

1. Das Spielprinzip

Gegeben sind 4 Stapel a, b, c und d. Diese sollen auf eine vorgegebene Höhe gebracht werden. Ursprünglich gab es hierfür 8 Schalter: für jeden der 4 Stapel einen Schalter zum Wegnehmen und einen zum Hinzufügen von Kisten. Da es aber nach einem Blitzeinschlag nur noch 3 funktionierende Schalter gibt, mussten diese neu verdrahtet werden, damit das Umladen weiterhin möglich ist. Diese 3 Schalter A, B und C wurden jetzt so verdrahtet, dass sie entweder jeweils von mehreren Stapeln eine Kiste entfernen (\downarrow) oder eine neue Kiste hinzufügen (\uparrow). Die betroffenen Stapel (a, b, c, d) werden rechts neben den Schaltern (Verdrahtung) angezeigt.

Ziel: Bringe alle Stapel mit Hilfe der Schalter auf die gewünschte Höhe (rote Linie)!

2. Die Bedienung

Mit „**neu**“ wird eine neue Aufgabe gestartet.

Bei \rightarrow kann die Anzahl der Stapel geändert werden

Bei \downarrow kann die Anzahl der Schalter geändert werden.

Beachte: die Anzahl der Schalter kann maximal so groß wie die Anzahl der Stapel werden!

Mit „**nochmal**“ kann die letzte Aufgabe nochmal gestartet werden.

Mit „**KI fragen**“ wird eine Lösung der Aufgabe berechnet.

Die Lösung wird so ausgegeben: <Anzahl><Schalter> | ...

zB. bedeutet:

1A | 2B | 1C : einmal Schalter „A“, zweimal Schalter „B“ und einmal Schalter „C“.

Mit „**A**“, „**B**“ und „**C**“ können die angegebenen Stapel entsprechend der Verdrahtung geändert werden. Eine Verdrahtung beginnt mit \downarrow oder \uparrow , gefolgt von den Stapeln, welche damit geändert werden (zB: \uparrow a,-,c,d = erhöhe die Stapel a, c und d um jeweils eine Kiste).

3. Die Spielstrategie

Verwende die Schalter mit Bedacht: die Änderungen sind irreversibel. Nicht immer lassen sich alle Stapel in beide Richtungen (hoch und herunter) verändern.

Die Reihenfolge, in welcher die Schalter betätigt werden, ist egal (Kommutativgesetz).

Achte auf Alleinstellungsmerkmale:

- mehrere Schalter mit der gleichen Verdrahtungen und Richtung können auf einen Schalter reduziert werden
- kann ein Stapel mit nur einem Schalter in nur eine (passende) Richtung geändert werden, so kann dieser Schalter bis zur Sollhöhe des Stapels betätigt und dann ignoriert werden
- kann ein Stapel mit mehreren Schaltern nur in eine Richtung geändert werden, so ist die Summe der Betätigungen der betreffenden Schalter gleich der Differenz des Stapels zur Sollhöhe
Vorsicht: wenn in diesem Fall die Sollhöhe nach oben bzw. unten überschritten wird, hilft nur noch ein Neustart der Aufgabe mit „**nochmal**“
- wenn es zwei Verdrahtungen gibt, welche den gleichen Stapel einmal nach oben und einmal nach unten enthalten, so können diese beiden dazugehörigen Schalter so zusammengefasst werden, dass dabei dieser eine Stapel entfällt (Superschalter)
- auch die Superschalter können eventuell wieder zusammengefasst werden (siehe Beispiel)
- gibt es 3 oder mehr Stapel mit der gleichen Höhe und gibt es dazu einen Schalter, welcher genau diese Stapel in der passenden Richtung verändert, dann ist dieser Schalter vermutlich ein Teil der Lösung und kann eventuell bis zum Ausgleich auf die Sollhöhe betätigt werden

Nutze diese Merkmale sinnvoll, um weitere Einschränkungen der Lösung zu erreichen.

Beispiel für die Methode Superschalter:

Mit „neu“ und gleichzeitig <Control>-Taste kann ein Aufgabe mit einem Code gestartet werden, siehe dazu auch Kapitel 5.

Die Aufgabe mit dem Aufgabencode: **8,4,8,4,6,8,-*.**,+.***,+***.**

hat diese Verdrahtung:

A: ↓a,-,c,d

B: ↑-,b,c,d

C: ↑a,b,c,-

und diese Spalten:

a ##### |

b ##### |

c ##### |

d ##### |

Ideal wäre jetzt ein Superschalter X: ↑4b, ↑2c, der nach einmal Betätigen zur Lösung gelangt.

Wir suchen jetzt nach Schaltern, die mit einer Spalte in gegensätzliche Richtungen verknüpft

sind und fassen diese zu Superschaltern (nacheinander betätigte Schalter) zusammen:

Wegen Spalte „c“ und „d“ kann A und B zu AB zusammengefasst werden.

