and the user then releases the printer (this corresponds to the substring `rgd`). Then the first user's pending request is granted, the second user requests access to the printer again, and the first user (who presently has access to the printer) relinquishes the printer (this corresponds to the substring `GrD`). At this point there is one pending request by the second user. Before this request is honoured, however, the first user requests, is granted and relinquishes access to the printer twice (this corresponds to the substring `RGDRGD`). After that, the pending request of the second user is granted and, finally, the printer is released (this corresponds to the final substring `gd`).

The second string, y, corresponds to an incorrect behaviour of the system: the second user relinquishes the printer without having previously gained access to it (there is a `d` symbol but no preceding `g` symbol). The third string, z, corresponds to an incorrect behaviour in which the mutual exclusion property is violated: after the prefix `rgRG` both users have access to the printer at the same time.

We can describe the correct behaviour of the system, as a language L_{sys} over the alphabet Σ consisting of those strings in which the sequence of events does not violate the desired behaviour. Specifically, a string x in Σ^* belongs to L_{sys} if and only if all of the following hold.

(a) if x contains a symbol in $\{\mathtt{R}, \mathtt{G}, \mathtt{D}\}$ (respectively, $\{\mathtt{r}, \mathtt{g}, \mathtt{d}\}$), then the first such symbol is `R` (respectively, `r`);

(b) if $\mathtt{R}y$ (respectively, $\mathtt{r}y$) is a suffix of x, then y contains some symbol in $\{\mathtt{R}, \mathtt{G}, \mathtt{D}\}$ (respectively, $\{\mathtt{r}, \mathtt{g}, \mathtt{d}\}$) and the first such symbol is `G` (respectively, `g`);

(c) if $\mathtt{G}y$ (respectively, $\mathtt{g}y$) is a suffix of x, then y contains some symbol in $\{\mathtt{R}, \mathtt{G}, \mathtt{D}\}$ (respectively, $\{\mathtt{r}, \mathtt{g}, \mathtt{d}\}$) and the first such symbol is `D` (respectively, `d`);

(d) if $\mathtt{D}y$ (respectively, $\mathtt{d}y$) is a suffix of x, and y contains a symbol in $\{\mathtt{R}, \mathtt{G}, \mathtt{D}\}$ (respectively, $\{\mathtt{r}, \mathtt{g}, \mathtt{d}\}$), then the first such symbol is `R` (respectively, `r`);

(e) if $\mathtt{G}y\mathtt{g}$ (respectively, $\mathtt{g}y\mathtt{G}$) is a substring of x, then y must contain `D` (respectively `d`).

The first four properties ensure that each user goes through (zero or more) cycles of requesting the printer, being granted access to it, and then relinquishing it. In particular, a user can't just get to use the printer and then relinquish it without being granted access first; and the printer can't grant access to a user who has not requested it. Furthermore, if a user does request access, then the system is obligated to eventually grant it; and if a user is granted access, he/she is obligated to relinquish it. The last property ensures mutual exclusive access to the printer: If at some point access to the printer is granted to one user, access to the printer cannot be subsequently granted to the other user until the first user is done.

Note that the above description of the system behaviour permits a situation where a user requests access to the printer, but the other user requests, is granted access and relinquishes the printer a thousand times before the first user's original request is honoured. If such "unfair" behaviour is undesirable, we may wish to rule out such strings from our language,