

Machines en Berekenbaarheid

 tuyaux.winak.be/index.php/Machines_en_Berekenbaarheid

Machines en Berekenbaarheid

Richting Informatica, Wiskunde

Jaar 2BINF, Bachelor Wiskunde Keuzevakken

Bespreking

Dit vak is het vervolg op het vak Talen en automaten uit 1e Bachelor. De wijze van lesgeven en de beoordelingscriteria blijven dan ook analoog hiermee. Let er echter wel op dat dit vak wat moeilijker is dan Talen en automaten en dat ook het oplossen van de vragen in de verslagen wat lastiger is. Probeer deze verslagen toch echter goed te doen aangezien deze je punten te goede kunnen komen. Ook is het groepswork nu wat makkelijker doordat je medestudenten minder neiging hebben om af te haken.

Op het examen, probeer zo snel genoeg te schrijven aangezien de tijd beperkt is. Let ook op dat je de moeilijkere bewijzen ook leert aangezien deze veel kans hebben om gevraagd te worden. Voor de praktijk mag je opnieuw je boek gebruiken.

Puntenverdeling

Groepswork

Theorie: 5/20. Praktijk: 5/20. Groepswork: 10/20.

Zonder groepswork

Theorie: 10/20. Praktijk: 10/20.

Examenvragen

Academiejaar 2019 - 2020 (1ste zittijd)

Oefeningenexamen

1. De taal $\{ xryszt | r \geq 0, t \geq 0, s = 3 \cdot (r+t) \}$ is contextvrij. Toon dit aan door een contextvrije grammatica voor deze taal op te stellen.
2. Voer de stappen uit om volgende grammatica om te zetten in CNF (P is startsymbool). Geef daarbij de unit pairs, generating, reachable en nullable symbolen. Stappen 2 en 3 van het CNF algoritme werden niet gevraagd (enkel clean up dus)
 - $P \rightarrow il|dl|WP \rightarrow il|dl|W$
 - $I \rightarrow x|y|z|\epsilon \mid I \rightarrow x|y|z|\epsilon$
 - $W \rightarrow wIPoW \rightarrow wIPo$
3. CYK toepassen voor string aabba op grammatica. Geef resulterende tabel.
 - $S \rightarrow AB|BCS \rightarrow AB|BC$
 - $A \rightarrow BA|aA \rightarrow BA|a$
 - $B \rightarrow CC|bB \rightarrow CC|b$
 - $C \rightarrow AB|aC \rightarrow AB|a$

4. Pas de LR(1) parse table hieronder toe op de string xxxyyy. Geef de stackinhoud en de resulterende string.

	x	y	EOS	S	X	Y
1	shift 2		$X \rightarrow \lambda X \rightarrow \lambda$	9	8	7
2	shift 2	shift 6	$X \rightarrow x X \rightarrow x$		3	4
3			$X \rightarrow x X X \rightarrow x X$			
4		shift 5				
5		$Y \rightarrow x Y y Y \rightarrow x Y y$	$Y \rightarrow x Y y Y \rightarrow x Y y$			
6		$Y \rightarrow x y Y \rightarrow x y$	$Y \rightarrow x y Y \rightarrow x y$			
7			$S \rightarrow Y S \rightarrow Y$			
8			$S \rightarrow X S \rightarrow X$			
9			accept			

Individuele verdediging

Hieronder kan je per algoritme de inputs terugvinden die je moest inlezen op de individuele verdediging.

input-cfg2pda1.json

input-cfg2pda2

input-cfg2pda3.json

input-cnf1.json

input-cnf2.json

input-cyk1.json

input-cyk2.json

input-ll1.json

input-ll2.json

input-lr1.json

input-lr2.json

input-multitm1.json

input-multitm2.json

input-pda2cfg1.json

input-pda2cfg2.json

input-tm1.json

input-tm2.json

input-tmsubroutines.txt

readme.txt

subroutines.txt

todot.sh

Academiejahr 2018 - 2019 (2de zittijd)

