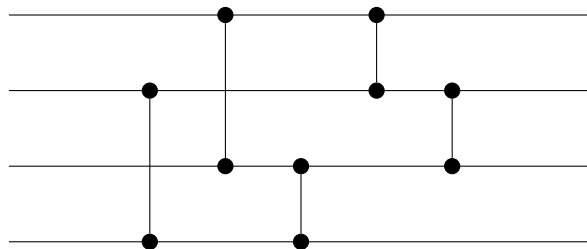


Übungen zur Vorlesung Formale Spezifikation und Verifikation

Blatt 8

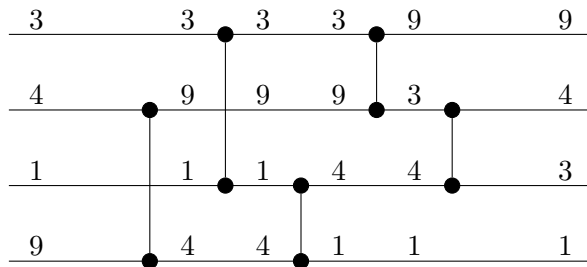
Aufgabe 8-1 Ein Sortiernetzwerk ist eine Beschreibung eines Sortieralgorithmus, der eine feste Anzahl von Werten sortieren kann. Sortiernetzwerke werden typischerweise verwendet, wenn man eine Sortierfunktion fest verdrahtet auf einem Chip realisieren möchte.

Das folgende Bild stellt ein Sortiernetzwerk für vier Werte dar:



Die vier Eingaben werden gleichzeitig links an die vier (horizontalen) Leitungen angelegt und werden gleichzeitig von links nach rechts durchgereicht. Die vertikalen Verbindungen einiger Leitungen stellen sogenannte Komparatoren dar. Erreichen zwei Werte die beiden Endpunkte eines Komparators, so werden sie vertauscht nach rechts weitergegeben, wenn der Wert auf der oberen Leitung kleiner als der auf der unteren ist. Andernfalls werden sie unverändert weitergegeben.

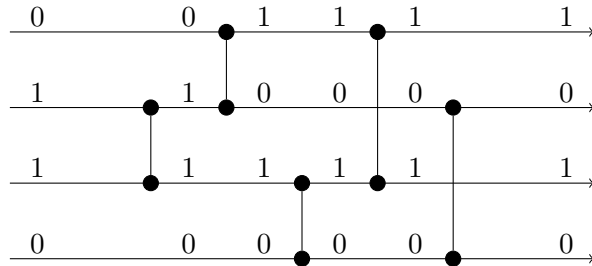
Zum Beispiel würden die Eingaben 3, 4, 1, 9 folgendermaßen durch das obige Netzwerk durchgereicht:



Ein Sortiernetzwerk heißt nun *korrekt*, wenn es für alle möglichen Eingaben eine sortierte Liste ausgibt.

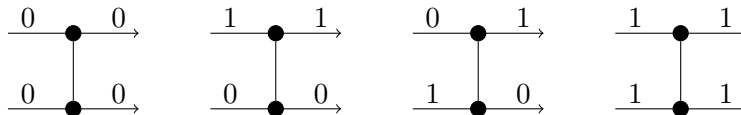
Das obige Netzwerk ist korrekt: Die Ausgabe ist für beliebige Eingaben stets sortiert.

Das horizontal gespiegelte Netzwerk wäre aber nicht korrekt, da die Eingabe 0, 1, 1, 0 zum Beispiel nicht richtig sortiert wird.



Das Ziel dieser Aufgabe ist nun, ein gegebenes Sortiernetzwerk auf Korrektheit zu prüfen. Zur Vereinfachung beschränken wir uns in dieser Aufgabe auf Netzwerke, die nur die Zahlen 0 und 1 als mögliche Eingaben verarbeiten können¹. Auf den Leitungen können also nur die Werte 0 oder 1 vorkommen.

Damit gibt es für die Belegung der Zahlen vor dem Komparator und damit die Wirkung des Komparators folgende vier Möglichkeiten:



Ihre Aufgabe ist es, ein Programm zu schreiben, das entscheidet, ob ein gegebenes Sortiernetzwerk korrekt ist.

