ApuntesExamenFinalSalas.pdf

Cristinasj

Lógica y Métodos Discretos

1º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Apuntes examen final LMD

Tema 1: recurrencia

- Ejercicio 1. Ecuacion de recurrencia

Lineal homogénea

$\sum_{j=0}^{k} a_j x_{n-j} = 0$

Dada la recurrencia (de orden k)
$a_0 x_n + a_1 x_{n-1} + ... + a_k x_{n-k} = 0$

Su ecuación característica es
$a_0 x^k + a_1 x^{k-1} + ... + a_{k-1} x + a_k = 0$

Se buscan sus raices α_0, α_1 ... α_k

Solucion general: $g_n = A\alpha_0^n + B\alpha_1^n + ... + Z\alpha_k^n$

Para la solución particular usamos las condicioenes iniciales

Caso especial: numeros complejos

Caso especial: multiplicidad de las raíces
$x_n = A\alpha^n + Bn\alpha^n + Cn^2\alpha^n ...$

Lineal no homogénea

$\sum_{j=0}^{k} a_j x_{n-j} = f(x)$

- a_f es el coeficiente de f(x)
- m es el grado de f(x)
- p(x) es el polinomio característico asociado de la recurrencia

La ecuacion caracteristica de su homogénea asociada es:

$$p(x)(x - a_f)^{m+1}$$

Se buscan sus raices α_0, α_1 ... α_k

Solucion general para la homogénea asociada: $g_n = A\alpha_0^n + B\alpha_1^n + ... + (Cn^2 + Dn + C)a_f^n)$

Se calcula la particular

Se calcula la solución general de la sucesión:

Tema 2: Bool

-Ejercicio 2: Función booleana

Forma canónica disyuntiva

Suma de conjunciones

- Usamos De Morgan hasta que * afecte solo a las variables
- Usamos propiedad distributiva hasta que quede suma de conjunciones
- Usamos

$$u = u1 = u(x + x*) = ux + ux*$$

para que todas las variables aparezcan en todos los monomios
- Usamos idempotencia para eliminar repeticiones

$$a + a + b = a + b$$

- Lo escribimos en forma de $m_0 + m_2 ...$

Forma disyuntiva no simplificable
- Ordenamos monomios y variables
  1) Eliminar monomio: $f = u + R = R$ cuando $uR = u$
  2) Eliminar literal: $f = u + R = u' + R$ cuando $u'f = u'$

Petrick

Forma disyuntiva reducida
Toene todos los implicantes primos Karnaugh

Forma canónica conjuntiva
Implicantes primos nucleares/esenciales
Si se quita, se queda un 1 sin cubrir
Quine
Algoritmo de Petrick

Tema 3: Grafos

- Ejercicio 3: Demolicion-reconstruccion

Demolición:

- Se elige el vértice con mayor grado. Se pone a 0 y se disminuyen en 1 tantos vértices como indique su grado.
- Si se llega a una sucesión de ceros, la sucesión es gráfica
- Si hay un número mayor que el mumero de elementos no nulos de la sucesión, no es gráfica

Reconstrucción:

- Se dibuja el grafo solo con sus vértices
- Se miran las sucesiones generadas por el algoritmo de demolición para reconstruirlo
- Se mira el vértice usado como pivote en la fila superior con los vértices disminuidos

Teorema de Havel-Hakimi

Dado un grafo, existe otro con la misma sucesión gráfica que cumple que los bértices de mayor grado son adyacentes. Hay grafos distintos que dan lugar a la misma sucesión gráfica.

- Ejercicio 4: Polinomio cromático

G (lado e une vértices u y v)
G_e quitamos lado e
G'_e unimos u y v

Algoritmo de la suma:

$$p(G_e, x) = p(G, x) + p(G'_e, x)$$

Algoritmo de la resta:

$$p(G, x) = p(G_e, x) - p(G'_e, x)$$

Polinomios cromáticos conocidos de algunos grafos:

Completo: $P(K_n, x) = x^{\underline{n}}$

Camino: $P(P_n, x) = x(x-1)^{n-1}$

Número cromático: menor exponente descendente

- Ejercicio 5: Prim o Kruskal

Kruskal:

BUILDING-UP/CONSTRUCTIVO
- Se ordenan los n vértices de menor a mayor
- Se eligen n - 1 lados de forma que no se formen ciclos

CUTTING-DOWN/DESTRUCTIVA
- Se ordenan los vértices de mayor a menor
- Se descartan $l - (n-1) = l - n + 1$ lados de forma que se rompan ciclos

Prim:

$$T = \{v\}$$

$$E = \{\}$$

Se añade u a T y e a E:
- u no esté en T
- u sea adyacente a alguno de T
- e no forme ciclo con E
- e sea de peso mínimo
- se eligen n - 1 lados

- Ejercicio 6: Recorrido de árboles

Preorden:
Se empieza por la raiz y se recorre cada subárbol hijo como si fuera otro árbol, de izquierda a derecha