1. De grammatica voor $\{ arbsct|r \geq 0, s \geq 0, (t-s)/2 = r \mid arbsct|r \geq 0, s \geq 0, (t-s)/2 = r \}$ wordt onvolledig gegeven. Vul het body van symbool X aan.
 - $S \rightarrow XS \rightarrow X$
 - $X \rightarrow \dots X \rightarrow \dots$
 - $Y \rightarrow \epsilon \mid bYc \mid Y \rightarrow \epsilon \mid bYc$
2. Voer de stappen uit om volgende grammatica om te zetten in CNF (L is startsymbool). Geef daarbij de unit pairs, generating, reachable en nullable symbolen. Stappen 2 en 3 van het CNF algoritme werden niet gevraagd (enkel clean up dus)
 - $L \rightarrow aFPb \mid F \mid \epsilon \mid L \rightarrow aFPb \mid F \mid \epsilon$
 - $F \rightarrow r \mid kF \rightarrow r \mid k$
 - $P \rightarrow LsP \mid k \mid LP \rightarrow LsP \mid k \mid L$
3. CYK toepassen voor string accbb op grammatica. Geef resulterende tabel.
 - $S \rightarrow ABS \rightarrow AB$
 - $A \rightarrow a \mid BC \mid ABA \rightarrow a \mid BC \mid AB$
 - $B \rightarrow b \mid AC \mid BCB \rightarrow b \mid AC \mid BC$
 - $C \rightarrow c \mid a \mid ABC \rightarrow c \mid a \mid AB$
4. LR(1) parse table uit de cursus (extra bladen p123 denk ik) toepassen op string zzzz. Geef stackinhoud en resulterende string.

Academiejahr 2015 - 2016 - 2de zittijd

Theorie

Voor dit deel van het examen moest je maar 5 van de 6 vragen beantwoorden. Je kan dus m.a.w. 1 vraag waar je niet zeker van bent openlaten en hier worden geen punten voor afgetrokken.

1. Geef de formele definitie van $N(P)$, de taal aanvaard door een PDA met lege stack.
2. Stel $G = (V, T, P, S)$ een CFG, A in V en w een terminal string. Geef 4 equivalenties met: 'er bestaat een parse tree G met wortel (root) A en yield w '.
3. Geef en bewijs het theorema over het bestaan van een parse tree indien er een afleiding met recursieve inferentie is.
4.
 - Geef de formele definitie van een Turing machine.
 - Wat maakt een Turing machine krachtiger dan een pushdown automaat?
 - Hoe zou je pushdown automaten (zonder tape en met behoud van het stackconcept en zijn werking) kunnen uitbreiden zodat een klasse van automaten ontstaan die dezelfde talen aanvaardt als de klasse van Turing machines?
5. Wat is het verschil tussen ambigu en inherent ambigu?
6. Formuleer twee eigenschappen (theorema's) over het verband tussen recursieve functies en programmeertalen (zonder bewijs).

Academiejahr 2015 - 2016 - 1ste zittijd

Theorie

Voor dit deel van het examen moest je maar 6 van de 7 vragen beantwoorden. Je kan dus m.a.w. 1 vraag waar je niet zeker van bent openlaten en hier worden geen punten voor afgetrokken.

1. Geef de definitie van "ambigu" en "inherent ambigu"
2. Bij de conversie van final-state PDA naar equivalente empty-stack PDA en omgekeerd wordt er telkens een nieuw stacksymbool toegevoegd. Is da in beide gevallen noodzakelijk? Motiveer je antwoord.
3. Gegeven een CFG $G = (V, T, Q, S)$. Geef de constructie van de equivalente PDA P en bewijs dat als $w \in L(G) \Rightarrow w \in N(P)$.
4. Gegeven een PDA. Beschouw de equivalente CFG.
 1. Geef de grammaticaregels die overeenkomen met $(r, Y_1Y_2...Y_k) \in \delta(p, a, X)$.
 2. Leg de betekenis van zo'n regel uit.
 3. Hoeveel regels zijn er zo?
5. Welke talen kan een LR(k) parser parsen?
6. Een niet deterministische TM kan gesimuleerd worden met een deterministische TM. Leg kort uit hoe de huidige ID verwerkt wordt door de DTM.
7. Geef in detail (zonder bewijzen) de verbanden tussen TM, partieel recursieve functies en programmeertalen.

Oefeningen

Het oefeningen examen was multiple-choice (4 keuzes). Voor elk juist antwoord kreeg je 1 punt, voor elk fout antwoord -0.5, liet je de vraag open kreeg je 0 punten. Als de opties niet gegeven zijn, betekent dit dat ik ze mij niet meer herinner. (Of dat het lange opties zijn en ik ze niet wil uittypen.)

