

Das Sticker Modell

Seminararbeit von Sven Gerding

Inhaltsverzeichnis

1	Einführung	1
2	Das Sticker Modell	2
2.1	Informationsdarstellung	2
2.2	Grundlegende Operationen	2
2.3	Initialisierung	3
2.4	Beispielproblem	5
3	Physikalische Implementierung	6
3.1	Implementierung der Operationen	6
3.2	Initialisierung und Ergebnisermittlung	7
3.3	Speicherstrang und Stickerdesign	8
4	Eine Stickermaschine	8
5	Zusammenfassung	10

1 Einführung

Die Forschung auf dem Gebiet des molekularen Rechnens, zu dem auch DNA-Computing zählt, ist motiviert durch den Wunsch, eines Tages Rechner konstruieren zu können, mit denen sich Probleme lösen lassen, die von den heutigen Rechnern nicht oder nur sehr schwer zu bearbeiten sind. Die herausragenden Merkmale eines DNA Computers sind seine hohe Speicherdichte und seine massive Parallelität. Es sei darauf hingewiesen, daß bis dato alle Arbeiten auf dem Gebiet des DNA-Comutings, ausgenommen des von Adleman in [2] vorgestellten Prototyps eines DNA-Computers, noch rein theoretischer Natur sind. Dazu zählt auch das in [4] vorgestellte Sticker Modell, auf dem diese Arbeit beruht. Dieses Modell zeichnet sich zum Einen dadurch aus, daß die hierbei verwendeten Materialien - DNA Stränge - wiederverwendbar sind und zum Anderen keine kurzlebigen Enzyme benötigt werden. Dadurch ergäbe sich eine im Vergleich zu anderen Modellen relativ kostengünstige Realisierung.

Zunächst soll verdeutlicht werden, wie Information im Sticker Modell dargestellt wird und welche Rechenoperationen realisiert sind. Weiterhin werden mögliche physikalische Implementierungen der Rechenoperationen vorgestellt. Abschließend soll kurz eine evtl. mögliche Realisierung des Modells durch eine Sticker Maschine erläutert werden.

2 Das Sticker Modell

2.1 Informationsdarstellung

Das Material zur Speicherung von Information ist, wie in anderen Modellen auch, DNA. Im Sticker Modell existieren zwei Gruppen einzelsträngiger DNA. Ein *Speicherstrang* (engl. *Memory Strand*) ist ein DNA Strang, welcher N Basen lang ist. Diese Folge von N Basen ist aufgeteilt in K Regionen, die jeweils M Basen lang sind. Jede dieser K Regionen stellt genau eine Bit-Position dar. Zusätzlich existieren K unterschiedliche DNA-Stränge, deren Länge ebenfalls M Basen beträgt und die zu genau einer der K Regionen des Speicherstrangs komplementär sind. Diese Art von DNA-Strängen heißen Sticker. Siehe dazu auch Abbildung 1.

Hat ein Sticker an die zu ihm komplementäre Region des Speicherstrangs gebunden, so gilt das entsprechende Bit als gesetzt (1), ist die Region frei, entspricht das einem nicht gesetzten Bit (0). Speicherstränge stellen also Bit-Strings dar. Solche Bit-Strings werden auch *Speicherkomplexe* (engl. *Memory Complexes*) genannt. Eine Menge von Speicherkomplexen bezeichnet man als Tube. Zu beachten ist dabei, daß die Speicherstränge einer Tube immer identisch sind, also die gleiche Länge und Basensequenz aufweisen. Sie unterscheiden sich nur durch die an sie gebundenen Sticker.

