1. We prove that the two definitions given of lexicographic order are the same. Recall the definitions. The recursive definition:

BASIS.

1. $\epsilon < w$ for any string w other than ϵ itself.

2. If $c < d$, where c and d are characters, then for any strings w and x, we have $cw < dx$.

INDUCTION. If $w < x$ for strings w and x, then for any character c we have $cw < cx$.

The iterative definition. Let $C = c_1 c_2 \cdots c_k$ and $D = d_1 d_2 \cdots d_m$ be two strings. We say $C < D$ if either of the following holds:

1. That $k < m$ and for $i = 1, 2, \ldots, k$ we have $c_i = d_i$.

2. For some value of $i > 0$, the first $i - 1$ characters of the two strings agree, but the ith character of the first string is less than the ith character of the second string.

We prove first that the recursive definition is the same as the iterative definition by complete induction on the number of times the recursive rule is applied to the strings.

STATEMENT $S(n)$: If it is necessary to apply the recursive rule n times to show that $w < x$, then w precedes x according to the iterative definition of 'lexicographic order'.

We say that there is a necessary number of times to apply the recursive rule to the strings to show that $w < x$. There is a minimum number, which is the lowest number of applications needed until either basis case is satisfied. There is a maximum number, which corresponds to the length of w. The minimum number here is exactly what we mean by the necessary number in the statement.

We say this as opposed to "under the recursive definition, $w < x$ after n applications of the recursive rule ...". The number n cannot be arbitrary since there is a minimum and maximum. Thus we specify that we must meet this necessary number to show that $w < x$.

BASIS. The basis is $n = 0$, that is when either basis case holds trivially. Then $w < x$ by the recursive definition. Thus rule (1) of the iterative definition holds where $w = \epsilon$, and rule (2) holds where the basis (2) applies. Therefore the basis is true.

INDUCTION. Assume that $S(i)$ is true for $0 \leq i \leq n$. We shall prove $S(n+1)$. That is, we apply the recursive rule $n + 1$ times to show that $w < x$. Consider the $n + 1$th application of the recursive rule, in which we took two strings cw_1 and cx_1, where $w_1 < x_1$ is already known, and determined that $cw_1 = w < x = cx_1$. Since w_1 precedes x_1 without requiring more than n applications of the recursive rule, then the inductive hypothesis applies to both w_1 and x_1. Therefore w_1 precedes x_1 according to the iterative definition of lexicographic order.

We now must prove that $cw_1 = w < x = cx_1$ under the iterative definition. We have that cw_1 and cx_1 are only one character longer than w_1 and x_1. Hence in the iterative definition, we substitute k and m for $k + 1$ and $m + 1$, and rule (1) holds. For rule (2), we substitute i for $i + 1$, and thus the rule holds. Since the iterative definition holds for cw_1 and cx_1, then it holds for w and x. Therefore $S(n)$ is true for $n \geq 0$. ✦

We assume that the implication is true, that is the statement. Then we show that w, x satisfy the recursive definition by decomposing them into cw_1, cx_1, and this is simply an application of the recursive rule on w_1, x_1. We show that we can apply the inductive hypothesis to w_1, x_1. Then we take w_1, x_1 and from there we prove that the compositions cw_1, cx_1 satisfy the iterative definition. Since cw_1, cx_1 satisfy, then w, x satisfy, and thus the two definitions are the same.

We prove now that the iterative definition is the same as the recursive definition by complete induction on the number of initial positions that w and x have in common.

STATEMENT $S(n)$: If w and x have in common n initial characters and $w < x$, then w precedes x according to the recursive definition of lexicographic order.

If it requires n applications of the recursive rule to prove that $w < x$, then we shall say it takes n applications of the "iterative rule" to prove the same. The iterative rule is the operation that is applied n times. Clearly that operation is the number of comparisons on the n initial characters of w and x. Thus we have determined the iterative operation parallel to the recursive operation, and this lets us know what formal parameter to use for the statement.

BASIS. The basis is 0. There are zero initial characters in common, hence $w < x$. Either $w = \epsilon$ which satisfies the basis (1), or that the first characters c, d of w, x respectively are such that $c < d$ which satisfies the basis (2). The basis is proven.

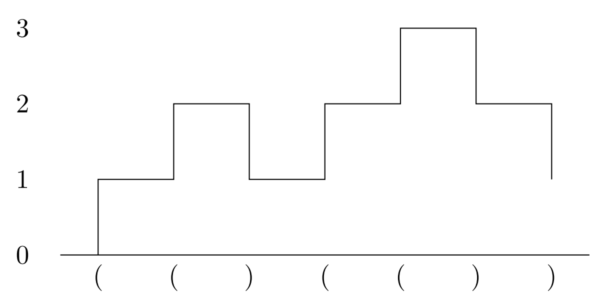
INDUCTION. Assume $S(i)$ for $0 \leq i \leq n$. We shall prove $S(n+1)$. Consider the strings w and x where they have in common $n+1$ initial characters and $w < x$. Write $cw_1 = w < x = cx_1$. Clearly $w_1 < x_1$. Hence the inductive hypothesis applies to w_1 and x_1. We do not consider any characters after the strings w_1 and x_1 because $w_1 < x_1$ regardless.