1. Welke CFG genereert strings uit de taal van de reguliere expressie $"(a+b+c)^*ab(f+g)c^*" ?$
2. Gegeven een CFG G en vier stellingen. Welke stelling is juist?
3. Beschouw de CFG G met volgende productie regels:

- $S \rightarrow bAbA \mid B \mid Cab$
- $A \rightarrow a \mid Dba \mid \epsilon$
- $B \rightarrow bb \mid A$
- $D \rightarrow ab \mid ba$

Hoeveel nullable symbols zijn er?

4. Welke ontbinding in CNF van de CFG in de vorige oefening is correct?
5. Gegeven een empty-stack PDA P met volgende transities:

- $\delta(q_0, a, Z) = (q_1, aZ)$
- $\delta(q_1, b, a) = (q_2, ba)$
- $\delta(q_2, c, a) = (q_2, \epsilon)$
- $\delta(q_2, c, b) = (q_2, \epsilon)$
- $\delta(q_2, \epsilon, Z) = (q_3, \epsilon)$

Welke productions heeft variabele $[q_0Zq_1]$?

6. Hoeveel extra states moet je minimum toevoegen om de PDA P uit de vorige oefening om te zetten naar een final-state PDA?
7. Je krijgt een LR(1) parse table, en de inhoud van de stack op een gegeven moment. De vraag is welke input er reeds geparset is op dat moment.

8. Beschouw volgende grammatica

- $A \rightarrow aBa \mid abBa$
- $B \rightarrow bb \mid bbc$

Hoeveel lookahead symbolen heeft een LL parser nodig om deze te parsen?

9. Er wordt een TM gegeven, jij moet zeggen wat deze TM berekent.

10. Je krijgt een transitie diagram van een DTM en een input string. Gevraagd is in hoeveel transities deze DTM de string aanvaardt.

Academiejahr 2014 - 2015 - 1ste zittijd

Theorie

1. Geef *Basis*- en *Inductiegeval* van de definitie van recursieve inferentie.
2. Zij PFPF een *Final State PDA*
 - Geef de constructie van een equivalente *Empty Stack PDA* PNP.
 - Waarom worden er nieuwe *stacksymbolen* toegevoegd?
 - Geef het overeenkomstige theorema.
 - Waar zou het bewijs fout lopen zonder de nieuwe stacksymbolen?
3. Zij gegeven een CFG $G=(V,T,Q,S)$ $G=(V,T,Q,S)$. Leg de werking van de equivalente *PDA* uit als $(p,w,\theta)(p,w,\theta)$ de huidige *ID* is, met $\theta\theta$ een niet-lege tail.
4. Zij gegeven een *PDA* en de equivalente *CFG*.
 - Geef de grammaticaregels die overeenkomen met $(r,Y_1Y_2\dots Y_k) \in \delta(p,a,X)$
 - $(r,Y_1Y_2\dots Y_k) \in \delta(p,a,X)$.
 - Wat betekent zo'n regel?
 - Hoeveel van deze regels zijn er?
5. Zij gegeven een *DPDA* en stel $(q,w,X) \vdash (p,\epsilon,\epsilon)(q,w,X) \vdash (p,\epsilon,\epsilon)$
 - Wat weet je over de overeenkomstige meest-linkse afleidingen in de equivalente *CFG*.
 - Bewijs.
6. Hoe kan je een *k-tape* Turing Machine simuleren met een *1-tape* Turing Machine.
7. Geef (zonder bewijs) de verbanden betreffend berekenbaarheid tussen Turing Machines, partieel recursieve functies en programmeertalen.

