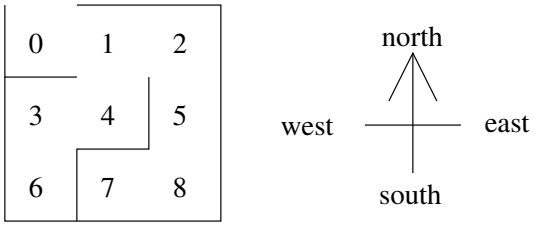
Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Rolf Hennicker

SS 2017
Übungsblatt 8

Entwurf und Implementierung paralleler Programme

Aufgabe 1

Gegeben sei das folgende Labyrinth:



Geben Sie einen parametrisierten FSP-Prozess *LABYRINTH(Start = ...)* an, so dass für jeden Startpunkt *S* durch Deadlock-Analyse des Prozesses *LABYRINTH(S)* der kürzeste Weg aus dem Labyrinth gefunden werden kann.

Aufgabe 2

Die folgende Klasse simuliert das Verhalten von Semaphoren in Java.

```
public class Semaphore {
    private int value;
    public Semaphore (int initial) {
        value = initial;
    }
    public synchronized void down() throws InterruptedException {
        while (value==0) wait();
        value--;
    }
    public synchronized void up() {
        value++;
        notify();
    }
}
```

Geben Sie eine Java-Implementierung des Erzeuger-Verbraucher-Systems von Übungsblatt 7 an, so dass die Prozesskoordination über zwei Semaphore *freePlaces* und *occupiedPlaces* realisiert wird und dafür die Klasse *Buffer* auf *wait* und *notify* Aufrufe verzichtet. Geben Sie auch ein FSP-Modell für Ihre Implementierung an und analysieren Sie das Deadlock-Verhalten.