Capítulo 2

Modelo básico de computo

En este capítulo, introduciremos conceptos básicos sobre representaciones de objetos geometricos, tales como polígonos, circulos, etc., y daremos algunos algoritmos elementales de la Geometría Computacional.

2.1 Representaciones de objetos geometricos elementales

Los objetos mas elementales que se manipulan en geometría computacional son colecciones de puntos, segmentos de rectas y polígonos. Un punto en el plano está definido por sus coordenadas x y y. Por tanto, en la computadora, es natural definir un punto en el plano como una arreglo p de dimensión 1×2, tal que $p(1)$ representa la coordenada x de p y $p(2)$ la coordenada y. Una colección de puntos $\{p_1, \ldots, p_n\}$ puede a su vez ser representada por un arreglo P de dimension $n \times 2$ de tal manera que $P(i, 1)$ y $P(i, 2)$ representan las coordenadas x y y de p_i respectivamente. Segmentos de lineas serán representados por parejas de puntos, o arreglos de dimensión 2×2.

Sea $\{p_1, \ldots, p_n\}$ una coleción de puntos en el plano tal que los segmentos de de linea $p_1p_2, \ldots, p_{n-1}p_n, p_np_1$ no se intersectan, salvo en sus extremos. El polígono P con vértices $\{p_1, \ldots, p_n\}$ es la región finita del plano acotada por la curva definida por la union de $p_1p_2, \ldots, p_{n-1}p_n, p_np_1$. Los segmentos $p_1p_2, \ldots, p_{n-1}p_n, p_np_1$ se llaman las aristas de P, y serán normalmente denotadas como $e_1, \ldots, e_n$. La forma mas simple de representar un polígono con n vertices, es nuevamente usando un arreglo de dimensión $n \times 2$, ver Figura 2.1.

0	0
2	-6
4	-2
11	-2
11	1
3	2

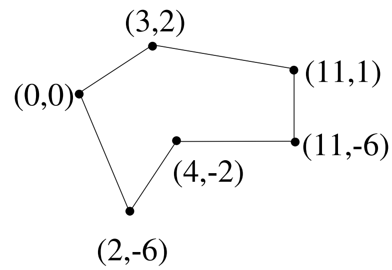

Figura 2.1: Representación de polygonos

2.2 Problemas de iclusión de puntos en polígonos

El primer problema que estudiaremos puede ser formulado de la siguiente manera:

Problema 2.1 *Sea P un polígono simple con n vertices, y p un punto en el plano. Desarroyar un algoritmo que determine si p pertenece a P.*

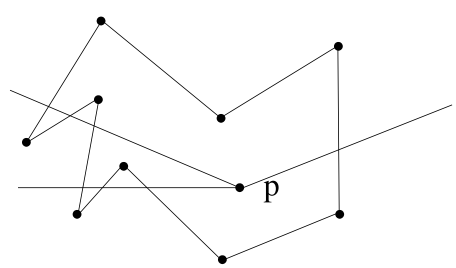

Figura 2.2: Verificando si un punto esta en el interior de un polígono

Para resover nuestro problema haremos uso de la siguiente observación:

Observación 2.1 *Sea p un punto en el interior de P, y r cualquier rayo que comienza en p. Si p pertenece al interior de P, r cruza la frontera de P un número impar de veces, ver Figura 2.2.*

Cabe hacer notar, que en la observación anterior, el termino cruzar, estrictamente hablando, significa pasar del interior al exterior de P, por tanto si p pertenece a la frontera o al exterior de P, cualquier rayo que parta de p cruzará la frontera un número par de veces. Esto nos lleve a obtener el siguiente algoritmo para determinar si un punto p pertenece a un polígono P:

Algorithmo 2.2.1 *(INCLUSIÓN)*

Datos de entrada *: Un Polígono P, y un punto p en el plano.*

Resultado *: PERTENECE ó NOPERTENECE.*

1. *Verificar si $p = (x_0, y_0)$ pertenece a alguna de las aristas de P, en caso afirmativo Resultado=PERTENECE, de otra manera pasar al siguiente paso.*

2. *Calcular el número cr de veces que el rayo r que contiene los puntos de la forma $(x_0, y); y \geq y_0$ cruza la frontera de P.*

3. *: Si cr es impar, Resultado=PERTENECE, de otra forma Resultado=NOPERTENECE*

2.2.1 La complejidad de INCLUSIÓN

Ahora procederemos a analizar la complejidad del algoritmo INCLUSION. El primer paso de INCLUSIÓN toma tiempo lineal, ya que verificar si un punto pertenece a un segmento de line puede hacerse en un número constante de operaciones elementales, ver ejercicio 2.

La segunda parte de nuestro algoritmo, también puede implementarse en tiempo lineal.El cálculo de cr puede hacerse de la siguiente forma:

1. Determinar todas las intersecciones de la frontera de P con la linea $y = y_0$.

2. Para cada intersección de r con la frontera de P verifique si esta intersección esta contenida a la izquierda de el punto (x_0, y_0), en tal caso descartese esta intersección. Si la intersección está a la derecha de (x_0, y_0) verifiquese si en esta intersección r cruza la frontera de P, en tal caso incrementemos el valor de cr en uno, de otra forma el valor de cr permanece constante, ver figura 2.4.

