Automaten en Berekenbaarheid
Bespreking examen juni 2019

Prof. Dr. Bart Bogaerts
Mathieu Reymond

1 juli 2019

Het slaagcijfer voor *Automaten en Berekenbaarheid* lag laag dit jaar. Ik hoop dat deze gedetailleerde bespreking van de resultaten een hulp kan zijn bij het studeren in augustus.

We zullen beginnen met wat algemene statistieken. De volgende tabel bevat een overzicht van de verschillende vragen en de verdeling van de scores op die vragen (het aantal studenten in elke scoregroep).

	Vr1	**Vr2**	**Vr3**	**Vr4**	**Vr5**	**Oefeningen**	**Theorie**	**Totaal**
Relatief gewicht	4	3	7	2	4			
Gemiddelde (op 20)	13.3	10.7	4.8	14.9	11.1	9.6	6.4	8.3
0	2	8	10	2	14	0	1	0
$\in]\mathbf{0}, \mathbf{5}[$	3	4	14	1	2	6	20	7
$\in [\mathbf{5}, \mathbf{8}[$	3	0	12	2	3	12	7	10
$\in [\mathbf{8}, \mathbf{10}[$	0	7	2	0	0	7	4	9
$\in [\mathbf{10}, \mathbf{12}[$	9	0	6	5	2	5	3	5
$\in [\mathbf{12}, \mathbf{15}[$	6	10	1	3	0	12	0	6
$\in [\mathbf{15}, \mathbf{20}]$	23	17	1	33	25	4	5	3

In de tabel zien we meteen dat de scores op vragen 1 en 4 beduidend hoger liggen dan de rest van de vragen. De reden hiervoor is waarschijnlijk dat deze vragen geen creativiteit/inzicht vereisen, maar kunnen opgelost worden door enkel geziene procedures uit te voeren. Vraag 2 valt ook in deze categorie vragen, maar daarbij waren er wat meer "pitfalls" die tot puntenaftrek konden leiden, en er waren ook enkele studenten die niet gezien hadden dat dit eigenlijk de vraag "zet een automaat om in een reguliere expressie" in een jasje was. Vraag 5 was, voor een oefening over Turing machines, redelijk gemakkelijk (kon opgelost met een klein aantal transities). Wat we daar zien is twee grote groepen mensen: zij die "gezien" hebben hoe het moet scoren heel goed ($\geqslant 15$), terwijl zij die het niet gezien hebben dicht bij de nul zitten. Vraag 3 (in detail besproken hieronder) was moeilijk. Ze vereiste inzicht (om de talen te klassificeren) en creativiteit (voor het toepassen van de pompstelling of het construeren van grammaticas/automaten). In die zin sloot vraag 3 het meest aan bij het mondeling examen, waar ook vooral naar inzicht gepeild werd; de scores op vraag 3 en het mondeling lagen dan ook erg in elkaars buurt (en waren erg laag).

Opmerking: het relatief gewicht in bovenstaande tabel zegt van elke vraag voor hoeveel punten ze meetelde in de berekening van de totaalscore. Dit komt niet steeds overeen met het cijfer waarop elke gevraagd gequoteerd werd.

Theorie

Het mondeling examen (theorie) was erbarmelijk slecht. Naast de traditionele redenen zijn volgende verklaringen misschien ook geldig (gerelateerd aan het feit dat ik dit vak voor het eerst doceer):

- Misschien waren de verwachtingen niet 100% correct over het detail waarin de leerstof begrepen moet zijn of hoe streng er gequoteerd wordt; mogelijks zijn daar verschillen met mijn voorganger.