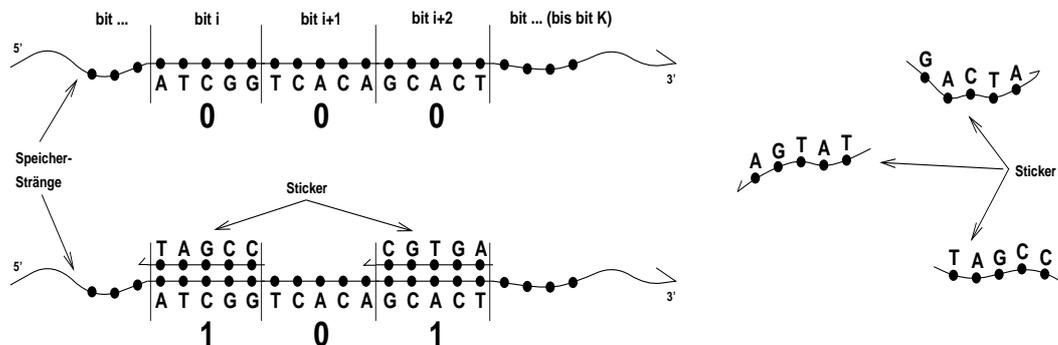


Abbildung 1: Speicherstränge und Sticker [4].

2.2 Grundlegende Operationen

Die vier grundlegenden Operationen auf Mengen von Bit-Strings sind die *Kombination* von zwei Mengen zu einer neuen Menge, die *Teilung (Trennung)* einer Menge in zwei neue Mengen und das *Setzen und Löschen* des k -ten Bits jedes Strings in einer Menge.

- Die einfachste Operation ist die Kombination zweier Mengen von Bit-Strings zu einer neuen. Logisch ist das die Multimengen-Vereinigung aller Strings der beiden Ausgangsmengen. Auf physikalischer Ebene entspricht das der Erzeugung einer neuen Tube, die alle Speicherkomplexe beider Ausgangs-Tubes enthält.
- Die Teilung einer Menge geschieht, indem zwei neue Mengen erzeugt werden, wobei eine Menge alle Bit-Strings enthält, bei denen ein bestimmtes Bit gesetzt ist und die Andere diejenigen Strings, bei denen dieses Bit gelöscht ist. Auf physikalischer Ebene bedeutet dies die Isolierung aller Komplexe, bei denen ein Sticker an eine bestimmte Region gebunden ist.
- Ein Bit wird in jedem String einer Menge gesetzt, indem der jeweilige Sticker für dieses Bit an alle Speicherkomplexe einer Tube gebunden wird.
- Um ein Bit in jedem String einer Menge zu löschen, muß der jeweilige Sticker von jedem Speicherkomplex in der Tube entfernt werden. Wie sich später noch herausstellen wird, ist diese Operation in einer physikalischen Implementierung des Modells am schwierigsten zu realisieren.

Abbildung 2 zeigt die Vorgänge der vier Operationen.

Die Arbeitsweise des Sticker-Modells besteht also aus einer Folge der vier Operationen Kombination, Teilung und Bit-löschen, bzw. -setzen.

2.3 Initialisierung

Ausgangspunkt aller Berechnungen im Sticker Modell ist eine Menge von Bit-Strings. Das Ergebnis ist eine oder mehrere Mengen von Strings, die jedoch auch leer sein können (z.B. bei Entscheidungsproblemen).

- Eine (K,L) Bibliothek enthält Bit-Strings der Länge K . Die Speicherkomplexe bilden die Menge aller möglichen Bit-Strings der Länge L , gefolgt von $K-L$ Nullen. Die Anzahl der Elemente in der Menge beträgt demnach 2^L . Zum Beispiel bildet eine $(5,2)$ Bibliothek die Strings: 00000, 01000, 10000, 11000.
- Die ersten L Bits codieren die Eingabe, während die folgenden $K-L$ Bits als Speicher dienen und die Ausgabe codieren.

Die Startmenge wird gebildet, indem alle möglichen Eingaben der Länge L erzeugt werden und der Arbeitsbereich $K-L$ gelöscht wird.

Das Ergebnis einer Berechnung kann durch Untersuchung der Speicherkomplexe der Ergebnis-Tube(s) auf gebundene Sticker ermittelt werden. Gegebenenfalls muß nur bestimmt werden, ob eine Ergebnis-Tube überhaupt Sticker enthält.

