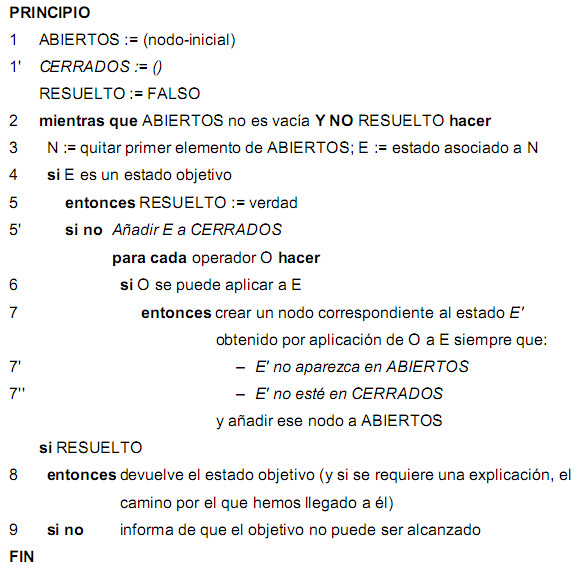
Se puede considerar una nueva versión del esquema genérico de búsqueda en el que los estados ya examinados son almacenados en una lista llamada (de estados) CERRADOS. Cuando se expande un estado sólo dan lugar a nuevos nodos aquellos estados que no aparezcan en ABIERTOS o que no estén en CERRADOS. Nótese que ABIERTOS es una lista de nodos (mientras que CERRADOS es una lista de estados), por lo que lo que habrá que comprobar es que un estado no es igual a ninguno de los estados asociados a los nodos ABIERTOS.

El esquema genérico que permite realizar búsquedas seguras y no redundantes en espacios de estados que sean grafos (dirigidos o no, acíclicos o no):

Vemos que las únicas diferencias con el esquema que hemos denominado búsqueda en árbol son:

* En la línea 1' se inicializa CERRADOS con la lista vacía.
* En la línea 5' se añade el estado que ya ha sido examinado a CERRADOS.
* En las líneas 7' y 7'' se comprueba si el estado generado es nuevo



Se podrían considerar variantes en las que las líneas 7' o 7'' no son incluidas.

Particularizando a la búsqueda en anchura las consecuencias serían las que siguen:

Si no incluimos la línea 7'' tendremos una versión que no cometerá cierto tipo de redundancias (en concreto, nunca re-examinará dos estados que aparerecieran en la versión anterior simultáneamente en ABIERTOS), pero no tiene en cuenta otras redundancias (cuando vuelve a aparecer un estado que ya ha sido examinado; nótese que esto puede darse tanto en grafos dirigidos acíclicos como en otros que tengan ciclos dirigidos) y, sobre todo, puede entrar en bucles infinitos si los estados objetivos no son alcanzables y hay ciclos dirigidos. Si eliminamos la línea 7' (manteniendo la 7'') este último problema no puede darse, pero un estado puede ser examinado varias veces (puesto que puede aparecer en dos nodos distintos de ABIERTOS).

En conclusión, la versión con las dos líneas tiene las siguientes propiedades en el caso de la búsqueda en anchura:

* Cada estado del espacio de estados es examinado, a lo más, una vez.
* Si existe solución, encuentra una de longitud mínima, independientemente del tipo de espacio de estados de que se trate (árbol, grafo dirigido acíclico, grafo dirigido o no-dirigido y cada uno de ellos finito o infinito).
* Si no existe solución, siempre terminará la ejecución salvo en el caso (inevitable para las estrategias exhaustivas) en el que el espacio de estados es infinito.

Como se ve las propiedades teóricas de la búsqueda en anchura en grafo son bastante buenas. Si embargo, no hay que olvidar que dicha mejora tiene un precio en cuanto al coste en tiempo (gestión de CERRADOS y algoritmos de búsqueda en ABIERTOS y CERRADOS) y, sobre todo, en espacio (la lista CERRADOS siempre aumenta, nunca se eliminan elementos de ella), por lo que si sabemos que el espacio de estados es un árbol será preferible utilizar la versión inicial. Por ejemplo, sería un derroche utilizar esta versión que acabamos de dar con la representación elegida para el problema de las n reinas. Este algoritmo de búsqueda en grafo permite clasificar a los estados en cuatro clases. En un instante cualquiera del proceso de búsqueda, distinguimos 4 tipos de estados:

* Los estados todavía no generados (que, por tanto, no aparecen asociados a ningún nodo de ABIERTOS ni están en la lista CERRADOS). Estos estados podemos considerar que tienen una existencia "potencial", puesto que hasta que no es generado un estado no aparece en ABIERTOS ni forma parte de CERRADOS. En general, muchos estados del espacio de estados no llegarán a ser nunca generados en el proceso de búsqueda.
* Los estados ya generados, pero todavía no examinados. Son aquéllos que aparecen en nodos de ABIERTOS.
* Los estados examinados, pero no expandidos. En realidad se trata de una clase con, a lo más, un elemento que es el estado que aparece ligado al nodo valor de la variable N.
* Los estados expandidos, que son aquellos que constituyen la lista CERRADOS.

Es importante hacer notar que la noción del árbol de búsqueda no varía (sigue tratándose de un árbol), aunque sí cambia su relación con el espacio de estados. En este nuevo algoritmo tenemos que una adyacencia del espacio de estados que sea examinada puede dar lugar a lo más a una adyacencia en el árbol de búsqueda, pero también puede no dar lugar a ninguna (en el caso en que el espacio de estados no sea un árbol). En el algoritmo de búsqueda en árbol, cada adyacencia examinada del espacio de estados daba lugar al menos a una adyacencia en el árbol de búsqueda, pero podía dar lugar a varias (en el caso en que el espacio de estados no sea un árbol).

Una última observación respecto a este algoritmo se refiere a que, contrariamente a lo que sucedía en la anterior versión, la igualdad entre estados pasa a ser muy importante, puesto que es necesario realizar tareas.