SVP / THP2 WiSe 14	Praktikum 4 (Prüfungsvorleistung)	Prof. Dr. B. Buth 3.12.2014

1 CSP Stable Failures

Definitionen von stable failures

- Prozess P *stable*

$$P \downarrow =_{Def} \neg(P \xrightarrow{\tau})$$

- *refusal* eines Prozesses P

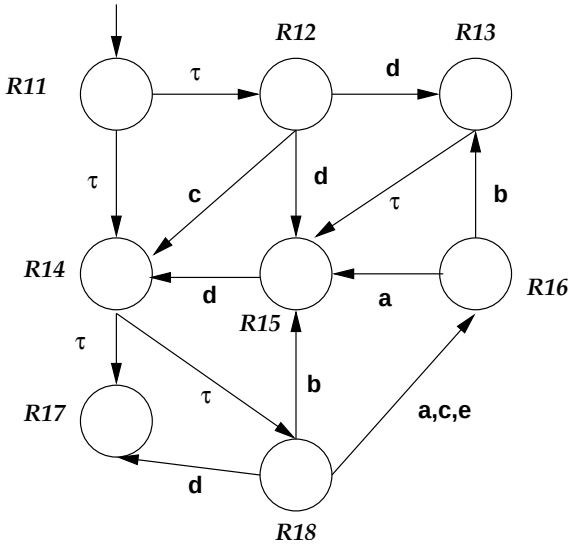
$$P \ ref \ X =_{Def} \exists P_1 \bullet P \stackrel{\langle\rangle}{\Longrightarrow} P_1 \wedge P_1 \downarrow \wedge (\forall a \in X \bullet \neg(P_1 \xrightarrow{a}))$$

- *stable failure* von P:

$$(tr, X) \in \mathcal{SF}(P) \Leftrightarrow \exists P_2 \bullet P \stackrel{tr}{\Longrightarrow} P_2 \wedge P_2 \downarrow \wedge P_2 \ ref \ X$$

1.1 Stable Failures in Transitionsgraphen

Bestimmen Sie mit Hilfe der Definitionen der Stable Failures die stabilen Zustände in den folgenden Transistionsgraphen und geben Sie für jeden Knoten die Refusal-Mengen R_{ij} als maximale Refusals an. Gehen sie von einer Umgebung aus mit $\Sigma = \{a, b, c, d, e, f\}$
Anmerkung: Tragen Sie $R_{ij} = \{\}$ ein, wenn der Zustand nicht stabil ist.



2 CSP Modelle

2.1 Scheduler in CSP spezifizieren

Spezifizieren Sie in CSP ein Prozess-System, dass einen FIFO-Scheduler und einem Timer modelliert. Der Scheduler soll demonstriert werden mit 3 Tasks. Dabei sollen die folgenden Anforderungen im Modell umgesezt werden:

- Der Scheduler kann grundsätzlich eine beliebige Anzahl von Tasks schedulen, dh auch mehr als 3 Tasks
- Eine Task befindet sich zu jedem Zeitpunkt in einem der Zustände idle (ohne Wunsch zu laufen), eager (will laufen, ist aber vom Scheduler nicht in die Liste der zu behandelnden Prozesse aufgenommen), waiting (steht in der waiting list des Schedulers) oder active (wird ausgeführt).

- Tasks identifizieren sich durch ihre Id (Tasknummer); die dummy-Task hat die Id 0.
- Tasks wissen die von ihnen benötigte Ausführungszeit und teilen sie dem Scheduler bei einem request mit.
- Eine Task wird aus der Systemumgebung heraus angestossen vom Zustand idle in den Zustand eager zu wechseln (start_task)
- Eine Task im Zustand eager kann dem Scheduler mitteilen, dass sie aktiv werden möchte (request)
- Der Scheduler teilt einer Task mit, wenn er sie auf die waiting list setzt (set_waiting), oder wenn sie aktiv wird (set_active) oder wenn sie de-scheduled wird (set_idle).
- Der Timer teilt dem Scheduler das Vergehen von Zeiteinheiten mit über ein clock-Signal
- Der Scheduler setzt eine Task als letzte in die waiting list, wenn sie einen request schickt.
- Der Scheduler kennt die aktive Task und ihre benötigte Ausführungzeit und zählt die Restzeit der Task herunter
- Nach Ausführung einer gesamten Task wird die nächste Task in der waiting list aktiv gesetzt - falls es keine solche Task gibt, wird eine dummy-Task (Id = 0) ge-scheduled; die dummy-Task ist auch diejenige, die zum Start des Systems ge-scheduled ist.

Verwenden Sie die folgenden Daten- und Channel-Deklarationen in Ihrer Spezifikation:

```
nametype Id = {0,1,2,3}
nametype Timeslots = {1..10}
channel set_idle, set_waiting, set_active, start_task : Id
channel request: Id.Timeslots
channel clock
```

a) Geben Sie die Spezifikationen als CSP-Prozes an für Prozesse TASK(..), SCHED(...), CLOCK(..) und das System SYS, das aus den benötigten Prozessinstanzen mit Tasks 1,2,3 geeignet zusammengesetzt werden soll. Dabei soll die Task 1 drei Zeiteinheiten, Task 2 fünf und Task 3 sechs Zeiteinheiten laufen.

b) Geben Sie geeignete Testprozesse an, die demonstrieren, dass Ihr System tatsächlich einen FIFO-Scheduler beschreibt.

Demonstrieren Sie das System und die Tests im Praktikum.

Bearbeitungszeitraum: Praktikumstermin 4

Die Bearbeitung der Aufgaben soll in den Teams von 3-4 Personen erfolgen,

Abnahme im Praktikum