1-D Algorithm Details

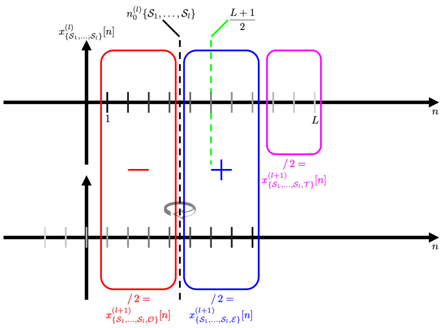
Fig. 1 recaps in graphical form the one-step decomposition operation of a generic node $x^{(l)}_{\{\mathcal{S}_1,\ldots,\mathcal{S}_l\}}[n]$ in level l, breaking down each possible occurrence depending on the $n_0^{(l)}\{\mathcal{S}_1,\ldots,\mathcal{S}_l\}$ value (in the following it is referred to as simply n_0).

In Fig. 1a, n_0 is in a half-integer position and to the left of the midpoint. Therefore, one can divide the original sequence into a "right" part and a "left" one of equal length. The causal even sequence then becomes the halved sum of the right part with the flipped version of the left part, while the anticausal odd sequence is the halved difference of the left part and the flipped version of the right part. The tail just copies the rightmost samples not involved in the previous computation.

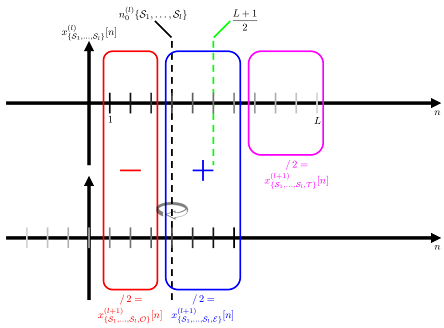
In Fig. 1b, n_0 is again to the left of the midpoint, but this time in an integer position. Therefore, the even sequence is one sample longer than the odd one. In particular, the first sample of the even sequence is exactly the sample in n_0. The other values of the decomposition sequences are obtained as in the previous case.

Fig. 1c is the analogous of Fig. 1a and Fig. 1d is the analogous of Fig. 1b, when instead n_0 is to the right of the midpoint. The procedure to obtain the even and odd children is the same, but the tail causal part (namely, its rightmost samples) is now the flipped version of the leftmost samples of the original sequence.

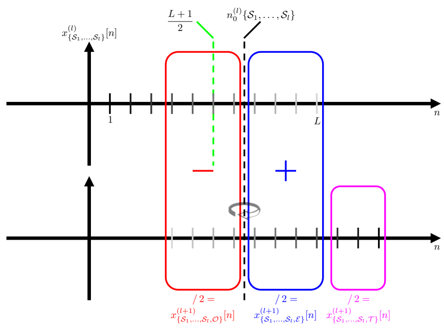
In Fig. 1e and Fig. 1f, n_0 is on the sequence midpoint. Therefore, there is no tail child node. If L is odd, the midpoint is in an integer position, so as for the previous cases in Fig. 1b and Fig. 1d the even sequence is a sample longer, whereas, if L is even, the midpoint is in a half-integer position and the even and odd children nodes have the same length, as in Fig. 1a and Fig. 1c.



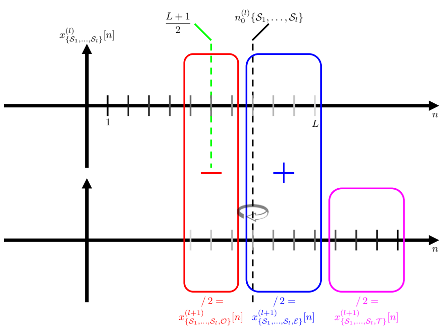
(a) $n_0 < \frac{L+1}{2}$, half-integer: the even and odd children have the same length, the tail is the original node right part not involved in the computation.



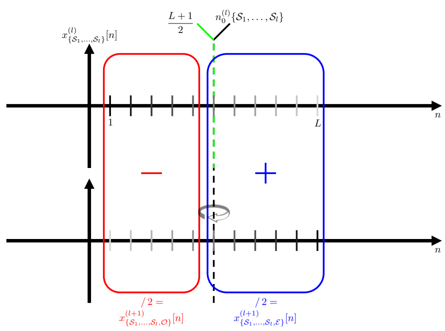
(b) $n_0 < \frac{L+1}{2}$, integer: the even child has a sample more than the odd one, the tail is the original node right part not involved in the computation.



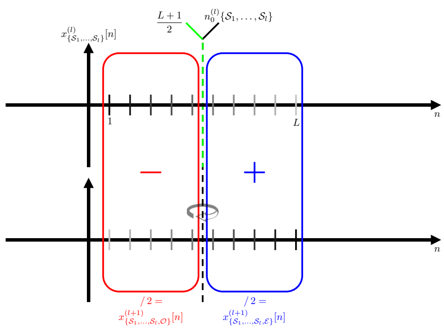
(c) $n_0 > \frac{L+1}{2}$, half-integer: the even and odd children have the same length, the tail is the flipped node right part not involved in the computation.



(d) $n_0 > \frac{L+1}{2}$, integer: the even child has a sample more than the odd one, the tail is the flipped node right part not involved in the computation.



(e) $n_0 = \frac{L+1}{2}$, integer (thus, L is odd): the even child has a sample more than the odd one, there is no tail.



(f) $n_0 = \frac{L+1}{2}$, half-integer (thus, L is even): the even and odd children have the same length, there is no tail.

Figure 1: Graphical depiction, detailing each possible occurrence on the position of the optimal symmetry point with respect to the sequence midpoint. For cases (a)–(d), $L = 11$ and n_0 is not the midpoint. Whether L is odd or even (thus placing the midpoint on an integer or half-integer position respectively) has no effect here. Instead, when n_0 is the midpoint, cases (e)–(f) detail the decomposition effect when L is odd ($L = 11$) or even $L = 10$.