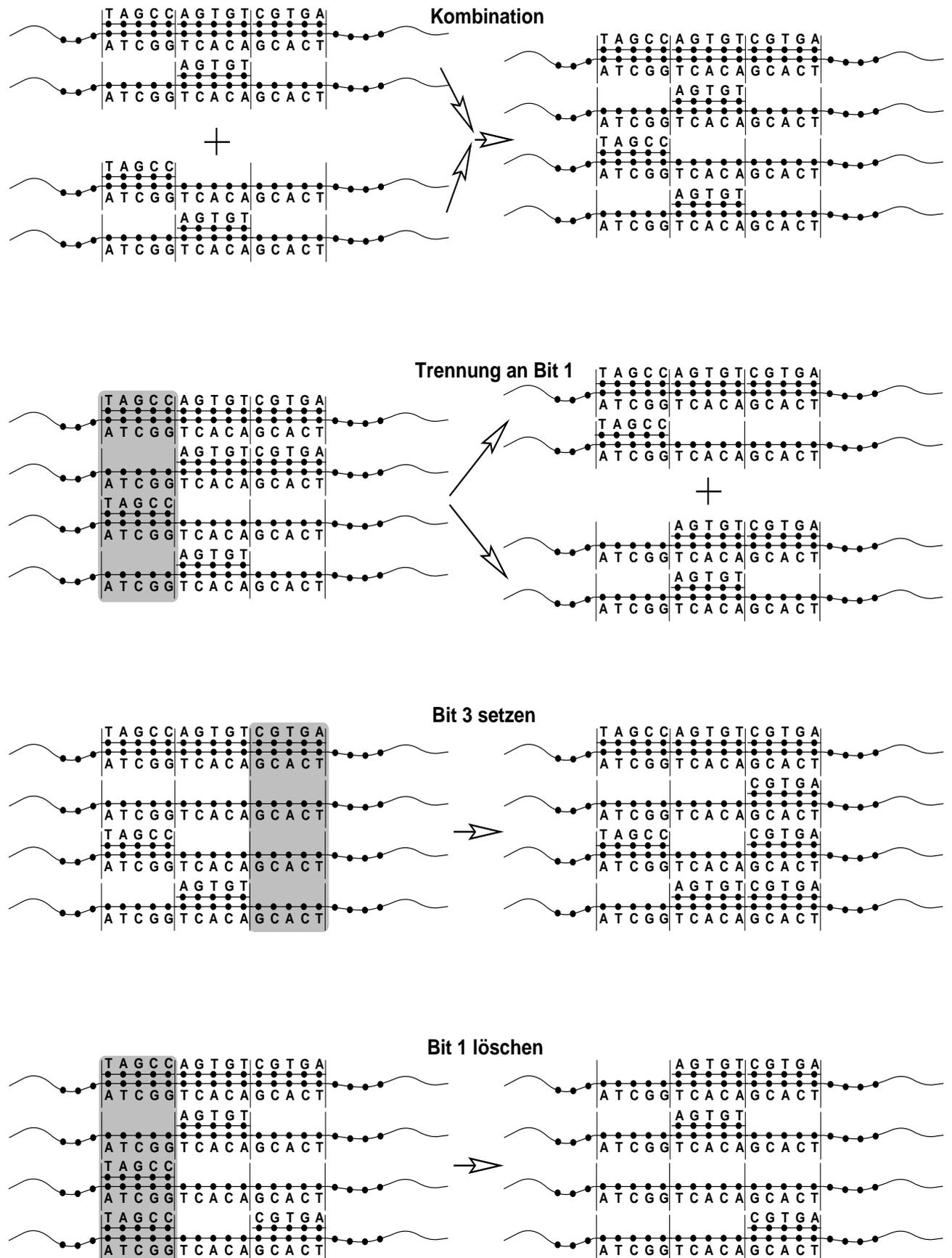


Abbildung 2: Die vier Operationen Kombination, Trennung, Bit-löschen und Bit-setzen [4].