Wegen Spalte „c“ kann A und C zu AC zusammengefasst werden.

Der Superschalter BC sieht so aus:

AB: ↓a, ↑b

AC: ↑b, ↓d

BC: ↑a,2↑b,2↑c,↑d

Wegen Spalte „d“ können auch die Superschalter AC und BC zusammengefasst werden:

ACBC: ↑a,3↑b,2↑c

Der Superschalter ACBC kommt der Lösung X: ↑4b, ↑2c schon recht nahe;

nur Stapel „a“ und „b“ stimmen noch nicht:

a ##### | #

b ##### |

c ##### |

d ##### |

Aber der Superschalter AB korrigiert genau diese beiden Stapel „a“ und „b“.

Die Lösung ist also: ACBC + AB = **2A 2B 2C**

a ##### |

b ##### |

c ##### |

d ##### |

Diese Methode entspricht in etwa dem Verfahren „Karnaugh-Veitch-Diagramm“ (KVD) zur Minimierung boolescher Ausdrücke.

4. Theorie (mathematische Ermittlung einer Lösung)

Mit der Hilfe eines Beispiels soll ein Weg zu einer Lösung gezeigt werden:

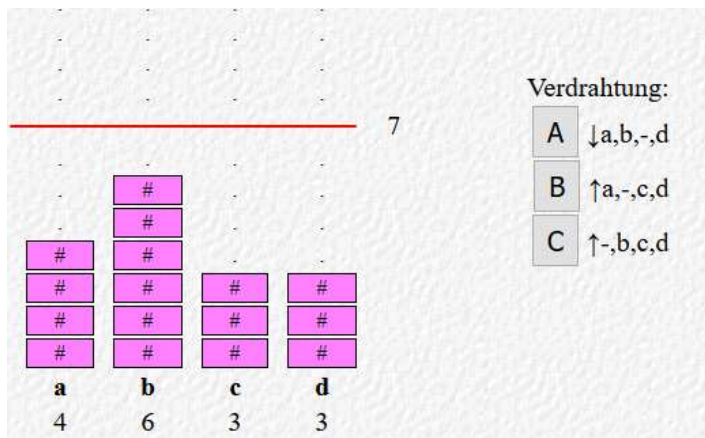


Abb. 1): Beispiel 1

Anhand der Verdrahtung kann ein lineares Gleichungssystem aufgestellt werden.

Hat dieses Gleichungssystem eine Lösung ohne negative Zahlen, gibt es eine Lösung für die gestellte Aufgabe.

Für jeden Stapel wird eine Gleichung aufgestellt. Dabei berücksichtigt, welche Schalter welche Aktionen hervorrufen. Die Gleichung muss die zur vorgegebenen Höhe noch fehlenden Kisten ergeben, also die Differenz der Stapelhöhen SOLL – IST.

Zum Beispiel ergeben sich für diese

Verdrahtung: (siehe Abb. 1)

A: ↓a,b,-,d

B: ↑a,-,c,d

C: ↑-,b,c,d

die folgenden Gleichungen:

Stapel a: Schalter A verringert die Höhe um 1, Schalter B erhöht die Höhe um 1 und Schalter C hat keinen Einfluss, die Schalter zusammengerechnet sollen SOLL – IST(a)

Kistenbewegungen ergeben:

$$a: -1A + 1B + 0C = \text{SOLL} - \text{IST}(a)$$

Analog gilt für die anderen Spalten:

$$b: -1A + 0B + 1C = \text{SOLL} - \text{IST}(b)$$

$$c: 0A + 1B + 1C = \text{SOLL} - \text{IST}(c)$$

$$d: -1A + 1B + 1C = \text{SOLL} - \text{IST}(d)$$

Im Beispiel ist SOLL = 7, IST(a...d) = 4,6,3,3

Daraus ergeben sich diese 4 Gleichungen:

$$1): -A + B = 3$$

$$2): -A + C = 1$$

$$3): B + C = 4$$

$$4): -A + B + C = 4$$

Die Lösung dieses Gleichungssystems lautet:

$$A=0, B=3, C=1$$

das bedeutet: keine negativen Lösungen: 3*Schalter B und 1*Schalter C führen zur Lösung!

5. Expertenmodus

Die Aufgabe kann als Code ausgegeben und (später) wieder gesetzt werden. Dadurch können besondere Aufgaben mit anderen geteilt und/oder archiviert werden. Der Code wird unten ausgegeben und kann dort für eine spätere Verwendung markiert und kopiert werden.

