Logische Complexiteit

Universele Turing machines

College 12

Donderdag 18 Maart

Hoog-niveau beschrijvingen en coderen

Vanaf nu: hoog-niveau beschrijvingen van TM's.

Daarbij worden objecten die geen eindige woorden zijn als zodanig gecodeerd.

Vb. De input van een TM bestaat uit paren van woorden in alfabet $\{a, b, c\}$. Een paar van twee woorden $w, v \in \{a, b, c\}^*$ kan bijvoorbeeld gecodeerd worden als

$$w\#v$$

Hierbij wordt het alfabet dus uitgebreid met $\#$.

Hoog-niveau beschrijvingen en coderen

Vb. De input van een TM bestaat uit de CFG's waarvan de variabelen $\{S, A\}$ en het alfabet $\{\epsilon, 0\}$ is.

De input zou gecodeerd kunnen worden als woorden in alfabet $\{S, A, 0, \epsilon, \rightarrow, \diamond\}$, waarbij $\diamond$ de regels van de CFG scheidt.

De regels van een CFG zijn bijv. $\mathcal{R} = \{S \rightarrow SAS, A \rightarrow A0A \mid \epsilon\}$. Daarbij hoort dan het woord

$$\diamond S \rightarrow SAS \diamond A \rightarrow A0A \diamond A \rightarrow \epsilon \diamond$$

De TM kan herkennen of een input de juiste vorm heeft, d.w.z. of het bestaat uit regels gescheiden door $\diamond$'s.

Hoog-niveau beschrijvingen en coderen

Vb. De input van een TM bestaat uit propositionele formules in variabelen $p_1, p_2, p_3, \ldots$.

Die kunnen als volgt gecodeerd worden in alfabet $\{0, 1, \wedge, \vee, \neg, \to, (,)\}$: variabele p_i wordt gecodeerd als de binaire representatie van i en de connectieven coderen zichzelf.

De formule $(p_1 \vee p_5) \wedge \neg p_4$ wordt dus gerepresenteerd door het woord:

$$(\, 1 \vee 101 \,) \wedge \neg 100$$

Hoog-niveau beschrijvingen en coderen

Om aan te geven dat de input gecodeerd wordt, volgens een van tevoren vastgestelde codering, worden haakjes gebruikt: de code van object O wordt aangeduid met $\langle O \rangle$.

Om aan te geven over welke input het gaat is de eerste zin van een hoog-niveau beschijving van een TM dan:
"Op input $\langle O \rangle$, waarbij O een ... is, doe ..."

Dit is weer een afkorting van:
"Op input w:
1. Ga na of w van de vorm $\langle O \rangle$ is voor e.o.a. O. Zo niet, reject.
2. ..."

Hoog-niveau beschrijvingen en coderen

Om aan te geven dat de input gecodeerd wordt, volgens een van tevoren vastgestelde codering, worden haakjes gebruikt: de code van object O wordt aangeduid met $\langle O \rangle$.

Om aan te geven over welke input het gaat is de eerste zin van een hoog-niveau beschijving van een TM dan:
"Op input $\langle O \rangle$, waarbij O een . . . is, doe . . . "

Dit is weer een afkorting van:
"Op input w:
1. Ga na of w van de vorm $\langle O \rangle$ is voor e.o.a. O. Zo niet, reject.
2. . . . "

Vb. Als de input van een TM uit formulas bestaat, dan wordt de eerste zin: "Op input $\langle \varphi \rangle$, waarbij φ een formule is, doe . . . ".

Dit is dus een afkorting van: "Op input w:
1. Ga na of w van de vorm $\langle \varphi \rangle$ is voor e.o.a. formule φ. Zo niet, reject.
2. . . . "

TM's met als input TM's

Er bestaat een eindig alfabet Σ_{TM} waarin *alle* TM's gecodeerd kunnen worden.

Def. De *universele Turing machine* U is de volgende TM:
U: "Op input $\langle M, w\rangle$, waarbij M een TM is en w een woord in het aflabet van M:
1. Simuleer M op w.
2. Als M een "accept" toestand bereikt, accept; als M een "reject" toestand bereikt, reject."

U heet universeel omdat het elke andere TM simuleert.

Voor elke TM M en elk woord w in het alfabet van M geldt:

- M accepteert w desda U het woord $\langle M, w\rangle$ accepteert.
- M verwerpt w desda U het woord $\langle M, w\rangle$ verwerpt.
- M stopt niet op w desda U niet op het woord $\langle M, w\rangle$ stopt.

Universele modellen

Def. Een model heet *universeel* als het TM's simuleert.

St. DFA's met 2 stacks zijn universeel.

St. Java, C, Lisp, PostScript, Excel, . . . zijn universeel.

St. Berekenen via biologische operaties op DNA is universeel.

St. Quantum computers zijn universeel.

St. Game of Life is universeel.

Beslisbare talen

Def. Een taal is *beslisbaar* als er een beslisser is (een TM die op elke input uiteindelijk stopt) die de taal herkent.

St. De volgende talen zijn beslisbaar:

$$A_{\text{DFA}} =_{def} \{\langle M, w\rangle \mid M \text{ is een DFA die woord } w \text{ accepteert}\}$$
$$A_{\text{NFA}} =_{def} \{\langle M, w\rangle \mid M \text{ is een NFA die woord } w \text{ accepteert}\}$$
$$A_{\text{REX}} =_{def} \{\langle R, w\rangle \mid R \text{ is een reg.ex. die woord } w \text{ genereert}\}$$
$$E_{\text{DFA}} =_{def} \{\langle M\rangle \mid M \text{ is een DFA met } \mathcal{L}(M) = \emptyset\}$$
$$A_{\text{CFG}} =_{def} \{\langle G, w\rangle \mid G \text{ is een CFG die woord } w \text{ genereert}\}$$
$$E_{\text{CFG}} =_{def} \{\langle G\rangle \mid G \text{ is een CFG met } \mathcal{L}(G) = \emptyset\}$$
$$EQ_{\text{DFA}} =_{def} \{\langle M, N\rangle \mid M, N \text{ zijn DFA's met } \mathcal{L}(M) = \mathcal{L}(N)\}$$

St. Elke context-vrije taal is beslisbaar.

Beslisbare en onbeslisbare talen

St. De taal

$$EQ_{\mathrm{DFA}} =_{def} \{\langle M, N\rangle \mid M, N \text{ zijn DFA's met } \mathcal{L}(M) = \mathcal{L}(N)\}$$

is beslisbaar.

St. De taal

$$EQ_{\mathrm{CFG}} =_{def} \{\langle G, H\rangle \mid G, H \text{ zijn CFG's met } \mathcal{L}(G) = \mathcal{L}(H)\}$$

is onbeslisbaar.

Onbeslisbare talen

St. De taal

$$A_{\mathsf{TM}} =_{def} \{\langle M, w\rangle \mid M \text{ is een TM die } w \text{ accepteert}\}$$

is Turing-herkenbaar (door de universele TM U), maar onbeslisbaar.

St. De taal

$$\overline{A_{\mathsf{TM}}} =_{def} \{\langle M, w\rangle \mid M \text{ is een TM die } w \text{ niet accepteert}\}$$

is niet Turing-herkenbaar.

Finis