

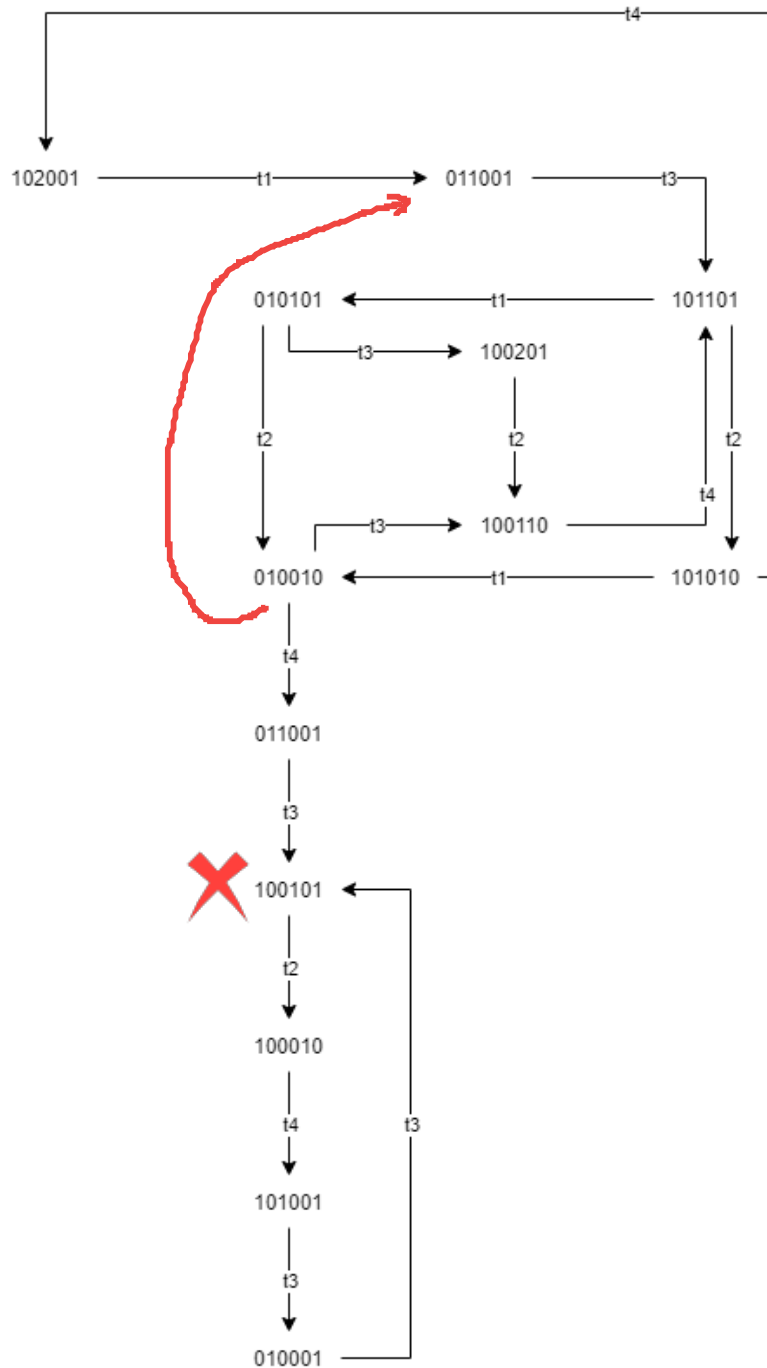
# Software Engineering Hausaufgabe 3

Youran Wang (719511, RAS), Yannick Fuchs (723866, ITS)

November 2021

# 1 Erreichbarkeitsgraph eines Petri-Netzes

## 1.1



\*Hier habe ich einen kleinen Fehler drin. Unten bei der Schleife, dass t3, dass von 101001 zu 010001 hinunter führt, muss eigentlich ein t1 sein.

## 1.2

Nein, dieses Petri-Netz besitzt keine Verklemmung, da bei einer Verklemmung keine Transposition aktiviert sein darf. Dies ist hier allerdings nie der Fall. In diesem Petri-Netz ist immer eine Transposition aktiviert.

## 1.3

Ja, dieses Petri-Netz ist lebendig, da alle Transitionen von jeder erreichbaren Markierung aus wieder aktiviert werden können.

## 1.4

Nein, dies ist nicht möglich, da die Transitionen t1 und t2 ja auch jeweils ein Token aus Positionen haben wollen, an welche wir ohne die Transitionen t3 und t4 gar nicht kommen. Diese setzen allerdings wieder die Schaltung von t1 und t2 voraus. Somit sind die Token konstant in Bewegung und es können maximal an den beiden mittleren Stellen je zwei Token zeitgleich sein, da diese sonst wo anders fehlen.

## 2 Entwurf eines Petri-Netzes

### 2.1

Siehe Petri-Netz auf nächster Seite. Mit **kleiner Korrektur**: Der Pfeil vom Einlagern zum Lager, hat das Gewicht 2, nicht wie dort gezeigt 1.

### 2.2

ja, unter der Bedingung, dass wir irgendwann komplett das Lager leer machen und alle Fahrräder vom Gegenspieler des Fahrradhändlers geklaut werden, so können wir nichts mehr machen und haben eine systemweite Verklemmung.

Ebenso wird es zu einer Verklemmung kommen, wenn alle bis auf eines oder zwei der Räder geklaut wurden und wir keine mehr im Lager und in der Ersatzteilstelle haben. Dann kann nämlich nur ein Reparaturplatz bzw. die beiden Reparaturplätze voll gemacht werden und wir haben keine Ersatzteile. Dies führt dann ebenso dazu, dass wir nicht mehr schalten können und somit eine systemweite Verklemmung haben.

### 2.3

Dieses Petri-Netz ist nicht lebendig, da es eine Markierungen gibt von welcher keine Transition mehr erreicht werden kann. Diese ist das Fahrradlager des Gegenspielers.

## 3 UML-Zustandsdiagramm

Siehe Zustandsdiagramm auf letzter Seite.