Diese Funktionen stehen zur Verfügung:

- a) Wird beim Drücken des Buttons „**nochmal**“ gleichzeitig die <Control>-Taste gedrückt:
 - Ausgabe des Aufgabencodes unten
- b) Wird beim Drücken des Buttons „**neu**“ gleichzeitig die <Control>-Taste gedrückt:
 - Abfrage eines Codes (aus a)) und Start mit dieser Aufgabe
- c) Wird beim Drücken des Buttons „**KI fragen**“ gleichzeitig die <Control>-Taste gedrückt:
 - Ausgabe der Gleichungen für die Schalter

einige besondere Codebeispiele (für „neu“ und <Control>):

7,4,4,6,3,3,-**.*,+**.*,+**.*	Beispiel 1 aus vorherigem Kapitel
7,4,10,5,7,4,-**.*,+**.*,-**.*	Schalter C bewegt nur 2 Stapel
7,4,5,4,7,6,+*...+.*...*.,+...*	alle Schalter bewegen je nur 1 Stapel
4,5,6,5,3,5,5,+**.*,+**.*,-**.*	5 Stapel und 3 Schalter
7,5,9,9,8,6,7,+**.*,-**.*,-**.*,+**.*	5 Stapel und 4 Schalter
4,5,2,4,3,3,4,+**.*,+**.*,-**.*,+**.*,-**.*	5 Stapel und 5 Schalter

6. Erzeugung einer lösbaren Aufgabe mit einer vorgegebenen Verdrahtung

Zum Beispiel diese Wunschverdrahtung:

A: ↑a,b,-,-

B: ↓-,b,c,-

C: ↑-,c,d

Schritt 1: Die Wunschverdrahtung in die Codeform A,B,C übersetzen (am besten im Editor):

+**..,-**.*,+**.* (+: nach oben, -:nach unten, *:Spalte verbunden, ..:Spalte nicht verbunden)

Schritt 2: Die Wunschverdrahtung invertieren (+ zu -, - zu +):

-**..,+**.*,-**.*

Schritt 3: die Wunschhöhe (zB. 7) und Spaltenanzahl (zB. 4) in die Codeform übersetzen:

7,4,7,7,7,7 (erste Zahl: Sollhöhe, zweite Zahl: Anzahl der Spalten, 3-...: Höhe der Spalten=Soll)

Schritt 4: Code aus Schritt 3 und 2 mit , zusammenfügen:

7,4,7,7,7,7,-..,+**.*,-**.***

Schritt 5: den Code aus Schritt 4 im Spiel mit „neu“ und <Control> eingeben

Schritt 6: beliebig die Schalter A, B und C betätigen

Schritt 7: den aktuellen Code mit „nochmal“ und <Control> ausgeben lassen:

zB.: **7,4,6,8,6,4,-**..,+**.*,-**.***

Schritt 8: den ausgegebenen Code kopieren und im Editor einfügen

Schritt 9: im Editor den Schaltercode (nach den Zahlen) durch die Wunschverdrahtung aus Schritt

1 ersetzen: **7,4,6,8,6,4,+**..,-**.*,+**.***

Schritt 10: Der Code im Editor ist jetzt der gewünschte Code und kann mit „neu“ und <Control> im Spiel ausprobiert werden. Die Lösung entspricht den Aktionen im Schritt 6.

Die KI findet manchmal eine andere Lösung (kürzer falls möglich)

Fertig !

Durch die invertierte Verdrahtung und den Start bei der Lösung wird die Aufgabe durch Rückwärtsrechnen erzeugt.

7. Ausblick

Das Spiel „Kabelsalat“ kann gelöst werden durch:

- probieren (mit etwas Übung geht das ganz gut)
- Aufstellen und Berechnung eines linearen Gleichungssystems (wie in Kapitel 4.)
- Constraint Programming: die Bedingungen erhält man, indem für die einzelnen Spalten entsprechend der Verdrahtung Gleichungen erstellt werden: Schalter müssen addiert den Unterschied von SOLL-IST ergeben. Damit kann dann nach einer Lösung gesucht werden.
- Backtracking: Erzeugung eines Spielbaums (Graph), der alle möglichen Positionen enthält und diese entsprechend verlinkt sind, dann Suche nach einem Pfad zur Lösung.
- Reduktionsverfahren (Superschalter) wie in Kapitel 3.
- graphische Lösung: Die 4 Stapel können als Koordinaten eines 4-dim. Raumes aufgefasst werden. Die Ausgangslage ist ein Punkt in diesem System; die Lösung ist ein zweiter Punkt. Die Schalter können als Vektoren interpretiert werden. Gesucht ist ein Weg von der Ausgangslage zum Lösungspunkt, der nur die vorhandenen Vektoren (Schalter) enthält.
- Rückwärts Rechnen: Beginnend bei der Lösung wird mit invertierten Schaltern versucht, die Ausgangslage zu finden. Findet eventuell kürzere Lösungen.