AUTOMATA & COMPUTABILITY

JENSEN BERNARD

INHOUDSOPGAVE

ABSTRACT

Dit document is tot stand gekomen om een hulp te zijn bij het studeren van het vak **Automaten en Berekenbaarheid**. Het bevat enkel voorbeelden van examenvragen, samen met een uitgewerkte oplossing. In geen enkel geval kan dit document de cursus vervangen. Ik raad aan om de cursus, die verkrijgbaar is via de cursusdienst van **Scientica**, eerst door te nemen om op zijn minst te weten waarover het gaat. Indien u fouten vindt of informatie wil toevoegen, kan dit altijd via Github. Good luck*.

* *You'll need it.*

1 VOORWOORD

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2 TALEN EN AUTOMATEN

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

3 TALEN EN BEREKENBAARHEID

3.1 A_{TM} is niet-beslisbaar

Bewijs in detail dat A_{TM} niet beslisbaar is en steun daarbij niet op de stelling van Rice. Zou het helpen als het toegelaten was op de stelling van Rice te steunen? Is A_{TM} herkenbaar? Co-herkenbaar?

A_{TM} is niet beslisbaar

Bewijs. Beter bekend als het acceptatieprobleem voor Turingmachines. De geassocieerde taal is

$$A_{TM} = \{< M, s > \mid M \textit{ is een Turingmachine en } s \in L_M\}$$

Stel er bestaat een beslisser B voor A_{TM}. Dat betekent dat bij input $< M, s >$ B accepteert als M bij input s stopt in zijn q_a en verwerpt als M bij input s stopt in zijn q_r of loopt. We schrijven

$$B(< M, s >) \textit{ is accept als } M\ s \textit{ accepteert en anders reject}$$

Construeer nu de contradictie machine C met eigenschap:

$$C(< M >) = \mathtt{opposite}(B(< M, M >)) \textit{ voor elke Turingmachine } M$$

Daarbij is $\mathtt{opposite}(\mathtt{accept}) = \mathtt{reject}$ en $\mathtt{opposite}(\mathtt{reject}) = \mathtt{accept}$.
Neem nu voor M hierboven C zelf, dan krijgen we:

$$C(< C >) = \mathtt{opposite}(B(< C, C >))$$

Als $C(< C >) = \mathtt{accept}$, dan is $B(< C, C >) = \mathtt{accept}$, dan is $\mathtt{opposite}(B(< C, C >)) = \mathtt{reject}$, dan is $C(< C >) = \mathtt{reject}$, dan is $B(< C, C >) = \mathtt{reject}$, dan is $\mathtt{opposite}(B(< C, C >)) = \mathtt{accept}$, dan is $C(< C >) = \mathtt{accept}$ ...
Dus C kan niet bestaan, dus B bestaat niet. Dus A_{TM} is niet beslisbaar. □

A_{TM} is herkenbaar

Bewijs. De herkenner A voor A_{TM} laat, met input $< M, x >$, M lopen op s. Indien deze M s accepteerd, dan accepteerd A zijn input. Indien deze de input $\mathtt{reject}$, of gewoon niet stopt, dan zal A deze ook niet accepteren. A_{TM} is dus herkenbaar. □

A_{TM} is niet co-herkenbaar

Bewijs. A_{TM} kan echter niet co-herkenbaar zijn. We bewijzen dit met contradictie. Indien A_{TM} co-herkenbaar is, is deze dus herkenbaar en co-herkenbaar (zie vorig bewijs). Wanneer een taal deze beide eigenschappen bezit, is deze beslisbaar. Dit is een contradictie met het eerste bewijs. □

De stelling van Rice

Deze stelling bewijst dat elke taal (die aan bepaalde voorwaarden voldoet) niet-beslisbaar is. Hier gaan we zo dadelijk dieper op in. We moeten eerst de volgende twee definities toelichten alvorens we verder kunnen gaan met het bewijs van A_{TM}, gebruik makend van deze stelling (wat korter zal zijn dan voorgaande bewijzen).

Theorem 1 (De niet-triviale eigenschap)**.** *Een eigenschap* P *van Turingmachines heet niet-triviaal indien* $\mathtt{Pos}_P \neq \emptyset$ *en ook* $\mathtt{Neg}_P \neq \emptyset$.

