

Prof. Dr. Margarita Esponda

# Nichtsequentielle Programmierung, SoSe 2017

## Project 1

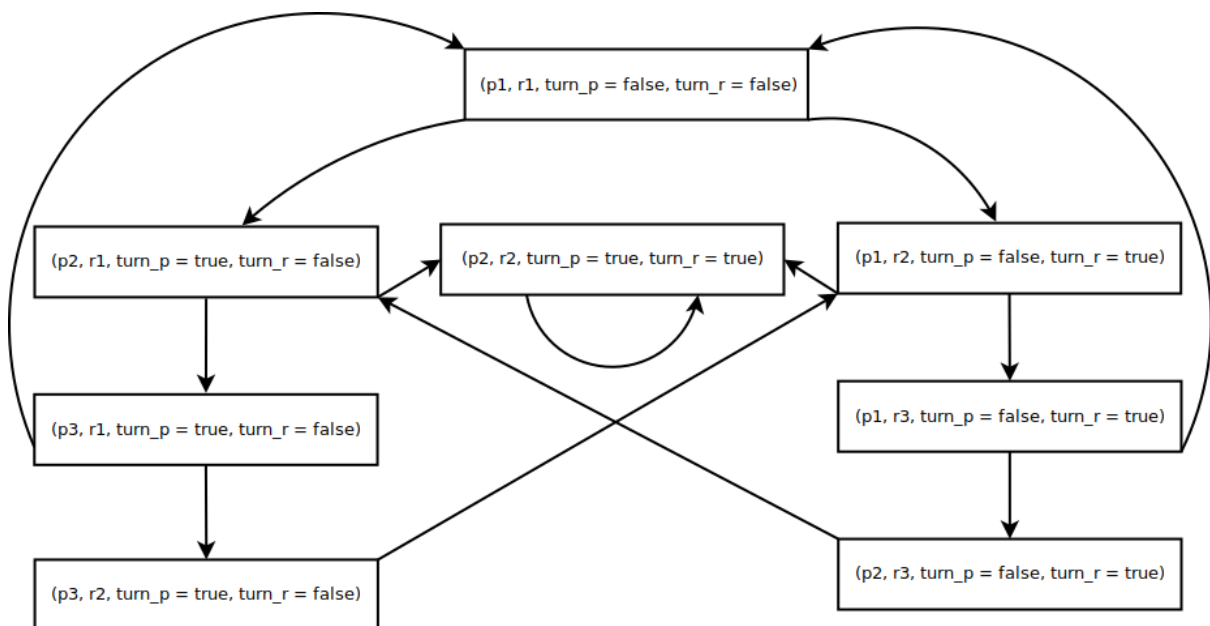
TutorIn: Lilli Walter  
Tutorium Tutorium 6

