

Assignment I3

Jens Kristian Refsgaard Nielsen

2. april 2019

1 Question 1a

Consider the dominos

$$\langle (b, bbb), (babbb, ba), (bab, aab), (ba, a) \rangle \quad (1.1)$$

We wish to define CFG's corresponding to the top and bottom parts of the dominos, we proceed by the procedure presented in lecture 17 slide 61, where G_A is the "top" of the dominos and G_B corresponds to the "bottom" part.

$$\begin{aligned} G_A : S &\rightarrow bS1 \mid babbbS2 \mid babS3 \mid baS4 \mid \\ &\quad b1 \mid babbb2 \mid bab3 \mid ba4 \\ G_B : S &\rightarrow bbbS1 \mid baS2 \mid aabS3 \mid aS4 \mid \\ &\quad bbb1 \mid ba2 \mid aab3 \mid a4 \end{aligned}$$

Notice though that in the context of the PCP the third domino is superfluous, since all rules in G_A start with a b and only ever has 1 a in a row, so we can reduce to

$$\begin{aligned} G'_A : S &\rightarrow bS1 \mid babbbS2 \mid baS3 \mid \\ &\quad b1 \mid babbb2 \mid ba3 \\ G'_B : S &\rightarrow bbbS1 \mid baS2 \mid aS3 \mid \\ &\quad bbb1 \mid ba2 \mid a3 \end{aligned}$$

However these grammars do not generate the same languages, so if we want the languages in general use G_A and G_B .

2 Question 1b

We wish to find a word in the intersection $\mathcal{L}(G_A) \cap \mathcal{L}(G_B)$. Here we can use the grammars G'_A and G'_B and we get the great idea to use bab^6a3112 , with derivations

$$\begin{aligned} S &\Rightarrow bab^3S2 \Rightarrow bab^3bS12 \Rightarrow bab^4bS112 \\ &\Rightarrow bab^6a3112 \\ S &\Rightarrow baS2 \Rightarrow bab^3S12 \Rightarrow bab^6S112 \\ &\Rightarrow bab^6a3112 \end{aligned}$$

So $bab^6a3112 \in \mathcal{L}(G_A) \cap \mathcal{L}(G_B)$

3 Question 2a

The Universal Halting Problem (UHP) takes as single input a word $w \in \{0,1\}^*$. The question is if the Universal Turing Machine U halts on w . We wish to produce a reduction that shows that UHP is undecidable.

Recall the Halting problem (HP) that takes as input a Turing machine M and a word w , with question does M halt on w ?

Claim: $HP \leq UHP$

Proof: We wish to use definition JM9.6 and produce a function $F : TM \times Words \rightarrow \{0,1\}^*$ such that $F(M, w)$ is a yes instance of UHP iff (M, w) is a yes instance of HP.

Define then $F(M, w) := e(M)w$, the encoding of the Turing machine followed by the encoding of the input word.

Assume now that (M, w) is a yes instance of HP, i.e. that M halts on w , now by definition of the Universal Turing machine we know that M halts on w if and only if U halts on $e(m)w$, so $e(m)w \in \{0,1\}^*$ so is a yes instance of UHP.

Assume on the other hand that $e(M)w$ is a yes instance of UHP, then U halts on it, and by definition of U once more, M halts on w so (M, w) is a yes instance of HP. \square

To drive us home recall that HP is undecidable, and then $HP \leq UHP$ implies that UHP too is undecidable, by Theorem lecture 15 slide 9.

4 Question 2b

The Halt-Sometimes Problem (HSP) takes a Turing Machine M as input. The question is if there is some word for which M would halt.

Claim: $UHP \leq HSP$

Proof: We wish to use definition JM9.6 and produce a function $F : \{0,1\}^* \rightarrow TM$ such that $F(w)$ is a yes instance of HSP iff w is a yes instance of UHP.

Define then $F(w) := U(w) \rightarrow \text{AcceptAll}$ to be the TM that takes any input η , and starts out by running w through the Universal Turing Machine U , if U halts then our TM runs η on AcceptAll, which of course accepts all inputs, so especially η .

Now to prove that F has the desired property let w be a yes instance of UHP, i.e. U accepts w , then the machine $F(w)$ accepts all η , so especially one η and $F(w)$ is a yes instance of HSP. Assume $F(w)$ is a yes instance of HSP then there is an input η for which it halts, and by construction this must mean that $U(w)$ halts, because it is the first task $F(w)$ performs, so w is a yes instance of UHP. \square

To drive us home we saw that UHP is undecidable, and then $UHP \leq HSP$ implies that HSP too is undecidable, by Theorem lecture 15 slide 9.

5 Question 2c

The Accepts-Something Problem (ASP) takes a Turing Machine M as input. The question is if there is some word that would be accepted by M .

We wish to prove that ASP is a language property:

Let M, T be turing machines assume $\mathcal{L}(M) = \mathcal{L}(T)$, we then have to show that ASP is true for M iff it is true for T . Assume that M is a yes instance of ASP, then there is a word w that M accepts. Recall that $\mathcal{L}(H) := \{w \mid H \text{ accepts } w\}$ so $w \in \mathcal{L}(M) = \mathcal{L}(T)$ and T accepts w , so T is a yes instance of ASP \square

So ASP is a language property, let us see that it is a non trivial one, we have to find two turing machines, one yes instance and one no instance. For the yes instance take the TM discussed earlier AcceptAll that accepts everything, so especially something, so a yes instance of ASP. To find a no instance take RejectAll or HangAll that rejects all inputs respectively hangs on all inputs, these TM's accept nothing so are no instances of ASP, and ASP is a non-trivial language property.

Recall Rice's theorem that all non trivial language properties are undecideable, so especially ASP. \square