, T)$ alle, die nicht die Länge p besitzen. Es gibt noch eine spezielle Selektion: $blunt(T)$ enthält alle vollständigen Doppelmoleküle aus T . Beim Selektieren bleibt die ursprüngliche Füllung nicht erhalten.

Schließlich gibt es noch einen Test, der für ein Reagenzglas ermittelt, ob es leer oder gefüllt ist: $empty?(T)$.

Es wird vorausgesetzt, dass alle Moleküle immer in ausreichender Zahl in den Lösungen vorhanden sind und dass beim Zufügen alle möglichen Verbindungsreaktionen auf der Basis der Watson-Crick-Komplementarität stattfinden.

Zufügen und Selektieren können außerdem noch von Bedingungen abhängig gemacht werden. Die Arbeitsschritte können sequentiell komponiert werden, sie können außerdem x -mal oder für $i = 1, \dots, x$ wiederholt werden ($x \in \mathbb{N}$).

Um mit dieser Art Verfahren Probleme zu lösen, kann man einen passenden Eingabebereich wählen. Von der Eingabe hängt dann die Wahl der Reagenzgläser ab und die Wahl der Einzelmoleküle, die als Einzelstränge und durch Zusammenfügen in einer Vorverarbeitungsphase hergestellt werden, sowie die Bedingungen, die pro Arbeitsschritt gefordert werden können. Inwieweit am Ende des Verfahrens das Problem gelöst ist, muss durch eine Korrektheitsaussage formuliert werden, die den Zusammenhang zwischen Problemlösung und Inhalt der Reagenzgläser herstellt.

Das folgende Beispiel soll verdeutlichen, wie die allgemeinen Ausführungen gemeint sind. Ziel ist, die Wege eines gerichteten Graphen bis zu einer bestimmten Länge zu konstruieren. Eine Eingabe besteht deshalb aus einem gerichteten Graphen, dessen Knoten durchnummeriert sind und der von jedem Knoten zu jedem Knoten höchstens eine Kante hat, und aus einer Zahl, die die Weglänge beschränkt. In der Vorverarbeitung werden Einzelmoleküle hergestellt, die Informationen über die Knoten und ihre möglichen Beziehungen repräsentieren. Jedem Knoten entspricht ein dreiteiliger Einzelstrang, wobei der linke Teil für hineingehende Kanten verwendbar ist, der rechte Teil für hinausgehende Kanten und der Mittelteil aus technischen Gründen gebraucht wird. Außerdem sind jedem Knoten weitere Moleküle zugeordnet, nämlich je ein Molekül für jede mögliche eingehende Kante, je eins für jede mögliche ausgehende Kante sowie je eins für jedes mögliche Paar ein- und ausgehender Kanten. Für diese drei Arten von Molekülen wird jeweils ein Reagenzglas deklariert, das im ersten Schritt mit seinem Molekül gefüllt wird, falls der Graph passende Kanten besitzt.

Außerdem gibt es Reagenzgläser, die dem Aufbau von Wegen von jedem Knoten zu jedem anderen dienen. Während des Aufbaus wird außerdem festgehalten, welcher Knoten als nächstes besucht werden soll. Der Wegebau beginnt im zweiten Schritt mit den Knotenmolekülen, die ausgehende Kanten repräsentieren. Sie repräsentieren so auch Wege von einem Knoten zu sich selbst der Länge 0 mit jeweils potentiellen Folgeknoten. In den nächsten Schritten werden jeweils die Moleküle zugegeben, die den potentiellen Folgeknoten repräsentieren und neue potentiellen Folgeknoten gemäß den ausgehenden Kanten vorsehen. Das wird so oft wiederholt, wie die zweite Eingabe vorgibt. Zum Schluss werden die Moleküle zugefügt, die die potentiellen Folgeknoten repräsentieren, aber keine ausgehenden Kanten mehr nennen. Um dann Moleküle zu haben, die die Wege von einem Knoten zu einem zweiten Knoten darstellen, werden die unvollständigen Moleküle entfernt.

In der Korrektheitsaussage wird schließlich formuliert, welche Moleküle in welchen Reagenzgläsern welchen Wegen entsprechen.

Wegebau

Eingabe: $G = ([n], E \subseteq [n] \times [n]), m \in \mathbb{N}$

Vorverarbeitung:

(1) *Einzelstränge herstellen:*
 l_i, b_i, r_i für $i \in [n]$, die alle gleich lang und paarweise verschieden sind.

