**Práctica Arquitectura De Computadores**

**Javier de Jesús Silva Marín**

**Michael Moreno Valoyes**

**Santiago Salazar Ramírez**

**Juan José Navarro Saldarriaga**

**Universidad Nacional de Colombia: Sede Medellín**

**3007863: Arquitectura de computadores**

**Hugo de Jesús Mesa Yepes**

**27 de agosto de 2021**

**1.Clasificador de posiciones relativas entre circunferencias**

**Teoría:**

En cuanto se empieza a clasificar las relaciones entre círculos se deben tomar sus distancias entre sus centros, como también se debe tomar la diferencia o suma existente en sus radios. Ya reconociendo el uso de estos datos se deben clasificar, su clasificación para el fácil entendimiento de ello.

Datos:

* R1:Radio del primer círculo.
* R2:Radio del segundo círculo.
* X1:Coordenada X del centro primer círculo.
* X2: Coordenada X del centro segundo círculo.
* Y1: Coordenada Y del centro primer círculo.
* Y2: Coordenada Y del centro segundo círculo.
* D = (Distancia entre los centros)

Con estas consideraciones expuestas podemos clasificar correctamente las relaciones de posición entre los círculos, los cuales serían:

* Exteriores: D > (R1+R2)
* Tangentes exteriormente: D = (R1+R2)
* Secantes: (|R1-R2|)<D < (R1+R2)
* Tangentes interiormente: D = |R2-R1|
* Interiores excéntricas: D < |R2-R1|
* Interiores concéntricas: D = 0; R1 ≠ R2
* Coincidentes: D = 0; R1 = R2

**Procedimiento:**

Inicialmente se le solicita al usuario 6 datos, siendo justamente estos datos los que se expusieron anteriormente. Con los datos obtenidos llamamos a la función “MainClassifier”, del módulo “clasify\_main” que contiene a esta función y a otra más llamada “DistanceBetweenCenters”, “MainClassifier” recibe en ST(5) al valor absoluto de la diferencia de los radios, en ST(4) se recibe la suma de los radios y de ST(3) a ST(0) se reciben las coordenadas de los centros de la circunferencia, en el orden presentado en la sección datos de este mismo documento. La función “MainClassifier” llama a la función “DistanceBetweenCenters”, la cual calcula la distancia entre los centros de los círculos, utilizando los valores que fueron anteriormente pasados, esta función regresa en ST(0) la distancia entre los centros y después de terminar la ejecución de la misma se llama a la función “BranchBasedOnDistance”, la cual según el valor de ST(0) llevará al programa ya sea a los módulos “different\_center” o “equal\_center”, mediante las funciones “DifferentCenterCase” e “EqualCenterCase” respectivamente. En estos dos módulos se darán todos los procedimientos que conciernen a la clasificación de las circunferencias. Finalmente, la función “DifferentCenterCase” llama a la función “CalculateIntersections”, que calcula y muestra el o los puntos de intersección de las circunferencias ingresadas de ser esto necesario (casos secantes, tangentes interiores y tangentes exteriores).

**2.Aproximación de decimales:**

Se escribió un módulo extra para la conversión de números en punto flotante a un formato más cómodo para el usuario, el cual le permite visualizar el número de una forma más cómoda. Esta función hace un parsing a un número en formato IEEE, del cual obtiene su parte entera, su signo y su parte decimal, que son impresas en la pantalla de forma separada. Para obtener la parte entera se hace un shift hacia la derecha en mantisa - exponente unidades, para hallar el exponente primero se elimina el bit del signo haciendo un shift en una unidad hacia la izquierda y luego hacia la derecha y luego haciendo un shift de 23 unidades (tamaño de la mantisa en este caso). Para hallar la parte decimal al número se le resta su parte entera y esta diferencia es multiplicada por una potencia de 10. Siendo el exponente de esta potencia lo que determina cuantas cifras decimales se van a utilizar.

Hacer esto directamente genera un error: para números que tengan ceros después del punto decimal, estos serán ignorados. Por ejemplo, el número 2.005 quedaría como 2.5. Para solucionar esto, se multiplica la parte decimal por 10 n veces (un n determinado por el programador/usuario que tenga acceso al código fuente), y se compara la parte entera de esta multiplicación con 0, si son iguales, se agrega un cero a un arreglo llamado “ZEROES\_ARRAY” este arreglo se imprime después del punto decimal (todo este proceso sucede de las líneas 122 a 175, en la label “zeroes\_loop”), de esta forma, los errores de en el formatting quedan reducidos solo el cómo identificar números negativos y qué hacer en caso de tener que imprimir uno.

Para los números negativos se chequea que el bit más significativo sea igual a uno, esto, en el formato IEEE, significa que el número es negativo. Si el bit resulta ser uno, una variable llamada “IS\_NEG” será igualada a 1. El resto del proceso se hace igual que para los positivos, exceptuando la parte donde se halla su parte decimal, que en lugar de restar al número su parte entera, se le suma. También, al final, se utiliza la variable IS\_NEG nuevamente para determinar si se tiene que imprimir un signo menos (-) antes de imprimir el número.

**3.Buscador de puntos de intersección**

**Teoría:**

La primera aclaración que se debe hacer respecto a los puntos de intersección es su procedimiento en cuanto al tipo de relación existente, puesto el procedimiento cambia al momento de tener una relación diferente.

Entre estas relaciones se encuentra:

* Tangentes exteriormente:

Diagrama

Descripción generada automáticamente

Para todos los casos calculamos los puntos de intersección de manera vectorial. En este caso notamos que podemos trazar un vector distancia entre los centros, aprovechándonos de que la línea que pasa por ambos centros también pasa por el punto de intersección. Podemos notar que el vector de posición que se ubica en el centro de la circunferencia más el vector libre R1 que es un vector de magnitud R1 y en dirección del vector distancia entre los centros da como resultado un vector de posición que se ubica en el punto de tangencia.

* Tangentes interiormente:

Diagrama

Descripción generada automáticamente

No es excepción a la regla de hallar el punto de intersección de forma vectorial. En este caso vemos que el R1 pasa por el R2, haciendo que ahí esté el punto de intersección. Vemos que el vector que se ubica en (X1,Y1) más el vector R1 y en dirección del vector distancia entre los centros daría el vector posición, justamente el que se ubica en el punto de intersección.

* Secantes:Forma

  Descripción generada automáticamente con confianza media

Con estas 3 ecuaciones vamos a resolver para A y K. Con estas magnitudes podremos calcular los vectores asociados y luego con estos los puntos de corte.

Restamos la tercera ecuación con la segunda.

Y luego reemplazamos a B en términos de A.

Sabemos que el vector asociado a A es un vector en la misma dirección al vector distancia entre los centros.

Ahora, para calcular el vector K se realiza un reemplazo en la tercera ecuación.

Sabemos que K es un vector normal a A, por lo que K sería igual a:

De último, como se puede observar gráficamente y llamando los puntos de intersección S1 y S2