Theorem 2 (Taal-invariante eigenschap)**.** *Een eigenschap* P *heet taal-invariant indien alle machines die dezelfde taal bepalen, hebben ofwel allemaal* P*, ofwel heeft geen enkele ervan* P*.*

$$L_{M_1} = L_{M_2} \Rightarrow P(M_1) = P(M_2)$$

Theorem 3 (Formelere stelling van Rice)**.** *Voor elke niet-triviale, taal-invariante eigenschap* P *van Turingmachines geldt dat* Pos_P *(en ook* Neg_P*) niet beslisbaar is.*

Bewijs. Veronderstel dat $M_\emptyset$ (de machine die de lege taal beslist) de eigenschap P niet heeft - indien dat niet zo is, verander dan P in zijn negatie. Vermits P niet-triviaal is, bestaat er een taal L_X zodat X een Turingmachine is met de eigenschap P. Stel dat Pos_P (en dus ook Neg_P) beslisbaar id: we zullen een beslisser B voor Pos_P gebruiken om een beslisser A te maken voor A_{TM}. A krijgt als input $< M, s >$ en doet het volgende:

1. construeer een hulpmachine $H_{M,s}$ die het volgende doet bij input x:
   a) laat M lopen op s
   b) indien M s accepteert, laat dan X lopen op x en accepteer als X x accepteert
2. laat nu B los op $H_{M,s}$
3. als B $H_{M,s}$ accepteert, dan $\mathtt{accept}$, anders $\mathtt{reject}$

$H_{M,s}$ accepteert ofwel de lege taal, ofwel L_X (TODO: waarom).

$$\begin{gathered} A \textit{ accepteert } < M, s > \\ \Updownarrow \\ B\ H_{M,s} \textit{ accepteert} \\ \Updownarrow \\ H_{M,s} \textit{ de eigenschap } P \textit{ heeft} \\ \Updownarrow \\ H_{M,s} \textit{ accepteert } L_X \\ \Updownarrow \\ M \textit{ accepteert } s \end{gathered}$$

Dus, A is een beslisser voor A_{TM}, hetgeen niet kan, dus kan B niet bestaan en Pos_P is niet beslisbaar. □

3.2 Reduceerbaarheid

Bespreek de twee noties van reduceerbaarheid ($A \leqslant_m B$ en $A \leqslant_T B$), hun verband en op welke manier die noties kunnen gebruikt worden om aan te tonen dat een taal (on)beslisbaar/herkenbaar is.

Veel-één reductie

Om over te gaan naar de definitie van de reductie van talen, kunnen we best eerst de definifie van Turing-berekenbaar erbij halen (indien we dit niet doen, kunnen we zeker zijn van deze bijvraag).

Theorem 4 (Turing-berekenbare functie)**.** *Een functie f heet Turing-berekenbaar indien er een Turingmachine bestaat die bij input s uiteindelijk stopt met $f(s)$ op de band.*

Theorem 5. *We zeggen dat taal L_1 (over Σ_1) naar taal L_2 (over Σ_2) kan gereduceerd worden indien er een afbeelding f met signatuur $\Sigma_1^* \longrightarrow \Sigma_2^*$ bestaat zodanig dat $f(L_1) \subseteq L_2$ en $f(\overline{L_1}) \subseteq \overline{L_2}$, en zodanig dat f Turing-berekenbaar is. We noteren dat door $L_1 \leqslant_m L_2$.*

4 HERSCHRIJFSYSTEMEN

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

5 ANDERE REKENPARADIGMA'S

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

6 TALEN EN COMPLEXITEIT

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

7 BEWIJZEN ONDER DE LOOP

7.1 Overzicht

In dit hoofdstuk zullen bewijzen verder uitgelegd (of vereenvoudigd) worden. Deze bewijzen worden gebruikt in de oplossingen van de examens, maar zijn soms moeilijk te begrijpen.

De modeloplossingen die doorheen dit document werden beschreven, bevatten soms moeilijke en complexe bewijzen. Dit hoofdstuk is bedoeld om deze volledig te beheersen. Probeer wel de vorige notaties te gebruiken voor het examen, aangezien dit de structuur is die de professor zelf gebruikt. Dit hoofdstuk is enkel ter verduidelijking.

7.2 De stelling van Rice

Om de stelling van Rice te bewijzen, is het best dat we eerst even twee definities uit de cursus herhalen, namelijk deze van de niet-triviale eigenschap en deze van de taal-invariante eigenschap.

Theorem 6 (Niet-triviale eigenschap)**.** *Een eigenschap* P *van Turingmachines heet niet-triviaal indien* $Pos_P \neq \emptyset$ *en ook* $Neg_P \neq \emptyset$*. Er bestaan dus Turingmachines die deze eigenschap* P *bezitten, maar ook machines die deze niet bezitten.*

Theorem 7 (Taal-invariante eigenschap)**.** *De eigenschap* P *heet taal-invariant indien alle machines die dezelfde taal bepalen hebben ofwel allemaal* P*, ofwel heeft geen enkele ervan* P*.*

$$L_{M_1} = L_{M_2} \Rightarrow P(M_1) = P(M_2)$$

Met deze twee definities in het achterhoofd, kunnen we overgaan naar de formele definitie van de stelling van Rice, met het bewijs als gevolg.

