

14a)

i)

Die Wurzelprozesse sind bei den beiden Aufrufen verschieden und somit nicht gueltig.

ii)

Die beiden Aufrufe blockieren und warten darauf, dass alle anderen Prozesse MPI_Bcast mit dem gleichen Wurzelknoten aufrufen. Da beide Prozesse verschiedene Wurzelknoten erwarten kommt es zu einem Deadlock.

14b)

Das linke Listing(synchronisierte Reihenfolge) verwendet keinen Systempuffer. Dabei blockieren P0 und P1 weil die „Gegenstelle“ noch nicht gesendet hat und keine Möglichkeit besteht die Daten zu puffern. Nachdem P1 und P2 kommuniziert haben wirkt der Broadcast wie eine Barriere, da die Anweisung blockiert bis alle den Broadcast durchgeführt haben. Danach können P0 und P1 kommunizieren.

Beim rechten Listing wird von Prozess 0 ein **MPI_Bcast** abgesetzt, während die beiden anderen Prozesse noch nicht bis zu ihrer **MPI_Bcast**-Anweisung gekommen sind. Dabei wird das **MPI_Bcast**, das vor einem **MPI_Bcast** eines parallelen Prozesses abgesetzt wird, im Systempuffer des Senders oder des Empfängers zwischengespeichert bis das entsprechende **MPI_Bcast** des empfangenden (wenn root dann sendenden) Prozesses ausgeführt wird und die Daten aus dem Puffer holt. An dieser Stelle könnte es zu einem Nichtdeterminismus kommen, wenn die **MPI_Send**-Anweisung aus P0 eine Datenabhängigkeit zum vorangehenden **MPI_Bcast** hat und deshalb blockiert, die Daten aber nie ankommen, da **MPI_Bcast** in P1 erst ausgeführt werden kann wenn P0 sein **MPI_Send** abgesetzt hat.