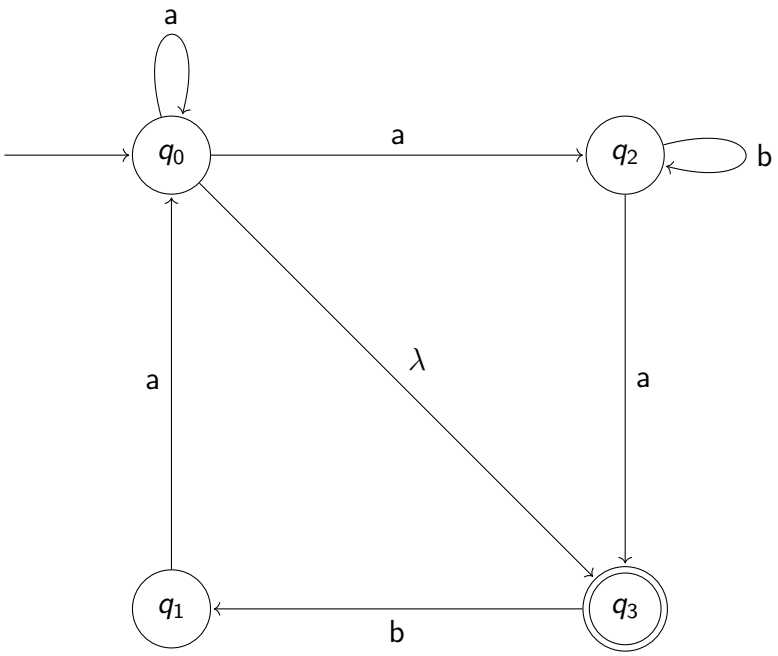
- Er circuleerden nog geen vorige versies van mijn mondelinge examens (gezien ik er nog geen had opgesteld). De bijvragen krijg je uiteraard niet cadeau, gezien die deels voorbereid, maar ook deels geleid zijn door je antwoord, maar alle opgaves die ik deze zittijd gebruikte, vinden jullie vanaf vandaag terug bij bestanden onder de map "examens". Dit is **geen** exhaustieve lijst van mogelijke examenvragen. Onthou ook (dit wordt lager nog meer in detail besproken): een letterlijk antwoord op deze vragen kunnen geven volstaat niet om te slagen: je moet begrijpen en kunnen uitleggen **waarom** je antwoord juist is en verbanden met de rest van de cursus kunnen bespreken.

Heel wat studenten lieten duidelijk zien dat ze de procedures/bewijzen vanbuiten hadden geleerd, maar toonden nauwelijks begrip en hadden dan ook veel moeite met het antwoorden op vragen zoals "Waarom werkt deze procedure?", "Waarom doen we niet dat ipv dit?", "Zou deze procedure ook nog werken als we...", "Wat betekent deze variabele in je bewijs", "Waar gebruik je deze eigenschap?", ... Het perfect vanbuiten leren van een bewijs, maar niet kunnen uitleggen levert je, zoals ook herhaaldelijk gezegd tijdens de les, geen punten op. Aan de andere kant van het spectrum waren er ook studenten die geen clue hadden hoe te beginnen aan bepaald bewijs/een bepaalde vraag, maar die dan mits minimale hulp het bewijs uit hun mouw konden schudden en hier duidelijk begrip in toonden. Dit levert dan weer wel veel punten op aangezien het laat zien dat je de essentie beet hebt.

Met betrekking tot de vragen (die vanaf nu ook online beschikbaar zijn): deze waren vrij divers (over examens heen), zowel in vorm als in gedekte context, maar tijdens het gesprek (het mondeling gedeelte), werd meestal heel de cursus wel aangeraakt. Bijvoorbeeld bij een vraag over een DFA kan je wel verwachten een bijvraag te krijgen over PDAs of over Turing Machines (iets als "gaat dit ook op voor PDAs?") of over grammaticas, of over de Chomsky hierarchie.

Vraag 1

Opgave. Zet onderstaande NFA (non-deterministic finite automaton) om naar een **minimale** DFA (deterministic finite automaton), door gebruik te maken van de geziene procedures. Teken het resultaat.
Leg in elke stap uit wat je doet.



Puntenverdeling. Deze vraag werd gequoteerd op vier punten: de helft van de punten stond op de omzetting NFA naar DFA, de andere helft op minimalisatie. Deze twee delen werden onafhankelijk gequoteerd, i.e., als je een fout maakte bij de omzetting naar DFA, maar je minimalisatie juist deed voor de bekomen DFA kon je nog de maximale score op minimalisatie halen.

Bespreking. Dit was een vrij gemakkelijke vraag aangezien je om ze op te lossen gewoon de geziene procedures moest toepassen. Indien je de procedure exact volgende kwam je een DFA uit die reeds minimaal is; wat je in het tweede deel dan nog moest doen was aantonen dat deze inderdaad minimaal is, i.e., dat elke twee toestanden onderscheidbaar waren. Enkele vaak gemaakte fouten:

- Verstrooidheidsfoutjes bij het omzetten naar DFA of minimaliseren (deze fouten zorgden slechts voor een minimale puntenaftrek)
- Vergeten de state $\varnothing$ (een trap state) toe te voegen in je DFA waardoor deze niet compleet is
- De initiële state slecht kiezen ($\{q_0\}$ ipv de lambda closure van q_0).