2.4 Beispielproblem

Anhand eines Beispielproblems soll nun das Zusammenwirken der einzelnen Operationen verdeutlicht werden.

- Gegeben sei eine Multimenge $M = \{M_1, \dots, M_B\}$ und eine Menge $X = \{1, \dots, A\}$ mit $M_i \subseteq X$. Was ist die kleinste Teilmenge I von $\{1, \dots, B\}$, so daß $\bigcup_{i \in I} M_i = X$?
- Vorgehen:
 - Erzeuge 2^B Speicherkomplexe der Länge $K = B + A$, die alle 2^B möglichen Teilmengen von M bilden. Bits $1, \dots, B$ repräsentieren Elemente aus M . Bits $B + 1, \dots, B + A$ repräsentieren Objekte aus X .
 - Setze für jede Menge M_i die Objekte, die sie enthält. Bsp.: Für $M_1 = \{1, 4\}$ werden in allen Bit-Strings, die M_1 enthalten, die Bits 1 und 4 im Arbeitsbereich gesetzt. *xx1.1xx1* wäre ein solcher Bit-String. (Hier sind zur besseren Lesbarkeit Eingabe- und Arbeitsbereich durch einen Punkt voneinander getrennt.)
 - Sortiere diejenigen Speicherkomplexe aus, die alle Bits $B+1, \dots, B+A$ gesetzt haben.
 - Ermittle die Speicherkomplexe, die die wenigsten Elemente aus M beinhalten.

Der Algorithmus für dieses Problem stellt sich wie folgt dar:

1. /* Trage in die Positionen B+1 bis B+A jedes Komplexes ein, */
 /* welche Elemente es enthält */
 FOR i = 1 TO B
 Separate T_0 INTO T_{on} AND T_{off} BASED ON BIT i
 FOR J = 1 TO $|M_i|$
 Set BIT B + $M_i[j]$ IN T_{on}
 COMBINE T_{on} AND T_{off} INTO T_0
2. /* Entferne alle Komplexe, die nicht alle A Elemente beinhalten */
 FOR i = B + 1 TO B + A
 Separate T_0 INTO T_0 AND T_{bad} BASED ON BIT i
 Discard T_{bad}
3. /* Zähle, wieviel Elemente benutzt werden. Am Ende enthält */
 /* Tube T_i alle Komplexe, die genau i Elemente beinhalten */
 FOR I = 0 TO B - 1

```

FOR j = 1 DOWNT0 0
  Separate  $T_j$  INTO  $T_{(j+1)'}$  and  $T_j$  BASED ON BIT  $i+1$ 
  Combine  $T_{j+1}$  AND  $T_{(j+1)'}$  INTO  $T_{j+1}$ 

```

3 Physikalische Implementierung

In diesem Abschnitt werden mögliche physikalische Implementierungen der in Abschnitt 1 beschriebenen logischen Operationen *Kombination*, *Trennung* und *Bit setzen/löschen* vorgestellt. Es sei nochmals darauf hingewiesen, daß alle hier vorgestellten Überlegungen rein theoretisch sind und sich erst in praktischen Experimenten herausstellen kann, ob die dargestellten Ansätze so realisierbar sind, oder ob andere Implementierungen zu bestimmten Operationen gefunden werden müssen.

3.1 Implementierung der Operationen

- **Kombination**

Auf den ersten Blick scheint diese Operation leicht implementierbar zu sein: Der Inhalt zweier Tubes wird einfach zusammengeschüttet. Jedoch können lange DNA-Stränge durch die beim Mischen, bzw. Pumpen entstehenden Kräfte zerbrechen. Auch muß beachtet werden, daß DNA-Stränge an Schläuchen, Gefäßwänden, etc. haften bleiben können und somit nicht in der Zieltube enthalten sind. Hieraus ergibt sich ein großer Nachteil, da im Sticker-Modell nur mit wenigen Kopien jedes Speicherstrangs gearbeitet wird.