Academiejahr 2013 - 2014 - 1ste zittijd

Theorie

1. Indien je voor het onderdeel koos: kies een eigenschap i.v.m. Turing machines en recursieve functies en bewijs deze eigenschap.
2. Geef de equivalenties tussen inferenties, afleidingen en parse trees formeel weer voor:
 - een terminal string ww
 - een string ww die variabelen bevat.
3. Beschouw volgend theorema uit de cursus: if $L=N(P)L=N(P)$ for some DPDA P , then L has an unambiguous CFG.
 - *Proof*: By inspecting the proof of theorem 6.4 (voor elke PDA P bestaat er een CFG G zodat $L(G)=N(P)L(G)=N(P)$ we see that if the construction is applied to a DPDA, the result is a CFG with unique leftmost derivations.
 - Het bewijs van theorema 6.4 bestaat uit twee delen. Geef het deel van het bewijs waarover het hier gaat. Laat duidelijk zien waar het verschil zit tussen een DPDA en een PDA, m.a.w. waarom is er niet noodzakelijk een unieke links-meeste afleiding als de PDA geen DPDA is?
4. Bewijs dat $L=\{0^n1^n2^n \mid n \geq 1\}$ geen context-vrije taal is. Leg uit hoe een TM voor deze taal geconstrueerd kan worden (je hoeft de TM niet in detail te bespreken).

5. Stel $P = (\{p, q, r\}, \{0, 1\}, \{X, Z0\}, \delta, q, Z0)$ $P = (\{p, q, r\}, \{0, 1\}, \{X, Z0\}, \delta, q, Z0)$ en de definitie van δ bevat o.a.:
- $\delta(q, 1, Z0) = \{(q, XZ0)\}$ $\delta(q, 1, Z0) = \{(q, XZ0)\}$
 - $\delta(q, \epsilon, X) = \{(q, \epsilon)\}$ $\delta(q, \epsilon, X) = \{(q, \epsilon)\}$
 - $\delta(p, 1, X) = \{(p, \epsilon)\}$ $\delta(p, 1, X) = \{(p, \epsilon)\}$
 - Stel G is de overeenkomstige grammatica volgens de constructie uit de cursus. Schrijf de regels in G die overeenkomen met bovenstaande transities volledig uit.
6. Geef de definitie van Chomsky normaal vorm (niet de constructie) en beschrijf kort een toepassing ervan.

Oefeningen

1. Toon aan dat de taal $\{x^r y^s z^t \mid r \geq 0, s \geq 0, t = 2(r+s)\}$ context vrij is.
2. Stel de Turing machine op die de vorige taal aanvaardt en stel de transities grafisch voor.
3. Gegeven volgend grammatica G met startsymbool U :

```

U -> V|W
W -> XY
V -> CD
Y -> bYc|E|cD
C -> aCb|aX|E
X -> aX|epsilon
D -> cD|epsilon
E -> bE|b

```

1.
 - beschrijf de taal $L(G)$.
 - geef de PDA P die de taal aanvaardt bij empty stack gebruik makend van het geziene constructie algoritme. Geef de formele definitie van P , waarbij je de transitiefunctie grafisch voorstelt.
 - Is de PDA deterministisch? Leg uit.
 - Geef de opeenvolging van ID's waarmee je aantoont dat de string $abbcc$ in $L(P)$ zit.
 - Stel voor deze CFG de PDA op waarop het principe van LR parsing is gebaseerd. (Indien je voor dit studiedeel koos)

Zet volgende grammatica om in Chomsky Normal Form. Voer eerst de nodige vereenvoudigingen uit. (Useless symbols, ϵ -producties, unit productions elimineren) Geef na elke vereenvoudiging duidelijk het overeenkomstig resultaat weer. Het startsymbool is L .

```

L -> aFPb|F|epsilon
P -> LsP|k|L
F -> r|k

```

Academiejahr 2010 - 2011 - 1ste zittijd

Theorie

1. Voor grammatica's hebben we inferenties, afleidingen en parse trees gedefinieerd. Welke equivalenties bestaan er tussen deze concepten?
2. We hebben het zeer uitvoerig gehad over volgend theorema: *Gegeven een PDA P , dan bestaat er een context-vrije grammatica G zodat $L(G) = N(P)$.* Geef (a) constructie en (b) bewijs. (c) Pas de constructie toe op $P = (\{p, q\}, \{0, 1\}, \{X, Z0\}, \delta, q, Z0)$ met $\delta(q, 0, Z0) = \{(p, XZ0)\}$, $\delta(q, \epsilon, Z0) = \{(q, \epsilon)\}$, $\delta(q, 1, Z0) = \{(p, \epsilon)\}$ $P = (\{p, q\}, \{0, 1\}, \{X, Z0\}, \delta, q, Z0)$ met $\delta(q, 0, Z0) = \{(p, XZ0)\}$, $\delta(q, \epsilon, Z0) = \{(q, \epsilon)\}$, $\delta(q, 1, Z0) = \{(p, \epsilon)\}$

