For me simply adding min-width: 0; to the overflowing div prevented the overflow:
article { min-width: 0; }

Explanation:

min-width in a flex context: While the default min-width value is 0 (zero), for flex items it is auto. This can make block elements take up much more space than desired, resulting in overflow. min-width is defined to win against competing width and max-width, meaning as soon as the element's width would shrink below its implicit auto width, min-width: auto will kick in and keep the element from shrinking.

The fix is obvious now: min-width: 0

It tells the browser that this element has no right to take up any minimum width, and now it will be rendered properly, taking up only as much width as is available.

A note about flex-shrink: While flex-shrink sounds like it could help here, it does not. The described issue is about element-sizing based on the element's content, while flex-shrink defines shrinkage relative to other flex elements in the same context.