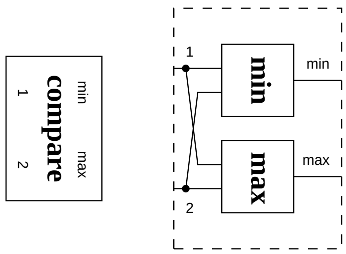
Dr. M. Grabert
Johannes Mayer

Systemnahe Software (SS 2001)

Abgabetermin: 14. Mai 2001

3 Einblick in den Mikrokosmos: Auswertung von Schaltkreisen (10 Punkte)

Nachdem Sie nun schon mal einen Prozess via `fork()` erzeugt haben, können Sie bestimmt auch mehrere Prozesse erzeugen. Bei elektronischen Schaltkreisen, wie sie z. B. im Computer vorkommen, findet wenn möglich Parallelverarbeitung statt, um die Geschwindigkeit zu erhöhen. Voneinander unabhängige (Teil-)Ergebnisse können hierbei gleichzeitig berechnet werden. Damit es interessanter wird, befassen wir uns nicht mit Binärschaltkreisen (d. h. in den Leitungen „fließen“ 0en und 1en), sondern mit **Integer-Schaltkreisen** (d. h. die Leitungen führen Integer-Werte). Hierbei verwenden wir nur zwei verschiedene Gatter-Typen, nämlich **Minimum-** und **Maximum-Gatter**. (Ein Minimum-Gatter berechnet das Minimum seiner beiden Eingaben und liefert dieses auf der Ausgabeleitung. Analog dazu funktioniert ein Maximum-Gatter.) Mit diesen beiden Gattern kann man ein **Vergleichs-Gatter** („compare“) wie folgt simulieren (Vergleichs-Gatter links und Simulation des Gatters rechts):

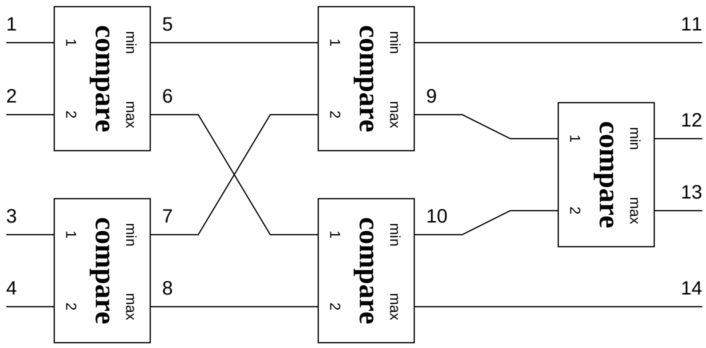


Ein Vergleichs-Gatter vergleicht die beiden Eingaben miteinander und liefert am Ausgang min die kleinere und am Ausgang max die größere von beiden.
Ein **Schaltkreis** besitzt **Eingaben, Ausgaben und Gatter**. Jedes Gatter (wir betrachten nur min- und max-Gatter) besitzt genau zwei Eingänge und genau einen Ausgang. Die Eingaben und die Gatter werden eindeutig durch jeweils eine ganze Zahl beschrieben (die Zahlen für die Eingaben und für die Gatter sind voneinander verschieden). Die Eingaben sind mit einem (oder mehreren) Eingängen eines Gatters verbunden. Die Ausgabe eines Gatters kann mit dem Eingang eines (oder mehrerer) Gatter verbunden werden. Die Ausgaben sind schließlich Ausgängen von Gattern oder Eingaben. Eine wesentliche Eigenschaft von Schaltkreisen ist, dass **keine Zyklen** vorkommen. (Man kann z. B. nicht den Ausgang eines Gatters mit dessen Eingang verbinden.)

