

Tentamen Formele Specificatiemethoden voor Software Engineering (213520)

10 april 2008, 9.00-12.30 uur.

BELANGRIJK: geef op je tentamen duidelijk aan:

- je studierichting
- of je beide huiswerkgaven gemaakt hebt, en bij welke werkcollegeleider.

Bij dit tentamen mag je gebruiken: de Z reference card, de FSP Quick Reference card, en de SDL survival kit (geen boeken).

Puntenwaardering:

opgave 1: 8 punten, opgave 2: 8 punten, opgave 3: 4 punten.

-
1. Het bureau onderwijszaken heeft behoefte aan een systeem voor het inschrijven van studenten voor vakken. In deze opgave wordt gevraagd dit systeem in Z formeel te specificeren.

Een bij het systeem geregistreerde student kan zich voor een vak maximaal 1 keer inschrijven. Uiteraard is het toegestaan je voor meer dan 1 vak in te schrijven. Elk door het systeem geregistreerd vak heeft een eigen maximale capaciteit, die ook door het systeem wordt vastgelegd. Er kunnen zich nooit meer studenten inschrijven voor een vak dan de maximale capaciteit van het vak.

- (a) Karakteriseer de mogelijke (legale) toestanden van het inschrijvingsstelsel d.m.v. een toestandsschema. Geef ook de initiële toestand van het systeem.
 - (b) Definieer een operatie om een nieuw vak in te voeren, met een bepaalde capaciteit.
 - (c) Definieer een operatie die een student bij een vak inschrijft. Is de student nog niet geregistreerd, dan gebeurt dit automatisch. Zorg dat de operatie robuust is, en gebruik schemasamenstelling.
 - (d) Definieer een operatie die het geven van een vak annuleert. Dat betekent dat alle inschrijvingen voor dat vak ongedaan gemaakt worden, en dat niemand zich meer voor dat vak kan inschrijven. De operatie dient tevens de verzameling studenten op te leveren die zich voor dat vak hadden ingeschreven. Het vak zelf dient wel geregistreerd te blijven.
 - (e) Definieer een operatie die, met als invoer een vak, als uitvoer geeft hoeveel studenten zich nog voor dat vak kunnen inschrijven.
2. In een sauna kan een klant een cabine reserveren bij de ingang. Zodra een cabine vrijkomt wordt ie aangezet, en wordt het nummer van de cabine aan de klant gegeven. De klant gaat naar de aangegeven cabine; na elke 5 minuten gaat of de cabine uit, of de klant drukt op een "more" knop en krijgt zo nog 5 minuten sauna. De maximum tijd die een klant kan krijgen is 15 minuten. Nadat de cabine uitgaat signaleert ie naar de uitgang het bedrag dat de klant moet betalen (1 eenheid voor elke 5 minuten). De klant betaalt vervolgens bij de ingang.

- (a) Specificeer de sauna voor 1 klant en 1 cabine. Neem het volgende proces voor de ingang:

```
ENTRANCE = ( reserve -> turn_on -> get_number -> ENTRANCE
             | amount [m:M] -> pay [m] -> ENTRANCE ).
```

Neem aan dat een klant die 15 minuten blijft er als volgt uitziet:

```
CUSTOMER = (reserve -> get_number ->
            more -> more -> end -> pay[m:M] -> CUSTOMER).
```

- (b) Specificeer nu een sauna met 2 cabines en 3 klanten (hint: denk goed na over hoe je het proces ENTRANCE moet veranderen om deadlocks te voorkomen). Je mag aannemen dat elke klant 10 minuten wil blijven.
 - (c) Specificeer een safety property die zegt dat het verschil tussen het aantal `reserve` acties en het aantal `pay` acties nooit groter is dan 2. Geldt deze eigenschap voor jouw sauna?
 - (d) Specificeer een progress property die zegt dat klant nummer 3 uiteindelijk een cabine krijgt. Geldt deze eigenschap? Geeft nu prioriteit aan de `request` acties van klant 1 en klant 2. Geldt de eigenschap nu nog steeds?
3. (a) Specificeer een SDL proces P dat een a uitvoert, een c ontvangt, een d ontvangt en termineert, en een proces Q dat a ontvangt, b ontvangt, c uitvoert en termineert.
- (b) Geef een SDL block met de twee bovenstaande SDL processen, en met signaal routes die de signalen b en d laten passeren over een unidirectioneel kanaal Env met de omgeving, en alle andere signalen laten passeren tussen de twee processen.
- (c) Geef het MSC van bovenstaand systeem.
- (d) Geef vijf verschillende traces (N.B. de events van deze traces zijn bijvoorbeeld $/a \ @ \ Q$, $b/ \ @ \ Env$ etc.).