

# Pauta Lectura 1 CC60Q: Geometría Computacional

Profesora: Nancy Hitschfeld Kahler. Ayudante: Diego Díaz Espinoza

August 23, 2009

## Abstract

bsection

### 1. Pregunta uno

- (a) Principalmente porque en las soluciones teóricas se asume un modelo aritmético exacto (asociado a números reales), algo que es inalcanzable (actualmente) en computadoras al momento de implementación.
- (b) Dado que los algoritmos geométricos se basan en decisiones que manejan el flujo de los mismos y dado que el computador trabaja actualmente con representaciones o aproximaciones de los números reales pueden darse casos en los que se tomen decisiones incorrectas; conduciendo a salidas incorrectas o incluso caídas de los algoritmos.
- (c) Caídas del algoritmo, estados inconsistentes, conjuntos de resultados (salidas) erróneas y loops infinitos.

### 2. Pregunta dos **Selectivos**: Se llaman selectivos aquellos que seleccionan un sub-conjunto del conjunto de entrada y lo devuelven como conjunto de salida.

**Constructivos**: Se llaman constructivos aquellos que devuelven como salida un conjunto nuevo de objetos geométricos usando nuevos valores numéricos calculados a partir del conjunto de entrada.

**Parte combinatorial**: Es la parte semántica de la salida y la entrada. Es decir, otorga un significado geométrico al conjunto de salida y entrada.

**Parte numerica**: Es la parte sintáctica de la salida y la entrada. Es decir, consta sólo de componentes que carecen de un significado para los conjuntos de salida y de entrada.

Ejemplos de la lectura: dos problemas clásicos. Convex-hull e intersección de segmentos de línea.

### 3. Pregunta tres

**Predicados Geométricos**: Son test condicionales que dirigen el flujo de los algoritmos geométricos. Se componen de primitivas geométricas (un sub-conjunto de operaciones que cubren la mayoría de los cálculos en

los algoritmos geométricos). Testean propiedades de objetos geométricos básicos.

Ejemplos de la lectura: intersección de segmentos de línea, testear si una secuencia de puntos define un giro a la derecha o si un punto está dentro o sobre el círculo definido por tres puntos.

#### 4. Pregunta cuatro

- (a) 49 Números reales distintos (24 positivos, 24 negativos y el cero).
- (b) Ver .odt adjunto.
- (c) 0.5
- (d) Según la lectura debería ser 4.0. Pero existe un error ya que al escribirlos sólo se puede llegar al 3.75
- (e) No. ( $1.110 * 2 = 3.5$ ,  $1.111 * 2 = 3.75$ )
- (f) Sí.  $1.010 * 2 = 2.5$
- (g) Error absoluto  $|3.6 - 3.5| = 0.1$ .
- (h) Error relativo  $\frac{|3.6-3.5|}{3.6} = 0.027$
- (i) 0.0625
- (j) se separan cuadráticamente entre sí (ver .odt adjunto). Notar que al graficarlos la distancia entre cada punto en representación decimal aumenta cuadráticamente. En el gráfico de diferencias se notan las pendientes de cada tramo (cada tramo es un  $e$  distinto).

#### 5. Pregunta cinco

- (a) No hacer cálculos que luego no se usarán (“lazy-evaluation”). Disminuye la posibilidad de errores en el flujo del programa a la vez que acelera al ejecución de los algoritmos.
- (b) “Floating-point filter” o filtros de punto flotante. La idea es eliminar (o filtrar) aquellos cálculos donde la computación de punto flotante da resultados incorrectos. Se basa en tomar un límite para el error de los cálculos con aritmética de punto flotante y compararlo con los valores numéricos calculados. Si no se puede decidir con el límite y el valor numérico entonces sólo en ese caso se usa aritmética exacta.
- (c) Filtros dinámicos el límite del error “on the fly” y de forma paralela a la evaluación de punto flotante. Los filtros estáticos calculan los límites de los errores a priori basados en información del hardware y el s.o. Los estáticos hacen las operaciones aritméticas más eficientes mientras que los dinámicos permiten que los cálculos de punto flotante pasen los test.
- (d) La diferencia es que se comportan distintos en caso de que las aproximaciones calculadas den signos incorrectos. En este caso, los filtros estáticos invocan evaluaciones más costosas, mientras que epsilon-urísticos asumen un valor y no incurren en más computaciones.

#### 6. Pregunta seis

- (a) Es cuando un problema no tiene solución única o esta es ambigua. Se da en casos en que un algoritmo contiene test de signos con resultados cero (no puede decidir). Los casos degenerados se pueden traspasar al implementador del algoritmo (como la mayoría de las publicaciones lo hacen).
- (b) Los problemas que causan son de robustez, es decir: el hecho de llegar a una solución correcta desde una entrada con casos degenerados.
- (c) La lectura menciona: cuatro puntos cocirculares (como Voronoi por ejemplo), tres puntos colineales o dos puntos con las mismas ordenadas.