Vraag 2

Opgave. Beschouw de taal

$$L_1 := L\left((aba)^*\right).$$

Is het complement van L_1 (i.e. $\{w \in \{a,b\}^* \mid w \notin L_1\}$) regulier? Indien ja, geef er dan een **reguliere expressie** voor en **beargumenteer** waarom jouw expressie correct is. Indien neen, toon aan met behulp van de **pompstelling**.

Puntenverdeling. Deze vraag werd verbeterd op 5 punten. Je verdiende hierbij:

- 1 punt als je een correct argument kon geven waarom de taal regulier is
- 1 punt als je de methode NFA $\rightarrow$ GTG $\rightarrow$ Regex had toegepast (al dan niet correct), i.e., als je een goede oplossingsstrategie gebruikt had
- 1 punt als je het complement van de automaat voor L_1 correct hebt berekend
- 1 punt als je ervoor gezorgd had dat je GTG een unieke eindstate had
- 1 punt als je omzetting van een GTG naar een reguliere expressie verder correct was

Bespreking. Dit was een redelijk gemakkelijke vraag. Ten eerste is het complement van een reguliere taal altijd regulier. Als je enkel dit antwoordde (eventueel impliciet) kreeg je al een vijfde van de punten die op deze vraag te verdienen waren. Om een reguliere expressie hiervoor op te stellen is de gemakkelijkste manier om eerst een DFA voor L_1 op te stellen, daar het complement van te nemen (finale en niet-finale states wisselen), die omvormen in een NFA met slechts 1 finale state, en dan de procedure met de GTG toe te passen om een reguliere expressie te bekomen. Op elk van deze stappen kon je punten verdienen. Zelfs als het uiteindelijk resultaat fout is, door ergens tussenliggend iets vergeten, kon je hier nog bijna de maximale score verdienen. Iemand die de juiste oplossingsstrategie gebruikte, maar helemaal niet tot het resultaat kwam (bijvoorbeeld wegens "geen tijd om de uiteindelijke GTG om te zetten naar een reguliere expressie"), kon al 4 van de 5 punten verdienen. Dit was dus een vraag waar gemakkelijk punten geraapt konden worden.

Enkele vaak gemaakte fouten:

- Een automaat voor L_1 opstellen zonder trap state en daar dan het complement van proberen nemen.
- Vergeten om de automaat voor L_1 een unieke finale state te geven.
- Het maken van een te grote automaat waardoor opstellen van de reguliere expressie hopeloos moeilijk wordt.
- (Geen puntenaftrek) Eindigen met een zinnetje: "Deze expressie is correct want het accepteert alle waarden die geen element zijn van L_1". Dit draagt niets bij: dat is immers gewoon de opgave herhalen. Het correct argument is: de NFA is correct, dus... de reguliere expressie ook (geziene procedure). Als je gewoon de procedure hebt toegepast, werd dit als voldoende argumentatie beschouwd. We als antwoord enkel een correcte reguliere expressie opschrijft zonder argumentatie, krijgt geen punten.

Vraag 3

Opgave. Veronderstel dat L_r een reguliere taal is.

Zijn de volgende talen over het vocabularium $\{a, b\}$ regulier? Zijn ze context-vrij? Zijn ze context-sensitief? Zijn ze (semi-)beslisbaar? Toon al jouw antwoorden aan (voor positieve antwoorden bijvoorbeeld door een grammatica of automaat van de juiste soort op te stellen of door geziene eigenschappen te gebruiken; voor negatieve antwoorden bijvoorbeeld door een pompstelling, een reductie van het halting probleem, of andere geziene eigenschappen te gebruiken).

