Formally Defining the Behaviour of Anchored Expressions:

$Lp(r) = \{w : w$ is accepted by r in programmer regex$\}$
$Lm(r) = \{w : w$ is accepted by r in Mathematical RE$\}$

$f(r) \rightarrow$ the mathematical version of programmer expression r
that is, $Lp(r) = Lm(f(r))$

Define f, and prove $Lp(r) = Lm(f(r))$ for any valid programmer's regular expression r.

$f = a \circ p$ (first parse.js conversion, then handle anchor "expansion")

p(word boundary) = Error
p(backreference) = Error
p([negative] look ahead/behind) = Error

p(...)	inside [...]	otherwise
\d	0-9	[0-9]
\D	Error	[^0-9]
\w	0-9A-Za-z_	[0-9A-Za-z_]
\W	Error	[^0-9A-Za-z_]
\s	<space>	<space>
\S	Error	[^]

$p(r+) = rr*$
$p(r\{x\}) = r^x$
$p(r\{x,\}) = r^x r*$
$p(r\{,x\}) = \mathop{\bullet}\limits_{i=1}^{\lfloor \log 2(x) \rfloor} (\varepsilon + p(r\{i\})) \cdot p(r\{, x - \lfloor \log_2 x \rfloor)$
$p(r\{x,y\}) = p(r\{x\})\, p(r\{, y-x\})$

Anything unspecified remains unchanged $p(x) = x$