- **Trennung**

Es müssen die Komplexe voneinander getrennt werden, bei denen ein Sticker an eine bestimmte Region gebunden hat. Eine mögliche Lösung sieht vor, daß man Einzelsträngige DNA-Sonden in die Tube gibt, wobei die Sonden an die Region der Speicherkomplexe binden, an der die Trennung erfolgen soll. Die Trennung kann dann auf physikalischem Weg stattfinden. So können die DNA-Sonden an irgendein Substrat gebunden sein. Speicherkomplexe, bei denen die entsprechende Region frei ist, binden an die Sonden, die übrigen Komplexe können abgefiltert werden. Anschließend müssen die DNA-Sonden wieder von den daran gebundenen Komplexen getrennt werden. Die Bindungs-Affinität der Sonden muß geringer sein als die der Sticker, so daß beim Abtrennen z.B. durch Erhitzen nicht auch Sticker gelöst werden.

- **Bits setzen**

Das Setzen von Bits ist trivial. Es wird einfach eine größere Menge von entsprechenden Stickern in die Tube gegeben. Diese Sticker binden dann an freie, zu ihnen komplementäre Regionen. Überschüssige Sticker werden anschließend durch Filtration entfernt.

- **Bits löschen**

Beim Löschen besteht die Schwierigkeit, daß nur eine bestimmte Region von Stickern befreit werden muß. Erhitzung, um die Sticker zu lösen kommt also nicht in Frage, da das zu einer Beeinflussung aller Sticker führt. Ein eventuell möglicher Lösungsansatz besteht hierbei in *PNA Strang Invasion* durch Triple-Helix-Formation [3]. PNA-Stränge, die zu dem zu lösenden Sticker komplementär sind, werden in die Tube gegeben. Durch Erhitzung bis zu einem gewissen Punkt binden diese PNA-Stränge an die Sticker und bilden eine Triple-Strang, wodurch der Sticker vom Speicherstrang gelöst wird. Das Löschen von Bits ist die Problematische der vier Operationen, und es liegen bis dato noch keinerlei experimentelle Erfahrungen vor.

3.2 Initialisierung und Ergebnisermittlung

Zur Erstellung einer (K,L) Bibliothek müssen zunächst 2^L identische Kopien eines Speicherstrangs synthetisiert werden. Anschließend werden die Sticker nach folgenden Schema aufgebracht:

- Die Speicherstränge werden in zwei gleichgroße Mengen aufgeteilt.
- Zu einer der beiden Mengen werden Sticker für den Bereich 1-L zugefügt, so daß alle Bits 1-L auf allen Strängen der Menge gesetzt sind. Überschüssige Sticker werden abgefiltert.
- Die beiden Mengen werden vereinigt und erhitzt, was bewirkt, daß sich alle Sticker lösen.
- Nach dem Abkühlen binden die Sticker wieder an die Speicherstränge. Für je zwei Stränge existiert nur ein Sticker, d.h. die Wahrscheinlichkeit, daß ein bestimmtes Bit eines Speicherstranges gesetzt ist, beträgt $1/2$. Jeder mögliche String wird mindestens einmal erzeugt mit der Wahrscheinlichkeit 63%. Diese Wahrscheinlichkeit läßt sich noch erhöhen, indem mehr als 2^L Speicherstränge erzeugt werden.

Um das Ergebnis einer Berechnung zu ermitteln, muß zunächst geprüft werden, ob sich überhaupt Speicherkomplexe in der Ergebnistube befinden. Ein solcher Test läßt sich auf verschiedene Art und Weise durchführen. Eine Möglichkeit wäre, die Speicherstränge fluoreszierend zu markieren. Des weiteren könnten Speicherstränge durch Filtration detektiert werden.

Zur Untersuchung von Speichersträngen auf gebundene Sticker wurde vorgeschlagen, die einzelnen Stränge elektronenmikroskopisch zu untersuchen. Es ist jedoch fraglich, ob solch ein Vorgehen bei einer größeren Ergebnismenge praktikabel ist. Eine anderer Ansatz besteht darin, eine PCR basierte Technik anzuwenden.