Es facil verificar que esto también puede hacerse en tiempo lineal. Por tanto tenemos:

Teorema 2.1 *Decidir si un punto pertenece a un polígono toma tiempo lineal.*

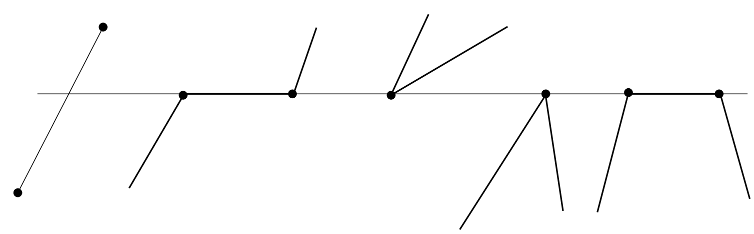

Figura 2.3: Los primeros dos casos muestran cuando la frontera de P cruza la recta $y = y_0$, en los ultimos tres casos, la recta $y = y_0$ intersecta sin cruzar la frontera de P.

2.2.2 Inclusión en polígonos convexos

En algunos casos, cuando se tiene mas información sobre la estructura de nuestro polígono, y haciendo un poco de trabajo preliminar, podemos acelerar el tiempo de respuesta al problema de inclusión de un punto en un polígono. Tal es el caso para polígonos convexos. Un polígono P es convexo, si para cualquier pareja de vértices de P, el segmento de recta que los une está totalmente contenido en P.

A contunuación probaremos el siguiente resultado:

Teorema 2.2 *Usando preprocesado y tiempo lineal, el problema de inclusión de un punto en un plígono convexo puede contestarse en tiempo logritmico.*

Sea P un polígono convexo con n vértices. Supongamos que los vértices de P estan etiquetados $\{p_1 = (x_1, y_1), \ldots, p_k = (x_k, y_k), \ldots, p_n = (x_n, y_n)\}$ de tal forma que p_1 y p_k son los vértices de P con la menor y mayor coordenada y respectivamente. Como P es convexo, $y_1 < y_2 < \cdots < y_k$ y $y_1 < y_n < y_{n-1} < \ldots, y_k$. Nuestro preprocesado consiste en crear dos listas ordenadas L_1 y L_2 que contienen $y_1 < y_2 < \cdots < y_k$ y $y_1 < y_n < y_{n-1} < \ldots, y_k$ respectivamente. Para detectar las intersecciones de cualquier linea horizontal $y = y_0$ con la frontera de P procedemos de la siguiente forma:

1. Si $y_0 < y_1$ o $y_0 > y_k$ la línea $y = y_0$ no intersecta P. Si esto no sucede, proseguir con el siguiente paso.

2. Encontrar lo elementos consecutivos y_i, y y_{i+1} en L_1 y y_j, y y_{j-1} en L_2 tales que $y_i \leq y_0 \leq y_{i+1}$ $y_j \leq y_0 \leq y_{j-1}$. La horizontal $y = y_0$ intersecta la frontera de P en las aristas $p_i - p_{i+1}$, y $p_j - p_{j-1}$.

El paso 2 puede ser implementado en tiempo logaritmico utlizando una busqueda binaria en una lista ordenada.

Por tanto teniendo L_1 y L_2 podemos responder al problema de inclusión de un punto $p = (x_0, y_0)$ en P en tiempo logaritmico como sigue: detectar las intersecciones de la horizontal $y = y_0$ con la frontera de P. Si una de estas intersecciones ocurre precisamente en (x_0, y_0) el punto pertenece a P. De otra forma detectar cuantas de estas intersecciones están a la derecha del punto (x_0, y_0). si es un múmero impar (x_0, y_0) pertenece a P, de otra forma no.

2.3 Ejercicios

Ex. 2.1 Escribir una subrutina en algún lenguaje de programacion para determinar si dos segmentos de recta se intersectan.

Ex. 2.2 Escribir una subrutina que determine si un punto dado pertenece a un segmento de linea.

Ex. 2.3 Escribir un programa que determine si un punto p pertenece a un polígono P con n puntos.

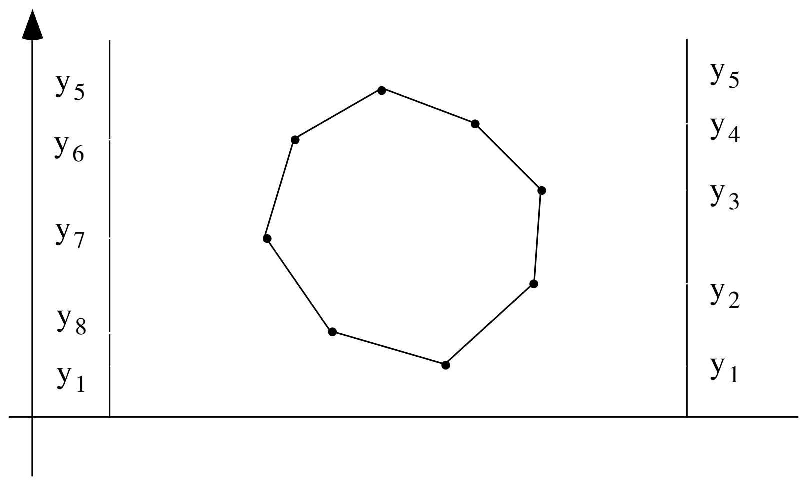

Figura 2.4: Dividiendo la frontera de un polígono convexo en la fromtera deracha y la izquierda.