Ihre Aufgabe ist es, ein C-Programm zu schreiben, das einen gegebenen Schaltkreis auswertet. Hierzu erhält das Programm als einziges Kommandozeilen-Argument den Namen einer Datei. Aus dieser Datei liest es die Beschreibung des Schaltkreises (deren Aufbau folgt noch). Nun erzeugt es für jede Eingabe eine Datei, deren Name der Nummer dieser Eingabe entspricht[1] und die den Wert dieser Eingabe enthält. Dieser Wert soll für jede Eingabe zuvor von der Standardeingabe gelesen werden. Danach erzeugt es für jedes Gatter einen Prozess, der wartet bis alle Eingabedateien existieren und danach eine Ausgabedatei mit der Nummer des Gatters[2] erzeugt, die den berechneten Wert enthält, und terminiert. Zum Schluss (d. h. wenn die Dateien für alle Ausgaben existieren und etwas enthalten) soll noch für jede Ausgabe deren Wert auf die Standardausgabe geschrieben werden. Außerdem sollen noch alle erzeugten Dateien gelöscht werden.

In der Eingabedatei steht zunächst eine ganze Zahl n, die die Anzahl der Eingaben des Schaltkreises angibt. Danach folgen[3] n ganze Zahlen, die die eindeutigen Nummern der Eingaben angeben. Dann kommt die Zahl m der Ausgaben des Schaltkreises gefolgt von m ganzen Zahlen, die die Nummern dieser Ausgaben repräsentieren (dabei handelt es sich jeweils um die Nummer eines Gatters oder die Nummer einer Eingabe). Dann folgt eine ganze Zahl k, die der Anzahl der Gatter des Schaltkreises entspricht. Für jeden der k Schaltkreise folgt dann eine ganze Zahl (die eindeutige Nummer des Gatters), ein String (der Gattertyp, d. h. „min“ oder „max“) und zwei ganze Zahlen (die Nummern der Eingaben; das sind Nummern von Eingaben des Schaltkreises oder anderen Gattern).

Beispiel: Im folgenden ist ein Schaltkreis dargestellt, der die vier Zahlen aus der Eingabe in sortierter Reihenfolge ausgibt. (Die Nummer eines Gatters steht in folgendem Bild jeweils an der Ausgabeleitung.)



Da wir nur Gatter mit genau einer Ausgabe verwenden wollen, simulieren wir jedes Vergleichsgatter durch ein Minimum- und ein Maximum-Gatter (wie beschrieben). Die zugehörige Schaltkreisbeschreibung sieht dann wie folgt aus:

```
4
1 2 3 4
4
11 12 13 14
10
5 min 1 2
```

[1] Es empfiehlt sich, vor diese Nummer im Dateinamen ein paar Zeichen, wie z. B. „ipc“, zu hängen, damit Sie keine bereits vorhandene Datei überschreiben!

[2] hier gilt die selbe Anmerkung wie bei den Dateinamen für die Eingaben

[3] jeweils durch Whitespace (d. h. Leerzeichen, Tabulator oder Zeilenumbruch) getrennt

```
6 max 1 2
7 min 3 4
8 max 3 4
11 min 5 7
9 max 5 7
10 min 6 8
14 max 6 8
12 min 9 10
13 max 9 10
```

Zum Üben erhalten Sie Schaltkreisbeschreibungen für das *Sortieren von 2 Zahlen*, das *Sortieren von 3 Zahlen*, das *Sortieren von 4 Zahlen*, die *Bestimmung des Medians von 3 Zahlen* und die *Berechnung des Maximums von 8 Zahlen*.

Wichtiger Hinweis: Damit Sie Turing, Thales, etc. nicht durch eine Prozesslawine bedrohen, sollten sie ihr Programm nur auf der lokalen Workstation laufen lassen und *nicht* via ssh auf Turing, ...! Wenn Sie Ihr Programm testen, sollten Sie auch mit `ps -u<loginname>` überwachen, ob Sie evtl. ungewollt Prozesse erzeugen. Eine Prozesslawine auf einem Rechner kann über das neue Kommando `killflood cmd` beendet werden (`man killflood`). Dazu sollten Sie immer ein zweites Terminal-Fenster auf diesem Rechner offen haben.

Viel Erfolg!