List of decision problems

Jens Kristian Refsgaard Nielsen

6. april 2019

1 The theory

A decision problem is something that takes an input and asks a question, we wish to determine whether it is possible to make a general algorithm that can answer this question.

1.1 Reduces to, $P_1 \leq P_2$

We say a decision problem $P_1 = (I_1, Q_1)$ reduces to another problem $P_2 = (I_2, Q_2)$ iff $\exists F : I_1 \to I_2$ such that $\forall i \in I_1 : i$ is a yes instance of Q_1 iff $F(i)$ is a yes instance of Q_2.

1.2 Reduction theorem

Given two problems and a reduction $P_1 \leq P_2$ we get two very important results:
Decideable if P_2 is decideable then so is P_1,
"if you live inside something solveable, then you too are solveable"
Undecideable if P_1 is undecideable then so is P_2,
"if something unsolveable lives inside of you, there is no salvation in more data"

1.3 Language property

A property R on TMs is a language property if

$$\forall M, T \in \mathrm{TM} : \mathcal{L}(M) = \mathcal{L}(T) \Rightarrow (R(M) \iff R(T))$$

And non trivial if there exists two TM's such that R(M) and ¬R(T), i.e. there is a language with the property and one without it.

1.3.1 Rice theorem

Every non-trivial language property of Turing machines is undecideable.

1.3.2 Examples

2 The list

In order to decide whether something is decideable or not, it is very important to have a good list of results to use for our deductions. What follows are lists of problems handled in the lectures

and exercise classes of "Beregnelighed & Logik"at Aarhus university in the spring of 2019.
Each will be of the form: Name I: input Q: question
Notation:

$$\begin{aligned}
\text{TM} &= \text{set of turing machines} \\
\text{CFG} &= \text{set of context free grammars} \\
\text{REC} &= \text{set of recursive languages} \\
\text{RECE} &= \text{set of recursive enumerable languages} \\
\text{REG} &= \text{set of regular languages}
\end{aligned}$$

2.1 Decideable problems

I: $G \in \text{CFG}, w \in \Sigma_G^*$	Q: $w \in \mathcal{L}(G)$?	(2.1)
I: $G \in \text{CFG}$	Q: $\mathcal{L}(G) = \emptyset$?	(2.2)
I: $G_1, G_2 \in \text{CFG}$	Q: $\mathcal{L}(G_1) \cup \mathcal{L}(G_2) = \emptyset$?	(2.3)
I:	Q:	(2.4)

2.2 Undecideable problems

Halting	I: T$\in$TM,$w \in \Sigma_{\text{T}}$,	Q: does T halt on w?	(2.5)
Self-Accept	I: M $\in$ TM	Q: Does M halt on e(M)?	(2.6)
Accepts	I: M $\in$ TM, w $\in \Sigma_M^*$	Q: Does M accept w?	(2.7)
Halts	I: M $\in$ TM, w $\in \Sigma_M^*$	Q: Does M halt on w?	(2.8)
Accept-Λ	I: M $\in$ TM	Q:$\Lambda \in \mathcal{L}(M)$?	(2.9)
Accepts All	I: M $\in$ TM	Q:$\mathcal{L}(M) = \Sigma_M^*$?	(2.10)
Subset	I: M$_1$,M$_2$ $\in$ TM	Q:$in\mathcal{L}(M_1) \subset \mathcal{L}(M_2)$?	(2.11)
Equivalent Λ	I: M$_1$,M$_2$ $\in$ TM	Q:$\mathcal{L}(M_1) = \mathcal{L}(M_2)$	(2.12)
WriteSym	I: M $\in$ TM, $a \in \Sigma_M^*$	Q: will M write a?	(2.13)
PCP	I: A,B : $\lvert$A$\rvert$ = $\lvert$B$\rvert$	Q: $\exists J : \Pi_{j \in J} A_{i_j} = \Pi_{j \in i} A_{i_j}$?	(2.14)
MPCP	I: A,B : $\lvert$A$\rvert$ = $\lvert$B$\rvert$	Q: $\exists J : A_1 \Pi_{j \in J} A_{i_j} = B_1 \Pi_{j \in i} A_{i_j}$?	(2.15)
AmbiCFG	I: $G \in$ CFG	Q: is G ambiguous?	(2.16)

KitchenTiling	I: set of tiles	Q: can we tile ANY square with matching sides?	(2.17)

2.3 Reductions

$$\begin{aligned}
\text{Accepts} \ &\leq \ \text{MPCP} \ \leq \ \text{PCP} && (2.18) \\
\text{SelfAccept} \ &\leq \ \text{Accept} \ \leq \ \text{Halts} && (2.19) \\
\text{Accept} \ &\leq \ \text{Accept-}\Lambda \leq \text{writeSym} && (2.20) \\
\text{Accept-}\Lambda &\leq \ \text{AcceptAll} \ \leq \ \text{Subset} \ \leq \ \text{Equivvalent} && (2.21) \\
\text{PCP} \ &\leq \text{AmbiCFG} && (2.22)
\end{aligned}$$