Word Equations: Sheet 6

Task 1 Improve the recompression approach to equality testing for SLPs so that it can handle SLPs in which we can use X^m in the rules, i.e. powers of nonterminals, which we treat as size 1 for size considerations.

Task 2 Suppose that we are given an SLP such that $\mathrm{val}(\mathcal{A})$ contains no two consecutive identical letters. Show that we can calculate (in linear time) a partition, such that it covers a constant fraction of letters in $\mathrm{val}(\mathcal{A})$.
Hint: Model assumption: we can fit $|\mathrm{val}(\mathcal{A})|$ in $\mathcal{O}(1)$ memory cells.

Task 3 [**More restricitve model**] Show that the running time of blocks compression (with appropriate modifications, perhaps) is the same, if we only assume that n fits in $\mathcal{O}(1)$ memory cells.

Task 4 [**More restricitve model**] Show that if we only assume that n fits in $\mathcal{O}(1)$ memory cells then the equivalence of two SLPs can be decided in $\mathcal{O}(n \log N \log n)$.
Hint: [1]

Task 5 [**Yet another LZ77 to SLP approach, 2 points**] In the following exercises by *factorisation* we denote an LZ77-style factorisation, but without the assumption on the minimality. We assume that it is not self-referencing.

Given a string and its factorisation devise a pairing satisfying the following conditions

(P1) there are no two consecutive letters that are both unpaired;

(P2) if the first (last) letter of a factor f is paired then the other letter in the pair is within the same factor;

(P3) if $f = w[i \mathinner{..} i + |f| - 1]$ has a definition $w[\mathit{start}[i] \mathinner{..} \mathit{start}[i] + |f| - 1]$ then letters in f and in $w[\mathit{start}[i] \mathinner{..} \mathit{start}[i] + |f| - 1]$ are paired in the same way.

Show that using such a pairing we can shorten the text by a constant fraction, replacing the pairs by new letters. The new text inherits the factorisation from the original text.

Show that this approach yields an approximation algorithm that constructs an SLP out of LZ77, what is the approximation ratio?

Task 6 Show that we can preprocess in linear time a factorisation so that the self-references are at least 2 letters back; during this we introduce at 1 new 1-letter factor per factor of larger amount of letters. How does this help with the algorithm above.

[1] The only real difference is the partition for pairs that cover many letters. Instead, find $\mathcal{O}(\log n)$ such partitions. How does this affect the size?