(2) *Zusammensetzen:*

$$s_i = l_i b_i r_i \text{ für } i \in [n],$$

$$i - in - j - out - k = \frac{b_j}{r_i l_j} \left[\frac{b_j}{b_j} \right] r_j l_k \text{ für } i, j, k \in [n],$$

$$i - in - j = \frac{b_j}{r_i l_j} \left[\frac{b_j}{b_j} \frac{r_j}{r_j} \right] \text{ für } i, j \in [n],$$

$$j - out - k = \left[\frac{l_j}{l_j} \frac{b_j}{b_j} \right] r_j l_k \text{ für } j, k \in [n].$$

Reagenzgläser: $(IO_{ijk})_{i,j,k \in [n]}$, $(I_{ij})_{i,j \in [n]}$, $(O_{jk})_{j,k \in [n]}$, $(P_{ijk})_{i,j,k \in [n]}$, $(P_{ij})_{i,j \in [n]}$.

Verfahren:

- (1) $IO_{ijk} \leftarrow i - in - j - out - k$ falls $(i, j), (j, k) \in E$ für $i, j, k \in [n]$,
 $I_{ij} \leftarrow i - in - j$ falls $(i, j) \in E$ für $i, j \in [n]$,
 $O_{jk} \leftarrow j - out - k$ falls $(j, k) \in E$ für $j, k \in [n]$;
- (2) $P_{ij} \leftarrow O_{ij}$ für $i, j \in [n]$,
- (3) m -mal: $P_{ikl} \leftarrow \sum_{j=1}^n (P_{ijk} + IO_{jkl})$ für $i, k, l \in [n]$;
- (4) $P_{ik} \leftarrow \sum_{j=1}^n (P_{ijk} + I_{jk})$ für $i, k \in [n]$.
- (5) $P_{ik} \leftarrow \text{blunt}(P_{ik})$ für $i, k \in [n]$.

Korrektheit:

$i_1 \dots i_k$ mit $1 < k \leq m + 1$ ist genau dann ein Weg von i nach j in G , wenn

$$\begin{bmatrix} s_{i_1} \dots s_{i_k} \\ \overline{s_{i_1}} \dots \overline{s_{i_k}} \end{bmatrix} \in P_{ij}$$

Dabei ist $i_1 \dots i_k$ ein Weg von i nach j in G , wenn $i = i_1, j = i_k$ und für $l = 1, \dots, k - 1$ gilt: $(i_l, i_{l+1}) \in E$.

Um halbwegs komfortabel Algorithmen im Adleman-Lab modellieren zu können, wird nun noch zusätzlich erlaubt, Problemlösungen durch Import wiederzuverwenden. Das Prinzip wird am Beispiel des Hamilton-Wege-Problems beispielhaft vorgeführt, das **Wegebau** importiert. Dabei ist ein Hamiltonscher Weg einer, der jeden Knoten genau einmal besucht. Im vorliegenden Fall werden Anfangs- und Endknoten des Weges als Eingabe vorgegeben.

Hamiltonscher Weg

Eingabe: $G = ([n], E \subseteq [n] \times [n]), i_0, x_0 \in [n]$

Import: **Wegebau**

Verfahren:

- (1) **Wegebau**($G, n - 1$)
- (2) für $i = 1, \dots, n$: $P_{i_0j_0} \leftarrow \text{select}(s_i, P_{i_0j_0})$
- (3) $\text{empty?}(P_{i_0j_0})$

Korrektheit:

G enthält genau dann einen Hamiltonschen Weg von i_0 nach j_0 , wenn $P_{i_0j_0}$ nicht leer ist, d.h. wenn $\text{empty?}(P_{i_0j_0})$ gilt.

Der Test funktioniert, weil im ersten Verfahrensschritt Wege bis zur Länge $n - 1$ aufgebaut werden, so dass jeder Knoten höchstens einmal vorkommen kann. Im zweiten Verfahrensschritt werden dann die Moleküle selektiert, die Wege repräsentieren, bei denen mindestens alle Knoten einmal vorkommen. Insgesamt ist also das Reagenzglas $P_{i_0j_0}$ am Ende genau dann nicht leer, wenn jeder Knoten genau einmal vorkommt.

Anmerkung: Bezüglich der Repräsentation von DNA-Molekülen wird folgendes vereinbart:

- (1) Einzelstränge sind Wörter über dem Alphabet $\{A, C, G, T\}$.
- (2) Die komplementären Einzelstränge werden überstrichen.
- (3) Doppelstränge werden durch Übereinanderschreiben der beiden komplementären Einzelstränge dargestellt. Um das zu betonen, werden vollständige Doppelstränge in eckige Klammern gesetzt.
- (4) Neben den vollständigen Doppelsträngen (blunt) werden nur Moleküle betrachtet, die zusätzlich links oder rechts unten oder oben lose Enden besetzen. Sie werden an der entsprechenden Stelle vor oder hinter die eckigen Klammern geschrieben.