Das Programm soll also als Eingabe ein Sortiernetzwerk bekommen. Ein Sortiernetzwerk ist hierbei gegeben sein durch die Anzahl k seiner Leitungen sowie eine Liste von Paaren natürlicher Zahlen $(i_1, j_1), \dots, (i_n, j_n)$. Das Paar (i, j) repräsentiert dabei einen Komparator, der die Leitungen i und j verbindet, wobei die Leitungen von oben nach unten von 0 bis $k - 1$ durchnummeriert sind. Die Liste von Paaren repräsentiert die Komparatoren im Netzwerk in ihrer Reihenfolge von links nach rechts.

Das zu Beginn der Aufgabe angegebene korrekte Beispielnetzwerk ist also durch $k = 4$ und die Liste $(1, 3), (0, 2), (2, 3), (0, 1), (1, 2)$ gegeben. Im nicht korrekten gespiegelten Netzwerk wäre die Liste von Paaren umgedreht.

Ihr Programm muss die Eingabe nicht lesen, sie kann auch im Programm durch geeignete Variableninitialisierung gegeben sein. Es bietet sich an, dazu an ein zweidimensionales `int`-Array zu verwenden und das korrekte Beispielnetzwerk folgendermaßen zu repräsentieren:

```
int k = 4;
int [] [] komparatoren = {{1,3},{0,2},{2,3},{0,1},{1,2}};
```

Ihr Programm soll für Netzwerke mit bis zu 40 Leitern die Korrektheit in akzeptabler Zeit entscheiden können (≤ 1 Minute). Neben den beiden Beispielnetzwerken hier finden sich noch vier weitere auf der Vorlesungsseite.

¹Das ist sogar ausreichend, um die Korrektheit eines allgemein Netzwerks zu zeigen. Man kann zeigen, dass ein Sortiernetzwerk, dass für die Eingaben 0 und 1 korrekt ist, dann auch für beliebige Eingaben korrekt ist.

Hilfestellung:

Nicht vergessen: Als Eingabewerte sind nur 0 oder 1 erlaubt.

Um alle Belegungen der 40 Leiter durchzuprobieren müsste man 2^{40} Belegungen testen. Dies ist auf heutigen Computern jedoch nicht in vertretbarer Zeit möglich. Allerdings gibt es mithilfe der Techniken aus der Vorlesung effizientere Möglichkeiten. Dazu bieten sich unter anderem folgende Ansätze an:

- Mit einem SAT-Solver: Man kodiert die Belegungen der Leitungen aller Zeitpunkte durch Variablen, stellt das korrekte Verhalten der Komparatoren sicher und testet, ob es eine Variablenbelegung gibt, in der die Liste rechts nicht sortiert ist.
Findet man so eine Variablenbelegung, so kann das Netzwerk nicht korrekt sein. Gibt es so eine Variablenbelegung nicht, so sortiert das Netzwerk alle Eingaben korrekt.
- Mit BDDs: Man berechnet für jeden Zeitschritt im Sortiernetzwerk ein BDD für die hier möglicherweise auftretenden Listen an 0en und 1en. Wenn man links mit dem BDD, das immer wahr liefert, anfängt und damit rechts an dem BDD ankommt, das nur sortierte Listen akzeptiert, so ist das Netzwerk korrekt.
- Mit NuSMV (Wenn Sie NuSMV verwenden, genügt es, je eine SMV-Datei für die sechs gegebenen Beispielnetzwerke abzugeben. Sie müssen dann nicht ein Programm schreiben, dass die Korrektheit allgemein entscheidet.): Man verwendet für jedes Element der Liste eine `boolean` Variable, ferner eine Zeitvariable der Form $\{0, 1, 2, \dots, n\}$, welche die Zeitpunkte kodiert. Dann erlaubt man nur Übergänge, die zu den Komparatoren passen und fragt, ob ein Zustand erreichbar ist, welcher alle Komparatoren hinter sich hat, und an dem die Liste nicht sortiert ist.

Abgabe: Sie können ihre Lösungen bis Donnerstag, den 20.6., um 10 Uhr über UniWorX abgeben.