Since w_1 and x_1 satisfy the recursive definition, then we need only apply the recursive rule once to cw_1 and cx_1. Therefore $cw_1 = w < x = cx_1$ under the recursive definition. This proves the inductive step, hence $S(n)$ for all $n \geq 0$. ✦

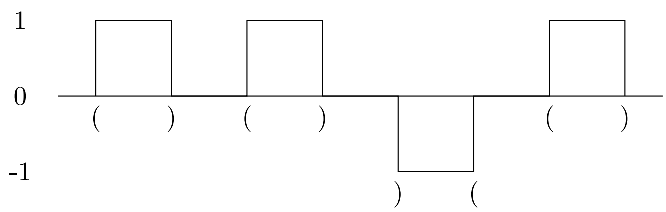
We conclude that the recursive and iterative definitions for lexicographic order are the same.

2.

a) `(()(())` is not profile-balanced, it does not satisfy rule (1).

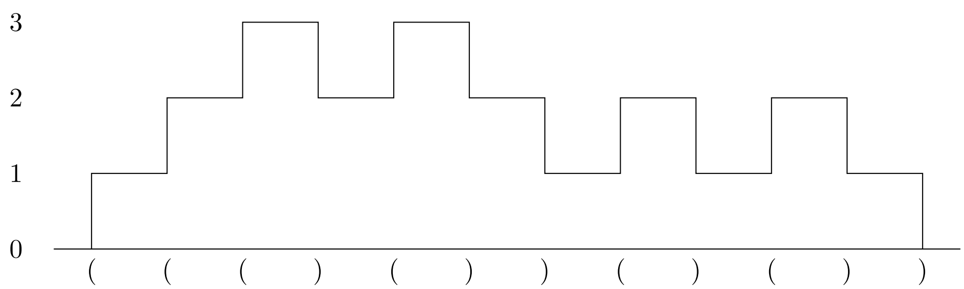


b) `()())(()` is not profile-balanced, it does not satisfy rule (2).



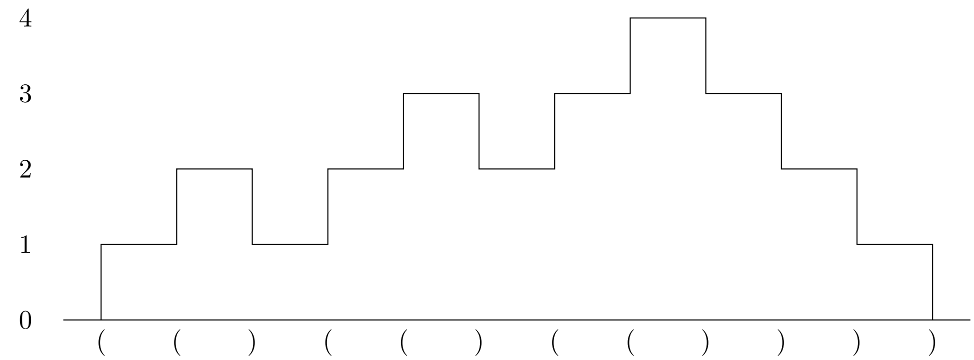
c) `((()())()())` is profile-balanced. By the recursive definition, we have

$x = y = \epsilon \rightarrow \texttt{()}$
$x = \epsilon,\ y = \texttt{()} \rightarrow \texttt{()()}$
$x = y = \texttt{()()} \rightarrow \texttt{(()())()()}$
$x = \texttt{(()())()()},\ y = \epsilon \rightarrow \texttt{((()())()())}$



d) `(()(()(())))` is profile-balanced. By the recursive definition, we have

$x = y = \epsilon \rightarrow \texttt{()}$
$x = \texttt{()},\ y = \epsilon \rightarrow \texttt{(())}$
$x = \epsilon,\ y = \texttt{(())} \rightarrow \texttt{()(())}$
$x = \texttt{()(())},\ y = \epsilon \rightarrow \texttt{(()(()))}$
$x = \epsilon,\ y = \texttt{(()(()))} \rightarrow \texttt{()(()(()))}$
$x = \texttt{()(()(()))},\ y = \epsilon \rightarrow \texttt{(()(()(())))}$



3. We shall prove by induction the following statement on the number of times the recursive rule of the definition of balanced parentheses is used.

STATEMENT $S(n)$: After n applications of the recursive rule of the definition of balanced parenthesis to construct a string S, then S is the string of parentheses of some arithmetic expression.

BASIS. The basis is $n = 0$, and ϵ is the string of balanced parentheses for any of the atomic operands that are arithmetic expressions. This proves the basis.

INDUCTION. Assume $S(i)$ for $0 \leq i \leq n$ and $n \geq 0$. We shall prove $S(n+1)$. Consider the string of balanced parentheses S constructed from the $n+1$th application of the recursive rule on x and y. That is, $S = (x)y$, where x and y are strings of balanced parentheses, and we can apply the inductive hypothesis to both x and y. We can take arithmetic expressions E_1 and E_2 with some binary operator θ in either of the recursive rules to form $(E_1 \theta E_2)$. We can also form $(-E)$ where E is also an arithmetic expression. Clearly S constitutes the string of parentheses of both expressions. This proves the inductive step. Therefore $S(n)$ for $n \geq 0$. ✦

4.

a) $<$ is an infix binary operator.

b) & is a prefix unary operator.

c) % is an infix binary operator.

5.

BASIS. A file is either

1) A regular file

2) A directory

INDUCTION. If D is a non-empty directory, then it has at least one file within it.