

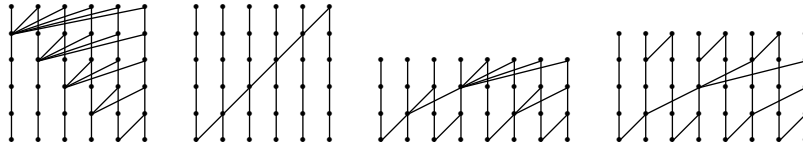
## Übungen zur „Deskriptiven Programmierung“

### Blatt 8

Wird ein scan in Hardware realisiert, spricht man von einem parallel prefix circuit (PPC) bzw. einem parallel suffix circuit.

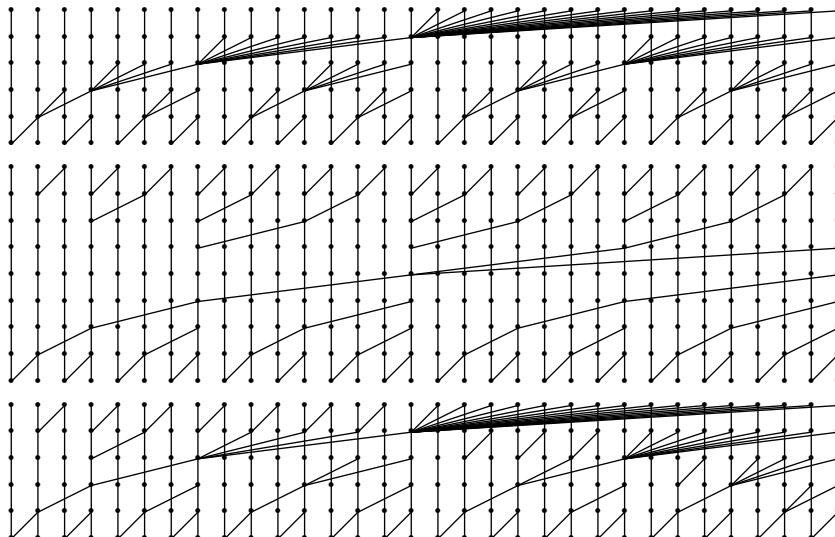
Im Unterschied zum scan wird beim PPC stets vorausgesetzt, dass die binäre Funktion  $\oplus$  assoziativ ist (sie muss aber nicht notwendigerweise ein neutrales Element besitzen). Weiterhin stimmt beim PPC die Anzahl der Eingänge mit der Anzahl der Ausgänge überein, während beim scan die Ausgabeliste ein Element länger ist.

PPCs lassen sich anschaulich durch „Operatorgitter“ darstellen:



Die Eingänge befinden sich unten, die Ausgänge oben; die Daten wandern entsprechend auf den Leitungen von unten nach oben. Treffen an einem Gitterpunkt zwei Leitungen zusammen, werden die Eingabedaten mit der binären Funktion  $\oplus$  verknüpft. Jeder Gitterpunkt hat höchstens einen Seitenzweig zu einem links liegenden Gitterpunkt auf der nächst tieferen Schicht.

**Aufgabe 23.** Entwerfe eine Sprache, um PPCs zu *beschreiben*. Die Sprache sollte eine kompositionale Darstellung von Schaltkreisen erlauben (größere Schaltkreise werden aus kleineren zusammengesetzt) und es ermöglichen, generische PPCs mit einer beliebigen Zahl von Ein- bzw. Ausgängen zu spezifizieren. Verwende die Sprache, um folgende Schaltkreise zu beschreiben.



Erkennst Du das zugrundeliegende Rekursionsmuster?

**Aufgabe 24.** Welche Gesetze erfüllen Deine Sprachkonstrukte? Wie läßt sich die Korrektheit der beschriebenen Schaltkreise nachweisen?

**Aufgabe 25.** Bette die in Aufgabe 23 entworfene Sprache in Haskell ein und teste die Beschreibungen aus.