We present a new algorithm for subsequence matching in grammar compressed strings. Given a grammar of size $n$ compressing a string of size $N$ and a pattern string of size $m$ over an alphabet of size $\sigma$, our algorithm uses $O(n+\frac{n\sigma}{w})$ space and $O(n+\frac{n\sigma}{w}+m\log N\log w\cdot \text{occ})$ or $O(n+\frac{n\sigma}{w}\log w+m\log N\cdot \text{occ})$ time. Here $w$ is the word size and occ is the number of minimal occurrences of the pattern. Our algorithm uses less space than previous algorithms and is also faster for $\text{occ}=o(\frac{n}{\log N})$ occurrences. The algorithm uses a new data structure that allows us to efficiently find the next occurrence of a given character after a given position in a compressed string. This data structure in turn is based on a new data structure for the tree color problem, where the node colors are packed in bit strings.