Theorem 8 (Stelling van Rice)**.** *Voor elke niet-triviale, taal-invariante eigenschap* P *van Turingmachines geldt dat* Pos_P *(en ook* Neg_P*) niet beslisbaar is.*

Bewijs. Neem aan dat de Turingmachine $M_\emptyset$ de lege taal beslist. Laten we er nu van uit gaan dat deze machine een bepaalde eigenschap P heeft. De stelling zegt ons dat de eigenschap P niet-triviaal is. Uit de definitie kunnen we dan afleiden dat $Pos_P \neq \emptyset$ (en ook $Neg_P \neq \emptyset$). Aangezien deze verzameling niet leeg is, moet er een Turingmachine X bestaan met deze eigenschap P. Laat ons zeggen dat deze Turingmachine de taal L_X beslist.

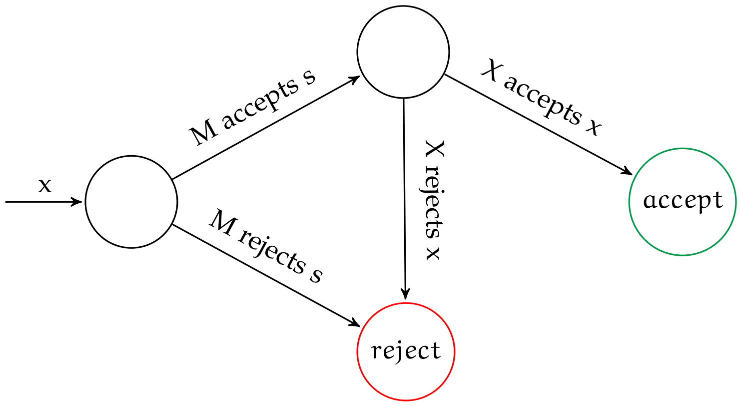
We gaan nu proberen een contradictie te bekomen door aan te nemen dat de stelling niet waar is. We nemen dus aan dat Pos_P (en dus ook Neg_P) beslisbaar is. We gaan nu een beslisser B proberen op te stellen voor Pos_P die deze beslist[1]. Om B te maken, gaan we eerst een hulpmachine $H_{M,s}$ opstellen.

Deze hulpmachine $H_{M,s}$ heeft een Turingmachine M en een string s in zich. Deze staan vast voor de machine en kunnen dus niet veranderen[2]. De input van deze machine is een string $x \in L_X$. Wanneer $H_{M,s}$ gestart wordt, zal deze eerst M laten lopen over s. Indien M s reject, zal $H_{M,x}$ **altijd** rejecten.

1 Later zullen we deze B gebruiken om een beslisser A te maken voor A_{TM}

2 Ze zijn als het ware gehardcoded.

Indien M s accept, dan zal $H_{M,s}$ overgaan naar fase 2. Hier zal de hulpmachine X over x laten lopen. Indien X x ook accept, dan zal de hulpmachine accepten. Indien X x reject, dan zal ook de hulpmachine rejecten.
Er zijn nu twee mogelijkheden voor $H_{M,s}$. Indien M s accept, dan gaat $H_{M,s}$ **altijd** overgaan tot het testen van x in X. In dit geval beslist $H_{M,s}$ de volledige taal L_X. De andere optie is dat M s reject. In dat geval gaat de hulpmachine **altijd** rejecten en dus enkel de lege taal accepteren.
Laat nu de beslisser B los op $H_{M,s}$. Dit wil zeggen dat de beslisser accept of reject voor de gegeven M en s.



Figuur 1: Visuele voorstelling van de hulpmachine $H_{M,s}$

Stel dat we nu een beslisser A maken voor A_{TM}. In dat geval moeten we dus elke M en s in A_{TM} testen. We kunnen dus zeggen dat A accept indien B $H_{M,s}$ accept, anders reject. We bekomen dus de volgende conclusie.

$$\begin{gathered} A \text{ accepts } < M, s > \\ \Updownarrow \\ B \text{ accepts } H_{M,s} \\ \Updownarrow \\ H_{M,s} \text{ heeft eigenschap } P \\ \Updownarrow \\ H_{M,s} \text{ accepts } L_X \\ \Updownarrow \\ M \text{ accepts } s \end{gathered}$$

Concluse: A is een beslisser voor A_{TM}, maar dit is onmogelijk aangezien A_{TM} niet beslisbaar is! Hieruit kunnen we concluderen dat B niet bestaat en dus is Pos_P niet beslisbaar. **Contradictie**. $\square$