3. (a) Toon aan dat voor context-vrije talen het inverse pumping lemma uit het pumping lemma verkregen wordt door contrapositie, d.i. $(H \Rightarrow C) \Leftrightarrow (\neg C \Rightarrow \neg H)$ (b) Wat kunnen we over de taal $\{x^n y^n z^n | n \geq 1\}$ bewijzen met het inverse pumping lemma? Toon aan. En over $\{x^n y^n | n \geq 1\}$?
4. (a) Geef de definitie van een ID van een Turing Machine. (b) Hoe wordt de 'move' $\delta(q, X_i) = (p, Z, L)$ vertaald naar ID's?

Praktijk

1. Zet de grammatica met volgende regels om in Chomsky Normaal Vorm, maar voer eerste de nodige vereenvoudigingen uit (epsilon-producties, unit-producties en useless symbols elimineren). Geef na elke vereenvoudiging duidelijk het overeenkomstig resultaat weer. Het startsymbool is F.

```

F -> N | P | E
E -> PeN | I
P -> NpN | I
N -> dNc | epsilon
I -> NI | epsilon

```

2. Ontwerp een DPDA die het type XML-document aanvaardt zoals in figuur 1 is weergegeven.

1. In je DPDA stel je xml-tags voor door symbolen. Het symbool bestaat uit de eerste letter van de tag-naam. Een sluitingstag duid je aan dmv een hoedje. Bijvoorbeeld: voor `<rss>` is het symbool voor de openingstag `r` en voor de sluitingstag `</rss>` is dit `?`. (hoedje op `r`).
2. De tekst binnen de tags mag voorgesteld worden door het symbool `x`. De `<link>` tag is optioneel. Er moet minstens 1 item zijn en juist 1 channel.

```

<rss> <channel> <title>Machines en berekenbaarheid</title>
<link>http://www.ua.ac.be/cjomski</link> <item> <title>Oefeningen reeks 1</title>
<link>http://blackboard.ua.ac.be/503</link> </item> <item> <title>Oefeningen reeks 2</title>
<link>http://blackboard.ua.ac.be/404</link> </item> <item> <title>Examen 2de Zit
2008</title> </item> </channel> </rss>

```

3. Ontwerp een standaard TM (geen extensies) door middel van een transitiediagram die enkel en alleen fibonaccistrings aanvaardt, bv. 010101101110111101111110 en 010101101110. Een fibonaccistring is een reeks van opeenvolgingen van 1-en gescheiden door 0-en zodat het aantal opeenvolgende 1-en altijd de som is van het aantal 1-en in de vorige 2 opeenvolgingen van 1-en. Een fibonaccistring begint altijd met 0101 en eindigt altijd met een 0.

Academiejahr 2009 - 2010 - 2de zitting

Theorie

1. Geef een overzicht van alle sluitingseigenschappen die we gezien hebben bij CFL
2. Bij het converteren van een CFG naar chomsky normal form is de volgorde van bewerkingen belangrijk. Toon dit aan met een voorbeeld
3. Def DPDA (Deterministische !!)
4. Gegeven 2 verschillende parse trees die naar een zelfde string `w` gaan. Wat weet je dan over de Left-Most derivations? en bewijs dit

Praktijk

1. Maak een Turing Machine voor volgende taal $L = \{a^m b^n c^m | m, n > 0\}$
2. Is $\{xaybzb | a < b < c\}$ context vrij? Toon dit aan.

3. Zet volgende grammar in chomsky normal form.

$L \rightarrow () \mid (FP) \mid (P)$
 $F \rightarrow a \mid d \mid \text{epsilon}$
 $P \rightarrow asP \mid LsP \mid a \mid L \mid \text{epsilon}$

Academiejahr 2008 - 2009

Theorie

1. Definitie van een DPDA
2. Geef het theorema van het bewijs in bijlage (p 245 en 246)
3. Op p 246 onder die figuur leg die Inductiehypothese uit. (bijlage 245 en 246 was gegeven)
4. Voor alle PDA P bestaat er een CFG G zodat $N(P)=L(G)$. Geef de constructie weer en bespreek de betekenis van de verschillende soorten productieregels in de geconstateerde grammatica.
5. Kan de syntax van elke programmeertaal worden voorgesteld door CFG (+ Motiveer).
6. Bespreek de simulatie van een computer door een Turingmachine.