Es ist jedoch auch möglich nur auf Detektion von Speicherkomplexen zurückzugreifen, um einen Ergebnisstring zu erhalten. Hierbei bedient man sich der Binärbaum-Decodierung:

1. Ausgangspunkt ist die Menge der Ergebnisstrings, also die Tube mit den Ergebnis-Komplexen.
Ermittle Speicherkomplexe darin.
2. Ist kein Speicherkomplex vorhanden, ist kein Ergebnis gefunden worden
3. Sind Speicherkomplexe gefunden worden, Trennung am ersten Bit des Ergebnisstrings.
4. Ermittle Speicherkomplexe in jeder der so entstandenen Mengen und behalte diejenigen, die nicht leer sind.
5. Wiederhole die Teilung und Ermittlung für jedes Bit des Ergebnisstrings.

3.3 Speicherstrang und Stickerdesign

Die wichtigste Anforderung beim Design von Speichersträngen ist die Stickerspezifität. Sticker dürfen nur an den für sie bestimmten Regionen binden und nirgendwo sonst. Die Basenfolge einer Region darf somit nirgendwo sonst in der Basensequenz des Speicherstranges vorkommen.

Mathematisch bedeutet das den Entwurf einer Sequenz der Länge N (Speicherstrang), so daß K nicht überlappende Teilsequenzen der Länge M (Regionen) existieren, mit der Eigenschaft, daß das Komplement (Sticker) jeder Region mindestens D_1 Differenzen mit jeder anderen Teilsequenz der Länge M in der kompletten Sequenz besitzt. Der Wert D_1 ist die kleinste Anzahl von Differenzen, die nötig sind, damit ein M Basen langer Sticker nicht an eine Sequenz bindet.

Sequenzen auf einem Strang sollten nicht komplementär zueinander sein, um zu verhindern, daß der Strang, sollte er sich verknäulen, an sich selbst bindet. Vermeiden ließe sich dieser Effekt, indem man verlangt, daß für den Speicherstrang nur Purinbasen (Adenin, Guanin) und für Sticker ausschließlich Pyrimidinbasen (Thymin, Cytosin) verwendet werden, bzw. umgekehrt.

Trennungs-Sonden müssen eine geringere Affinität als Sticker besitzen, damit sie später leichter zu entfernen sind und die Sticker bei dieser Prozedur nicht beeinträchtigt werden. Sie sollten mindestens D_2 Differenzen mit ihrer Region besitzen und $D_3 > D_2$ Differenzen überall sonst.

4 Eine Stickermaschine

Basierend auf den in den vorangegangenen Abschnitten vorgestellten Ideen soll hier eine beispielhafte Implementierung des Sticker Modells als Sticker Maschine erläutert werden.

Ein solcher Rechner besteht im Wesentlichen aus einem Steuerrechner, einem Roboter, Pumpen und Heiz/Kühlelemente und einer Menge von Daten- und Operator-Tubes. Der Steuerrechner ist verantwortlich für die Compilierung eines in einer höheren Sprache gegebenen Algorithmus in eine Folge von Operationen (Kombination, Teilung, Bit setzen/löschen) und somit die Ansteuerung des Roboters und der Pumpen und Heiz/Kühlelemente.

Daten-Tubes bestehen aus einem geschlossenen Zylinder an dessen Enden sich jeweils ein Anschluß befindet, so daß der Durchfluß von Lösung möglich ist. Zusätzlich existiert in jedem Zylinder an einem Ende eine Membran, die nur Lösung, jedoch keine Speicherstränge der Sticker durchläßt. Der Anschluß an dem Ende des Zylinders, an dem auch die Membran liegt nennt sich Clean-Side, der andere heißt Dirty-Side. Daten-Tubes beinhalten, sofern sie nicht leer sind, entweder Speicherkomplexe oder ungebundene Sticker, für jeweils eine Region.

