

Rechnernetze und verteilte Systeme

Übungsblatt 5

Ausgabe: 6. November, **Besprechung:** 14. November – 17. November, **keine Abgabepflicht**

Quizfragen

- Zu welchem Zweck hat TCP mehr verfügbare Sequenznummern als die rdt-Protokolle?
- Welches Fragment der verbindungsorientierten Kommunikation wird durch die rdt-Protokolle nicht abgedeckt?
- Ist es möglich, dass der Datentransfer über UDP fehlerfrei ist?
- Angenommen, die Round-Trip-Zeit ist dem Sender bekannt. Wäre dann im rdt3.0-Protokoll immer noch ein Timer notwendig?

Aufgabe 5.1

UDP und TCP benutzen Einerkomplement zur Fehlererkennung.

- (a) Für folgende 8-Bit-Bytes, was ist das 8-Bit-Einerkomplement deren Einerkomplementsumme: 00110101, 11110101, 11001001?
- (b) Wie erkennt der Empfänger Fehler mit dem Einerkomplement-Schema? Welche Fehler können erkannt werden, welche nicht?

Aufgabe 5.2

Zeichnen Sie den erweiterten Mealy-Automaten für den Empfänger von rdt3.0.

Aufgabe 5.3

In der Vorlesung fand eine kritische Auseinandersetzung mit der von Kurose und Ross verwendeten Syntax für Automaten, die Protokoll-Instanzen beschreiben, statt. Eine konsistente formale Syntax für einen erweiterten Mealy-Automaten finden Sie in den Vorlesungsfolien.

Bearbeiten Sie die Aufgabe aus einer alten Klausur.

Anmerkung: Die angegebenen 15 Punkte waren in der Klausur zu erreichen und werden in den Übungen nicht vergeben :)

Transportsystem und Protokolle, Erweiterter Mealy-Automat

[15 Punkte]

Gegeben ist ein Szenario, in welchem eine Transportprotokoll-Instanz S Nutzdaten an eine entfernte Transportprotokoll-Instanz E zu übertragen hat.

Der Netzdienst kann Pakete verlieren. Die Kombination aus positiver Quittierung, Zeitüberwachung und Wiederholung wird in Stop-and-Go-Version zur Verlust-Fehlerbehandlung eingesetzt. Wenn ein Datum dreimal erfolglos gesendet wurde, bricht die Instanz S ihre Aktivitäten ab. Andere Mechanismen sind nicht vorgesehen. Wir abstrahieren von der Adressierung der Netzdienst-Pakete und betrachten nur deren Nutzdaten $tpdu$. Dort abstrahieren wir von den genauen PDU-Formaten. Für Pakete von S nach E gelte $tpdu$ ="zu sendendes Nutzdatum d ". Für Pakete von E nach S gelte $tpdu$ ="ACK".

Die Instanz S hat folgende Eingaben:

- | | |
|-------------------|---|
| TDatReq(d) | Übergabe des Datums d von Anwendungsprozess an S |
| NDatInd($tpdu$) | Übergabe des empfangenen Datums pdu von Netzdienst an S |
| TimerAlert | Der Kurzzeitwecker signalisiert den Ablauf der Weckzeit |

Die Instanz S hat folgende Ausgaben:

- | | |
|-------------------|---|
| TAbsortInd | Abbruch-Anzeige der Instanz S an Anwendungsprozess |
| NDatReq($tpdu$) | Übergabe des zu sendenden Datums pdu von S an Netzdienst |
| TimerStart | Befehl zum Start des Kurzzeitweckers (ein aktiver Wecker wird zuvor angehalten) |
| TimerStop | Befehl zum Anhalten des Kurzzeitweckers |

Das Verhalten von S soll an Hand von folgendem erweiterten Mealy-Automaten beschrieben werden. Vervollständigen Sie die Variablendefinition, Initialisierungsbedingung und Transitionsklausen im Diagramm!

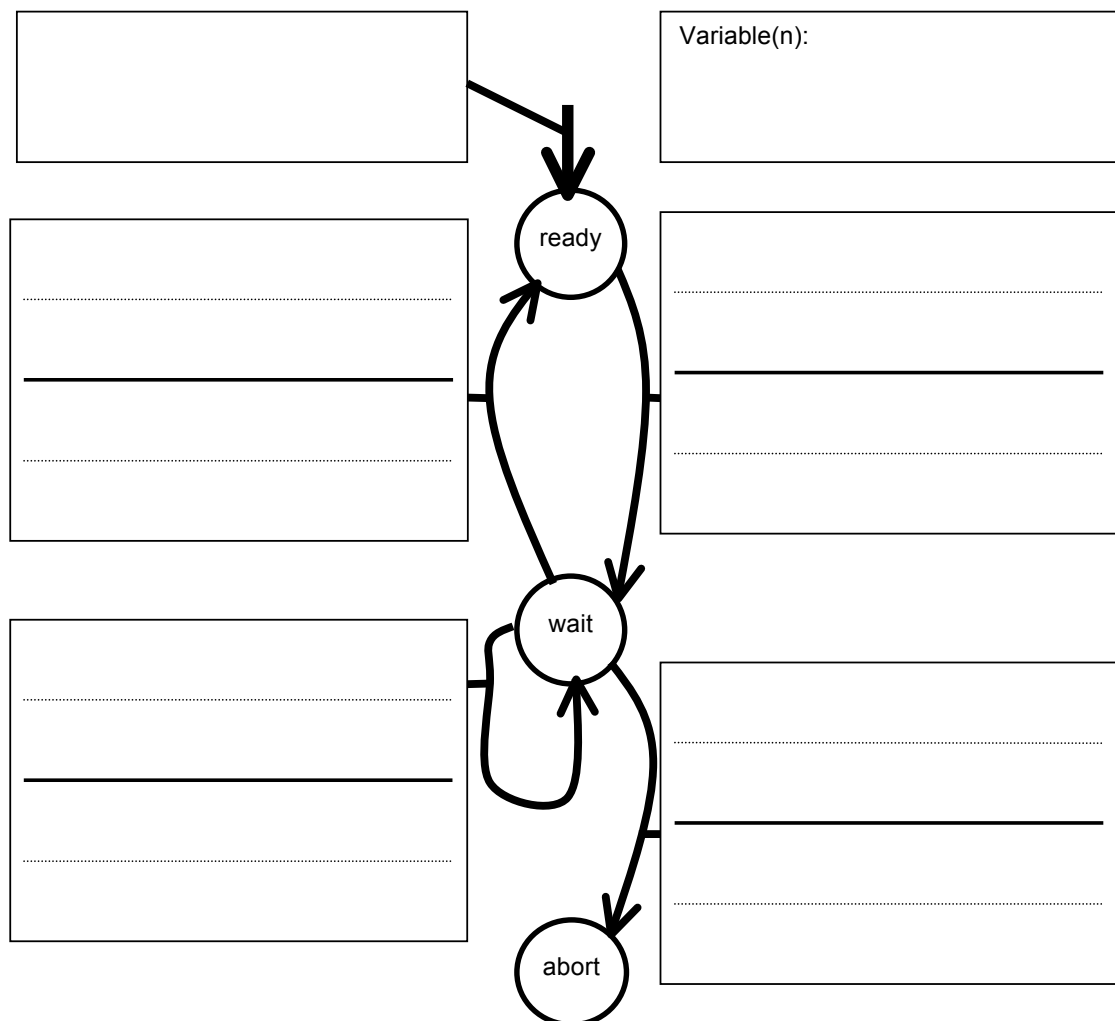


Abbildung 1: Aufgabentext zu Aufgabe 5.2