Praktijk

1. Ontwerp een one-tape Turingmachine (zonder extensies) die gegeven een string uit $\{0,1,2\}^*$ * die de verschillende karakters groepeerd en sorteert op basis van hun aantal (van klein naar groot). Indien de aantallen gelijk zijn worden ze lexicologisch geordend. Geef hiervan het grafisch transitiediagramma weer.

vb: 2012210101212001120 = 000000222222111111

2. Toon aan dat volgende 2 grammatica's hetzelfde zijn.

1. $S \rightarrow UAU \rightarrow rUcA|u| \epsilon A \rightarrow UiA|UfsiA|a| \epsilon$
 $S \rightarrow UAU \rightarrow rUcA|u| \epsilon A \rightarrow UiA|UfsiA|a| \epsilon$

2. $S \rightarrow AU|R| \epsilon U \rightarrow RU|ruC|u| \epsilon A \rightarrow RU|AiU|FU|a| \epsilon R \rightarrow rACC \rightarrow cF \rightarrow AfsiUI \rightarrow RuFA| \epsilon$
 $S \rightarrow AU|R| \epsilon U \rightarrow RU|ruC|u| \epsilon A \rightarrow RU|AiU|FU|a| \epsilon R \rightarrow rACC \rightarrow cF \rightarrow AfsiUI \rightarrow RuFA| \epsilon$

Er was een klein catch aangezien ze niet helemaal gelijk zijn.

3. Is $\{xaybzc|a < b < c\}$ context vrij of niet? Toon aan.
4. Bewijs dat de taal $\{a^{2(r+t)}brct|r, t > 0\}$ voldoet aan de prefix eigenschap.

Academiejahr 2007 - 2008

Theorie

1. Geef de definitie van een TM (inclusief transitiefunctie).
2. Geef een constructie om een PDA te contrueren die dezelfde taal genereert als een gegeven CFG.
3. Gegeven een PDA P_f , bewijs dat er een PDA P_n bestaat die by empty stack eenzelfde taal accepteert als P_f by final state.
4. Van welke theoretische aspecten is yacc een toepassing? Bespreek uitvoerig.

Praktijk

1. Ontwerp een standaard TM (geen extenties) door middel van een transitiediagram die enkel en alleen fibonaccistrings aanvaardt, bv. 0101011011101111011111110 en 010101101110. Een fibonaccistring is een reeks van opeenvolgingen van 1-en gescheiden door 0-en zodat het aantal opeenvolgende 1-en altijd de som is van het aantal 1-en in de vorige 2 opeenvolgingen van 1-en. Een fibonaccistring begint altijd met 0101 en eindigt altijd met een 0.
2. Zet de volgende grammatica om in Chomsky normaal vorm, maar voer eerste de nodige vereenvoudigingen uit (epsilon-producties, unit producties en useless symbols elimineren). Geef na elke vereenvoudiging duidelijk het overeenkomstige resultaat weer. Het startsymbool is C.

| C → jomSKI S X | ε → I IC | ε → K X X x

3. Ontwerp een DPDA die het type XML-document aanvaardt zoals in figuur 1 is weergegeven.
 1. In je DPDA stel je xml-tags voor door symbolen. Het symbool bestaat uit de eerste letter van de tag-naam. Een sluitingstag duid je aan door een hoedje. Bijvoorbeeld: voor <rss> is het symbool voor de openingstag r en voor de sluitingstag </rss> is dit ?. (hoedje op r).
 2. De tekst binnen de tags mag voorgesteld worden door het symbool x. De <link> tag is optioneel. Er moet minstens 1 item zijn en juist 1 channel.

```
<rss> <channel> <title>Machines en berekenbaarheid</title>
<link>http://www.ua.ac.be/cjomski</link> <item> <title>Oefeningen reeks 1</title>
<link>http://blackboard.ua.ac.be/503</link> </item> <item> <title>Oefeningen reeks 2</title>
<link>http://blackboard.ua.ac.be/404</link> </item> <item> <title>Examen 2de Zit
2008</title> </item> </channel> </rss>
```