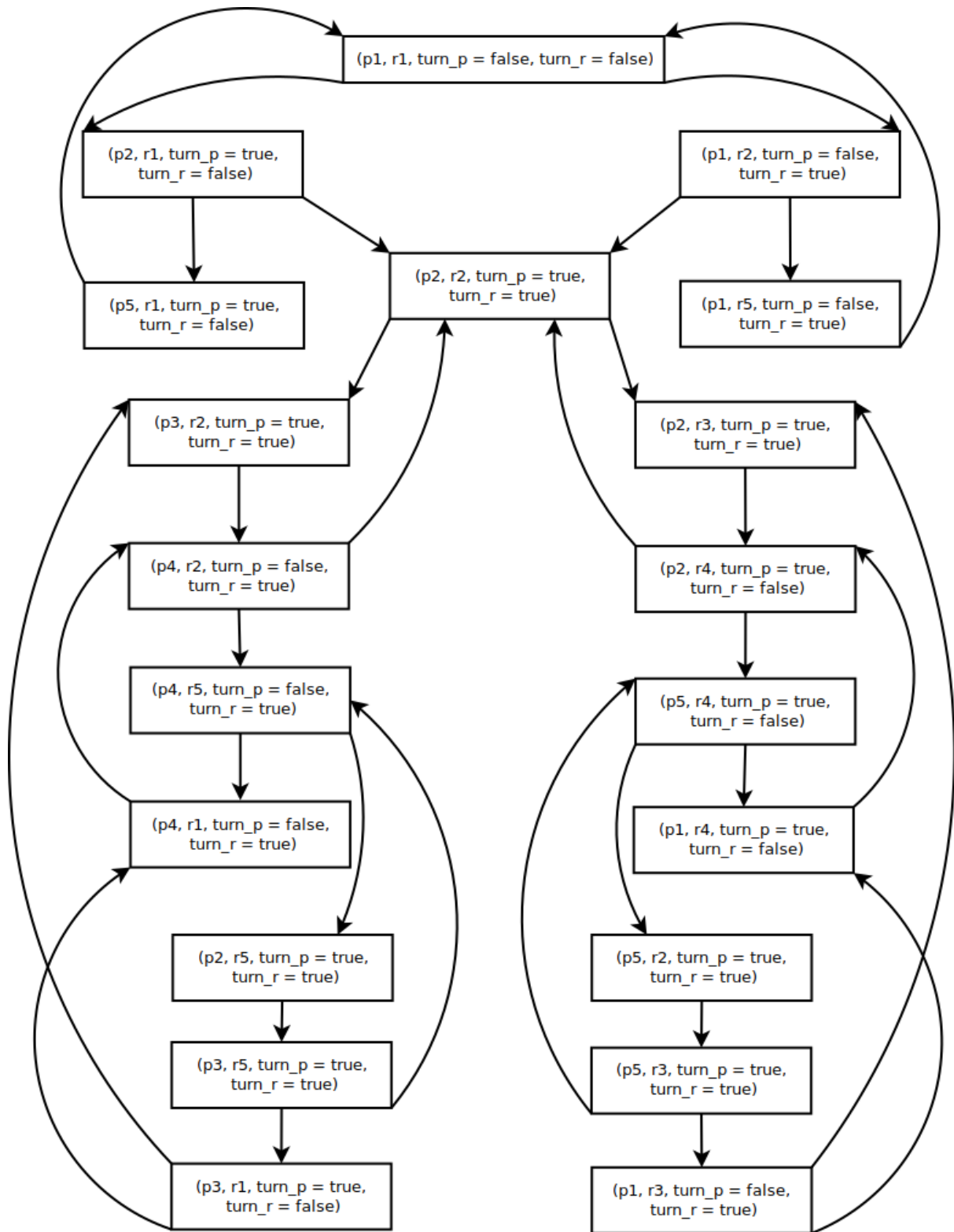
Boyan Hristov

16. Mai 2017

### 1 Aufgabe 1

Man kann erkennen, dass Wechselseitiger Ausschluss gewährleistet ist, da es keine Zustände  $(p3, r3)$  für den 1. Algorithmus und  $(p5, r5)$  für den 2. Algorithmus existieren.





## 2 Aufgabe 2

Wir haben irgendwann erkannt, dass die Idee die Aufgabe nicht solche ist, da diese nur 8 Punkte gibt. Eigentlich ist aber in Assembler so, dass man erstmal ein Exchange machen kann, und dann noch Compare + JE oder JNE. Wir haben deswegen gemeint, dass 'until !local\_r' eine Art von 'If true, jump to r2/p2, else continue to r4/p4'. Wir dachten dass es trotzdem Sinn macht diese Variante zu zeigen, wollten auch unsere Arbeit nicht wegschmeißen.

Man erkennt, dass es keine Deadlocks gibt, da es keine Zustände gibt mit nur eingehende Pfeile. Wechselseitiger Ausschluss ist auch gewährleistet, da die verbotenen Zustände (diese, die p4 und r4 gleichzeitig enthalten) nicht existieren.