1. $\{w \in \{a, b\}^* \mid n_a(w) = 2 * n_b(w)$ en voor elke initiele substring v van w geldt dat $n_a(v) \geqslant 2 * n_b(v)$.
   ($n_a(w)$ is het aantal a's in w en we noemen v een initiele substring van w als $w = vx$ voor een of andere $x \in \{a, b\}^*$.)
2. $\{(ab)^i b^j (ba)^{i+j} \mid i, j \in \mathbb{N}, i > j\}$
3. $\{w \mid ww^R \in L_r\}$ **(moeilijk – indien deze vraag niet meteen lukt is het aangeraden om eerst de andere vragen op te lossen)**

Puntenverdeling. Deze vraag werd gekwoteerd op 7 punten:

- 1 punt op correct gebruik chomsky hierarchie
- 2 punten op elk van de talen

Bespreking. Dit was de moeilijkste vraag van het hele oefeningenexamen. Waarschijnlijk komt de moeilijkheid van deze vraag voor een groot deel door het feit dat ze zowel inzicht als creativiteit vereist.

- Ten eerste werd er nogal veel gevraagd: over elk van die drie talen werden vijf vragen gesteld, namelijk of ze regulier zijn, context-vrij zijn,... Door gebruik te maken van de Chomsky hierarchie kon je al uitvissen dat je ten hoogste zes (twee per taal) van die vragen **echt** moest beantwoorden. Immers: elk positief resultaat propageert naar boven toe (bijvoobeeld als een taal context-vrij is, is ze uiteraard ook context-sensitief en beslisbaar en semi-beslisbaar) en elk negatief resultaat propageert naar beneden toe (bijvoorbeeld als een taal niet context-vrij is, zal ze ook niet regulire zijn). Door dit consequent te doen (1 zin per taal: "dus ze is ook (niet) ...)" kon je al een punt op deze vraag verdienen. Tot mijn verbazing hebben slechts 11 studenten dit systematisch gedaan.
- Ten tweede moest je op het zicht kunnen inschatten tot welke klasses een taal zou behoren om gericht te werk te kunnen gaan.
  - Bijvoorbeeld: bij de eerste taal kon je zien dat de opgave een variant op de haakjestaal is, maar waarbij elk sluitend haakje onmiddelijk twee openende haakjes sluit. Van deze taal weten we dat ze niet regulier is en wel context-vrij (en dus ook ...). Vervolgens kon je de pompstelling voor reguliere talen toepassen (bijvoorbeeld met de string $w = a^{2m}b^m$) en kon je een grammatica of automaat voor deze taal opstellen, bijvoorbeeld een push-down automaat die voor elke a die hij tegenkomt een a op de stack pusht, en voor elke b die hij tegenkomt, twee a's van de stack haalt.

– Voor de tweede opgave kon je zien dat deze qua structuur van de vorm $\{e^i f^j g^k\}$ is, waarbij i **twee keer** gebruikt wordt (1 keer om te vergelijken met j en 1 keer om na te gaan of $k = i+j$). Daaruit kon je vermoeden dat de taal niet context-vrij (en dus ook niet regulier) maar wel context-sensitief is. Dan kon je dan de pompstelling voor context-vrije talen toepassen (bijvoorbeeld op de string) $(ab)^{m+1}b^m(ba)^{2m+1}$ om de eerste claim hard te maken. Deze string heeft de eigenschap dat substring van lengte hoogstens m (bijvoorbeeld de substring vxy niet zowel met het ab gedeelte als het ba gedeelte kan overlappen; dit kon je uitbuiten. Tenslotte kon je bijvoorbeeld een grammatica opstellen (met dezelfde techniek als gezien voor de taal $\{a^n b^n c^n\}$) om hard te maken dat ze inderdaad context-senstief is. Een voorbeeldgrammatica is:

  Voeg een aantal (minstens 1) ab koppels toe aan het begin van de string en evenveel ba koppels achteraan:

  $S \rightarrow abSba|abAba$

  Eventueel doe je niks met A

  $A \rightarrow \lambda$

  Het alternatief is 1 keer ab, een keer b en een keer ba toe te voegen

  Laat dus A naar achter lopen voorbij alle btjes en voeg daar 1 *b, twee bas toe*

  $Ab \rightarrow bA$

  $Aba \rightarrow Bbbababa$

  Laat B terug naar voor lopen en voeg een koppel ab toe; herbegin met A

  $bB \rightarrow Bb$

  $abB \rightarrow ababA$

- De derde vraag was zoals aangegeven moeilijk. Intuitief zou je snel kunnen inzien dat deze taal context-vrij is. Immers, we weten dat er een DFA M voor L_r bestaat. We kunnen een PDA opstellen die terwijl hij w leest M simuleert en w op de stack pusht. Nadien, eens w volledig ingelezen is, zal deze met enkel lambda-transities w van de stack halen en (in omgekeerde volgorde dus) daarop M verder blijven uitvoeren. Als dit in een finale state van M terecht komt, wordt w geaccepteerd. Wie dit antwoordde kreeg 1 punt op deze vraag. Het tweede punt kon je verdienen door aan te tonen dat deze taal ook regulier is.