Operator-Tubes sind ähnlich im Aufbau wie die Daten-Tubes, doch unterscheidet man hier nicht zwischen Clean- und Dirty-Side. (Tatsächlich sind beide Anschlüsse Dirty). Die Operator-Tubes werden in Trennungs-, Sticker- und Leer-Tubes unterschieden.

Eine Leer-Tube ist einfach ein leerer Zylinder mit zwei Anschlüssen. Sticker-Tubes besitzen darüber hinaus noch eine Membran, die zwar Sticker, aber nicht Speicherstränge hindurchläßt. Trennungs-Tubes enthalten DNA-Sonden, die so an den Zylinder gebunden sind, daß sie diesen nicht verlassen können. Für jedes Bit existiert mindestens eine Trennungs-Tube. Eine anschauliche Darstellung der verschiedenen Tubes findet sich in Abbildung 3.

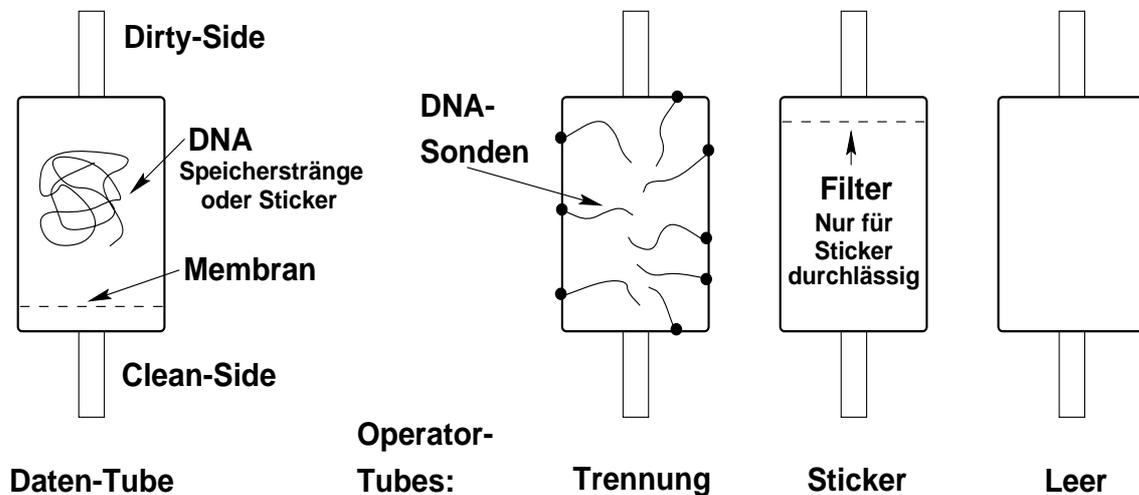


Abbildung 3: Daten- und Operator-Tubes [4].

Abbildung 4 zeigt einen beispielhaften Aufbau für eine Operation, deren Ablauf sich wie folgt gliedert:

- Der Roboter wird vom Rechner angesteuert und entnimmt einem Rack zwei Daten- und eine Operator-Tube.

- Die Dirty-Sides der Daten-Tubes werden mit der Operator-Tube verbunden, während die Clean-Sides der Daten-Tubes über eine Pumpe und Heiz/Kühlaggregat zusammengeführt werden.
- Durch die drei Tubes wird nun Lösung in verschiedene Richtungen gepumpt, entsprechend der auszuführenden Operation. Dauer und Richtung des Pumpvorgangs und die Tempertur der Lösung wird dabei vom Steuerrechner kontrolliert.
- Nach der Operation werden die einzelnen Tubes wieder durch den Roboter in das Rack zurückgelegt oder ausgesondert.

