Fragen, Antworten, Kommentare und Hinweise

Frage: Irgendwie ist die Entwicklung mit Promela recht aufwändig.

Antwort: Das stimmt aus mehreren Gründen. Zunächst gibt es neue Sprachkonstrukte und die Idee des Einsatzes des Nichtdeterminismus muss verstanden werden. Danach müssen Sie die Spezifikation entwickeln und mit dem Simulator ausprobieren, was grob der Programmierung und dem Testen entspricht. Hier wären Sie mit der Programmierung fertig, bei der Verifikation folgt aber dann der entscheidende Teil mit einer Formulierung der Anforderungen, die dann vom Model Checker überprüft werden. Dabei ist zunächst sicherzustellen, dass die Anforderungen präzise beschrieben sind und dann kann das Ergebnis des Model Checkers im Worst Case dazu führen, dass vorne wieder begonnen werden muss.

Frage: Bei der Aufgabe 12 werden die Prozesse bei mir in der richtigen Reihenfolge durchgeführt, jedoch stehen mehrere Auswahloptionen bei der interaktiven Ausführung zur Verfügung. Wann wird genau davon ausgegangen, dass ein alternativer Ablauf möglich ist? Immer dann, wenn ein Doppelpunkt vor der Anweisung steht oder lässt sich das nicht direkt verallgemeinern?

Antwort: Bei der Ausführung von Promela wird immer geprüft, welche der nächsten Anweisungen ausführbar ist. (Erinnerung, u. a. der Boolesche Term x==3 ist ausführbar, wenn x den Wert 3 hat). Bei der Aufgabe soll bei der interaktiven Simulation immer nur ein Schritt (und quit) Ihnen angeboten werden. Wenn Sie die drei Prozesse schreiben, die jeweils von 1 bis 10 hochzählen, werden Ihnen typischerweise drei Alternativen angeboten. Das macht Ihre Spezifikation im Wesentlichen auch, aber Sie erzeugen immer neue Prozesse (run). Die Prozesse laufen getrennt und bleiben bestehen. Sie sollten also in init im Wesentlichen nur die drei Prozesse starten.

Frage: Bei der Aufgabe 14 habe ich zunächst die grundlegende Spezifikation geschrieben. Für den ersten Wert (1) läuft es richtig durch. Wenn der zweite Wert an P2 geschickt werden soll, wird der Kanal nicht aufgerufen. Kann es daran liegen, dass der Kanal schon einmal genutzt wurde? Lässt sich dieser in irgendeiner Form leeren? Eine Möglichkeit zu prüfen, ob er voll ist, wurde ja bereits in der Vorlesung vorgestellt, nur habe ich keine Variante entdeckt, die mir das leeren ermöglichen würde.

Antwort: Bei der synchronen Kommunikation gibt es kein Leeren eines Kanals. Es kann immer versucht werden, auf diesen zu schreiben und von diesem zu lesen. Eine Kommunikation findet nur statt, wenn zwei Prozesse bereut sind, also einer senden will chan!42 und ein anderer empfangen kann chan?var. Bei der asynchronen Kommunikation werden Werte in einen Puffer des Kanals beim Senden geschrieben und beim Lesen wieder aus diesem Puffer entfernt.
Bei Ihrer Spezifikation wollen Sie die Werte in einer Schleife senden. Das ist ok. Beim Empfangen machen Sie das genau einmal in P2 und P3, die Prozesse sind dann fertig. Also fehlt in P2 und P3 eine Schleife. Tipp: Ersetzen Sie die for(i:...)-Schleife dadurch, dass Sie i selbst in einer do-Schleife hochzählen, das macht nachfolgende Teilaufgaben einfacher.