  Intuitief zou je waarschijnlijk denken dat deze taal niet regulier is, omdat het patroon ww^R er erg irregulier uitziet... Het antwoord is echter dat de taal **wel** regulier is. Beschouw immers een NFA voor L_r met initiele state q_0 die slechts 1 finale state q_f heeft. Als we in deze NFA alle bogen omdraaien (en initiele en finale state wisselen), krijgen we een NFA voor L^R; noem deze NFA M^R. Merk op dat M en M^R dezelfde states hebben. Wanneer je nu het product van M met M^R beschouwt (intuitief: beide automaten samen uitvoeren), kan je w inlezen. Als beide automaten na inlezen van w zich in **een zelfde** state q_i bevinden, dan zit ww^R in L. Immers, door het uitvoeren van M hebben we gezien dat $q_i \in \delta^*(q_0, w)$ en door het uitvoeren van M^R hebben we gezien dat $q_i \in (\delta^R)^*(q_f, w))$, dus dat $q_f \in \delta^*(q_i, w^R)$. Samen geeft dit dat $q_f \in \delta^R(q_0, ww^R)$.

Met betrekking tot **negatieve antwoorden**: Ik heb heel wat toepassingen van pumping lemmas gezien waarbij studenten zelf kiezen wat m is (bijvoorbeeld 2), of waarbij er een

onvolledig argument gegeven wordt waarom de gepompte string niet in de taal kan zitten (bijvoorbeeld: we veronderstellen dat y enkel uit abtjes bestaat en dan kan het niet..., zonder uit te leggen wat er gebeurt als de veronderstelling niet voldaan is, of oplossingen waarbij u, v, x, y en z vast gekozen worden).

Met betrekking tot **positieve antwoorden** heb ik heel wat constructies van automaten gezien die geponeerd worden zonder uitleg. Als die automaat simpel is, twee, max drie states, kan dit begrijpbaar zijn, maar voor een grotere automaat werkt dit niet (het is mogelijk dat je automaat juist is, maar zonder uitleg krijg je daar dan weinig punten voor).

Vraag 4

Opgave. Zet volgende grammatica in Chomsky Normal Form:

$$
\begin{aligned}
S &\to A \mid bC \mid FG \\
C &\to \lambda \mid Db \mid D \\
E &\to D \mid C \\
D &\to C \mid bB \mid \lambda \\
A &\to cA \mid bBC \\
F &\to bF \\
G &\to a \mid aG
\end{aligned}
$$

Puntenverdeling. Deze vraag werd gekwoteerd op 2 punten. 1 punt kon je verdienen door correct λ-, unit- en useless producties te verwijderen. Het tweede punt kon je verdienen door het resultaat correct om te zetten naar Chomsky normaalvorm.

Bespreking. Deze vraag was een straightforward toepassing van een gezien algoritme zonder al te veel plaatsen waar grote fouten konden gebeuren. De oplossing is ook behoorlijk kort. Het is dan ook niet verbazingwekkend dat er hier erg hoog op gescoord werd.

Sommige fouten die af en toe terugkwamen:

- Lambda en unit producties in de verkeerde volgorde verwijderen waardoor het resultaat toch nog unit producties bevat.
- Variabele B, die voor geen enkele productie aan de linkerkant staat, op een exotische manier gebruiken in de plaats van deze als useless te beschouwen (bijvoorbeeld door $B \to \lambda$ als productie toe te voegen aan de grammatica).
- Bij het verwijderen van useless variabelen deze gewoon uit de productie halen in de plaats van de productie te verwijderen (bijvoorbeeld als F useless is $A \to FG$ vervangen door $A \to G$)
- Een grammatica van de vorm

  $$
  \begin{aligned}
  A &\to Bc \\
  C &\to c
  \end{aligned}
  $$

  omzetten naar

  $$
  \begin{aligned}
  A &\to BV_c \\
  C &\to V_c \\
  V_c &\to c
  \end{aligned}
  $$

  waardoor het resultaat niet in Chomsky normaalvorm is (unit producties)

Vraag 5

Opgave. Gegeven is een **2-track** Turing Machine **M** die zich als volgt gedraagt.