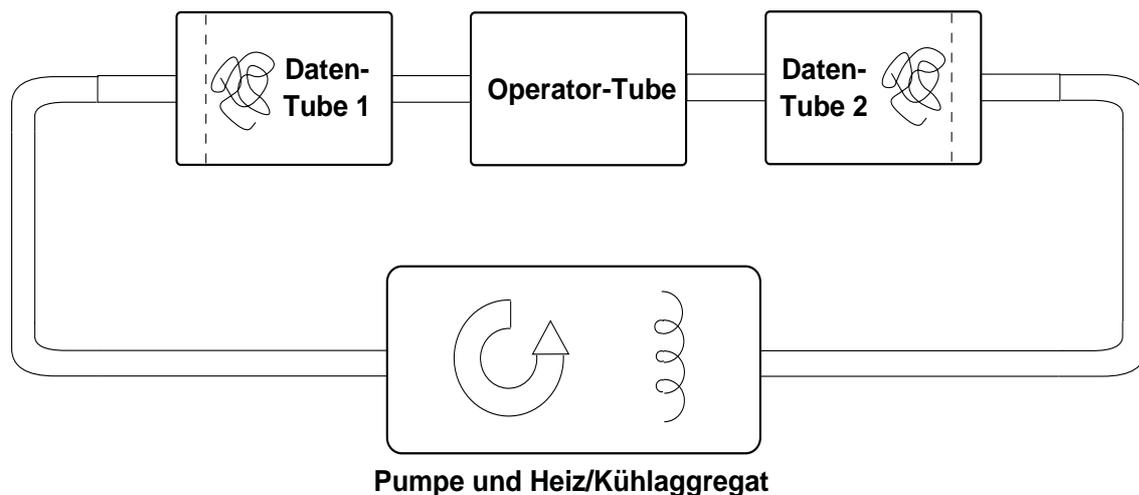


Abbildung 4: Aufbau für eine Operation [4].

Es zeigt sich das die Materialien, die zur Realisierung einer Sticker-Maschine nötig sind, relativ einfach aufgebaut sind und bereits heutzutage verfügbar sind. Zur Dimensionierung läßt sich sagen, daß der Platzbedarf, da eine Tube nur wenige ml Lösung enthält und wiederverwendbar ist, recht moderat und nicht viel größer als ein Laborarbeitsplatz sein dürfte.

5 Zusammenfassung

Im Folgenden seien nochmal die wesentlichen Punkte des Sticker Modells aufgeführt:

- Die massive Parallelität von DNA-Computing erlaubt die Bearbeitung komplexer Probleme, die mit heutigen Rechnern nicht, oder nur sehr schwer zu lösen sind.
- Das Sticker-Modell ist nicht problemspezifisch, sondern läßt sich universell für eine große Klassen von Problemen einsetzen.

- Die Umsetzung einfacher logischer Operationen kann sehr problematisch und aufwendig sein.
- Alle Arbeiten sind bis heute noch rein theoretisch. Eine erfolgreiche Implementierung des Modells ist daher abhängig von der weiteren Entwicklung in der Biotechnologie. Insbesondere sind hier Fehlerraten beim Arbeiten mit DNA zu reduzieren, die zwar in der Molekularbiologie hervorragend, für die Informationsverarbeitung nicht akzeptabel sind.

Literatur

- [1] Leonard Adleman. On constructing a molecular computer. Draft Jan. 11, 1995.
- [2] Leonard Adleman. Molecular computation of solutions to combinatorial problems. *Science* 266, pages 1021–1024, Nov. 11 1994.
- [3] Michael Egholm, Leif Christensen, Kim L. Dueholm, Ole Buchardt, James Coull, and Peter E. Nielsen. Efficient ph-independent sequence-specific dna binding by pseudoisocytosinecontaining bis-pna. *Nucleic Acids Research*, 23(2):217–222, 1995.
- [4] Sam Roweis, Erik Winfree, Richard Burgoyne, Nickolas V. Chelyapov, Myron F. Goodman, Paul W. K. Rothmund, and Leonard M. Adleman. A sticker based architecture for dna computation. In *Proceedings of the Second Annual Meeting on DNA Based Computers, held at Princeton University, June 10-12, 1996.*, DIMACS: Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, May 1996.

Weitere Informationen und Literatur zum Thema *DNA-Computing* sind unter <http://www.wi.leidenuniv.nl/~jdassen/dna.html> erhältlich.