new sub-parse-tree instance, set its step to Op, set its key to sub_key, call its parser by passing this element, give back its pointer to tokenizer to set next ParserTree object to use as sub-parse-tree

sub-parse-tree will give back its already built token to from-parse-tree other than tokenizer, and also give back element it gotten, this will call from-parse-tree's parser, and latter will delete this sub instance

if build is not null, use append, else new

use struct-array: true

element type

element *as*	type	*note*
=	operator	
<	operator	
>	operator	
{	open brace	
}	close brace	
unquoted token	string	
quoted token	string	characters within quote may encode to utf-8 or Unicode when displaying
end of file	EOF	is always included in "other"