- Deze machine verwacht een string die enkel bestaat uit een aantal '1' symbolen (x in unaire notatie) op de tweede track en verwacht dat de read-write head aan het begin van deze expressie staat.
- Als aan de bovenstaande verwachting voldaan is, beweegt de read-write head zich x^2 stappen naar rechts.
- Als er niet aan de verwachting voldaan is, doet deze machine niets (i.e., ze laat beide tracks en de positie van de read-write head ongewijzigd).

Het gedrag van **M** wordt hieronder geillustreerd:

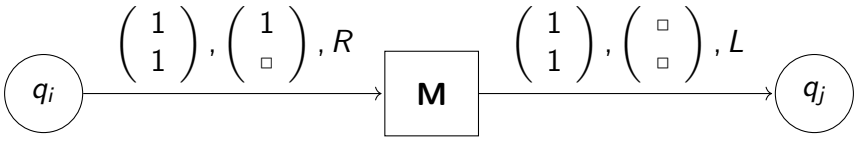
Voor:

	...	0	1	2	3	4	5	6	7	8	9	...
track 1	□	1	1	1	1	1	1	1	□	□	□	□
track 2	□	□	1	1	□	□	□	□	□	□	□	□
			△									

Na:

	...	0	1	2	3	4	5	6	7	8	9	...
track 1	□	1	1	1	1	1	1	1	□	□	□	□
track 2	□	□	1	1	□	□	□	□	□	□	□	□
							△					

Maak gebruik van **M** (enkel van **M**, niet van andere machines gezien in de les) om een **deterministische 2-track** Turing machine **SQ** te maken, die voor een natuurlijk getal $x > 0$ (dat op de eerste track staat en met lege tweede track), eindigt met x op de eerste track en $\sqrt{x}$ op de tweede track indien $\sqrt{x}$ zelf een natuurlijk getal is, en in die niet eindigt als $\sqrt{x}$ geen natuurlijk getal is.
Maak gebruik van **M** in **SQ**, als aangeduid op het volgende voorbeeld:



Geef uitgebreide uitleg over de werking van je Turing Machine. Zonder uitleg krijg je geen punten op deze vraag!

Puntenverdeling. Deze opgave werd gekwoteerd op 4 punten. De heft van de punten kon je verdienen door een hoogniveaubeschrijving van je Turingmachine te geven. De andere helft door effectief een correct machine op te stellen.

Bespreking. Deze vraag was op te lossen met een behoorlijk kleine Turing machine. Intuitief schrijft deze machine eerst een 1 op zijn tweede track, loopt naar het begin van de input en voert M uit. Er zijn drie mogelijkheden:

- De read-write head wijst (op de eerste track) naar een 1: in dat geval is het getal op de eerste track groter dan het kwadraat van het getal op de tweede track. In dat geval schrijven we een extra 1 op de tweede track en herbeginnen we.

- De read-write head wijst naar de eerste blank op de eerste track. In dat geval staat er nu $\sqrt{x}$ op de tweede track.
- De read-write head wijst naar een andere blank. In dat geval is het getal op de eerste track geen kwadraat.

Deze techniek (een getal gokken en dan een checker uitvoeren) hebben we ook in de les gezien.

We merkten dat de meeste studenten die het hoogniveau idee (hierboven beschreven) juist hadden, dit ook correct konden omzetten naar een Turing Machine. Er is uiteraard een groep studenten die dit idee niet inzag en niet tot een correcte TM kwam. Enkele veelgemaakte fouten:

- Sommige studenten "inspecteren de internals van M om een machine M^{-1} op te stellen die de vierkantswortel berekent". Daar krijg je geen punten voor. M is een black box. Een soort inverse van M opstellen was de opgave; je mag geen magische methodes gebruiken om die gratis te krijgen.
- Een aantal studenten maakte een off-by-one error door te checken of de read-write head na uitvoering van M op de laatste 1 ipv op de eerste blank staat. Dit resulteerde slechts in zeer kleine puntenaftrek.
- Een aantal studenten heeft letterlijk de transities gebruikt in het syntax-voorbeeld om M toe te passen (bijvoorbeeld: ze doen altijd een move naar rechts vlak voor M wordt opgeroepen), terwijl dat maar een voorbeeld was. Hier werden geen punten voor afgetrokken