



22. Deadlock – einfach verfügbare Ressourcen (1 Punkt)

Es gebe fünf Prozesse P_1, P_2, P_3, P_4 und P_5 sowie sechs Ressourcen $R_1, R_2, R_3, R_4, R_5, R_6$. Es gelte dabei:

- P_1 hat R_2 belegt und fordert R_5 an.
- P_2 hat R_4 belegt und fordert R_2 und R_3 an.
- P_3 hat R_3 belegt und fordert R_1 an.
- P_4 hat R_1 belegt und fordert R_6 an.
- P_5 hat R_5 belegt und fordert R_4 an.

- a) Zeichnen Sie den Ressourcen-Zuordnungsgraph für dieses Szenario und leiten Sie daraus ab, ob sich die fünf Prozesse im Deadlock-Zustand befinden. Begründen Sie Ihre Antwort.
- b) Überprüfen Sie Ihr Ergebnis, indem Sie den Deadlock-Erkennungs-Algorithmus (mit den Belegungs- und Anforderungsmatrizen) durchführen.

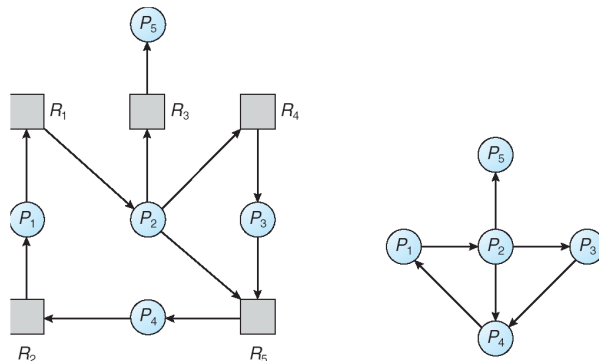
23. Wait-for-Graphen (1 Punkt)

Mit dem folgenden Algorithmus können Sie jeden Ressourcen-Zuordnungs-Graph in einen so genannten **Wait-for-Graph** umwandeln:

Seien P_1, \dots, P_n alle Prozess-Knoten und R_1, \dots, R_m alle Ressourcen-Knoten des ursprünglichen Graphs.

1. Der neue Graph enthält alle Prozess-Knoten P_1, \dots, P_n , aber keine Ressourcen-Knoten.
2. Für jede Kombination $P \rightarrow R \rightarrow P'$ im Ressourcen-Zuordnungs-Graph zeichnen Sie einen gerichtete Kante $P \rightarrow P'$ in den Wait-for-Graph

Beispiel:



- a) Beweisen Sie, dass es genau dann einen Kreis im Wait-for-Graph gibt, wenn es auch einen Kreis im zugrundeliegenden Ressourcen-Zuordnungs-Graph gibt. (Ein solcher „genau dann wenn“-Beweis besteht immer aus zwei Richtungen, \Leftarrow und \Rightarrow .)

Ersatzweise argumentieren Sie anschaulich, warum diese Behauptung gilt.

- b) Erzeugen Sie zum Graph aus Aufgabe 22) den Wait-for-Graph und untersuchen Sie diesen auf Kreise.



24. Deadlock-Erkennung (3 Punkte)

- a) Für drei Ressourcen-Klassen A, B und C sowie vier Prozesse P_1 bis P_4 seien folgende Ressourcenbelegungen (**C**) und Anforderungen (**R**) gegeben:

$$C = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 2 \\ 1 & 3 & 0 \\ 0 & 0 & 4 \end{pmatrix} \quad R = \begin{pmatrix} 2 & 0 & 6 \\ 3 & 4 & 3 \\ 1 & 2 & 2 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{matrix} E = (4 \ 9 \ 7) \\ A = (1 \ 3 \ 1) \end{matrix}$$

A gibt die aktuell noch verfügbaren Ressourcen an, E die Gesamtressourcen.

Ist dieses System im Deadlock-Zustand? Falls ja, nennen Sie die daran beteiligten Prozesse und Ressourcen.

- b) Es gibt auch eine Variante dieses Algorithmus, die mit Ressourcen-Zuordnungs-Graphen arbeitet. Lesen Sie die Beschreibung unter

http://olli.informatik.uni-oldenburg.de/Deadlock/Html/Erkennung2_Reduzierung.html

und erklären Sie, warum die dort beschriebene **Reduzierung des Graphen** äquivalent zum in Aufgabe a) eingesetzten Algorithmus ist.

- c) Die Beschreibung auf der genannten Webseite gibt nur ein Beispiel, in dem jeder Prozess jede Ressource maximal 1x anfordern kann. Beschreiben Sie zwei Möglichkeiten, wie sich auch Anforderungen in den Graph integrieren lassen, die mehr als 1x vorliegen.