Postorden:
Se recorren los subarboles hijos en postorden de izquierda a derecha y la raiz después

Inorden:
Se recorre el subarbol hijo en inorden, después la raiz y el resto de subárboles hijos

Top-down:
De arriba a abajo y de izquierda a derecha

·Bottom-up:
Se eliminan las hojas de arriba a abajo y de izquierda a derecha

- Ejercicio 7: Árboles etiquetados

Código Prüfer:

Se añade al código la etiqueta del adyacente a la hoja con menor etiqueta y se suprime la hoja hasta que queden dos vértices

Generación de árbol:

- El código tiene n - 2 números. Se escribe el grafo de n vértices vacío.
- Se crea la lista del código y el conjunto de vértices
- Se une el primer elemento del código y el vértice con menor etiqueta que no está en el código. Se tacha el elemento del código y del conjunto de vértices.
- Cuando el código sea vacío y el conjunto de vértices tenga dos, se unen.

Tema 4: Lógica proposicional

- Ejercicio 8: Turulandia

Polinomios de Gegalkine:

- $a \bigoplus b = a + b$

- $a \vee b = a + b + ab$

- $a \wedge b = ab$

- $a \to b = 1 + a + ab$

- $a \leftrightarrow b = 1 + a + b$

- $\neg a = 1 + a$

- Ejercicio 9: Davis Putnam

Forma clausulada:

1) $\alpha \leftrightarrow \beta = (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$

2) $\alpha \rightarrow \beta = \neg\alpha \vee \beta$

3) De Morgan para interiorizar la negación

4) Se busca la multiplicación de sumas

- $\alpha \vee (\beta_1 \wedge \beta_2) = (\alpha \vee \beta_1) \wedge (\alpha \vee \beta_2)$
- $(\alpha_1 \vee \alpha_2) \wedge \beta = (\alpha_1 \vee \beta) \wedge (\alpha_2 \vee \beta)$

5) Eliminación de redundancias

- Idempotencia:
  $\alpha \vee \alpha = \alpha$ y $\alpha \wedge \alpha = \alpha$

- Absorción:
  $(\alpha \vee \beta) \wedge \alpha = \alpha \vee \beta$

Implicación semántica:
Lo transformamos en un problema de insatisfacibilidad.

$T \models \alpha$ y $T \cup \{\neg\alpha\}$

Algoritmo:

- Buscamos los literales que están solos y hacemos que valgan 1

- Buscamos los literales sin opuesto y hacemos que se vayan

- Si no hay, ramificamos haciendo que cualquier liretal valga tanto 0 como 1

Si se llega a:
- Conjunto vacío $\rightarrow$ conjunto satisfacible (Demostración falsa, hipótesis verdaderas)
- Contiene clausula vacía $\rightarrow$ conjunto insatisfacible (La demostración es cierta)

- Ejercicio 10: Resolución

Un conjunto de cláusulas es insatisfacible si hay una deducción por resolución de la cláusula vacía

Saturación:
Vamos calculando todas las resolventes posibles en el conjunto. Las nuevas con las anteriores y entre sí.

Gegalkine:

Multiplicar las premisas por la negación de la conclusión debe dar 0

Tema 5: Formas normales

- Ejercicio 11: FNP FNS FC

Forma normal prenexa:

$$C_1x_1...C_nx_n\Phi$$

- $\forall xa \to b \equiv \exists(a \to b)$ si x no está libre en b
- $\exists xa \to b \equiv \forall x(a \to b)$
- Y el resto de equivalencias de la chuleta oficial

Forma normal de Skolem:

$$\forall_1x_1...\forall_nx_n\Phi$$

- Si el $\exists$ está fuera, se cambia la variable por una constante
- Si está detrás de uno o varios $\forall$ se cambia por una función de esos

Forma clausulada:

- Se deja la formula sin cuantificadores de forma clausular
- Se distribuyen los cuantificadores

Si ves algún error, mi telegram es @cristinasj

Algunas equivalencias lógicas.

Lógica proposicional.

1. $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \land (\psi \to \varphi)$
2. $\varphi \to \psi \equiv \neg\varphi \lor \psi$
3. $\neg(\varphi \lor \psi) \equiv \neg\varphi \land \neg\psi$
4. $\neg(\varphi \land \psi) \equiv \neg\varphi \lor \neg\psi$
5. $\neg\neg\varphi \equiv \varphi$
6. $\varphi \lor \psi \equiv \psi \lor \varphi$
7. $\varphi \land \psi \equiv \psi \land \varphi$
8. $\varphi \lor (\psi \lor \chi) \equiv (\varphi \lor \psi) \lor \chi$
9. $\varphi \land (\psi \land \chi) \equiv (\varphi \land \psi) \land \chi$
10. $\varphi \land (\psi \lor \chi) \equiv (\varphi \land \psi) \lor (\varphi \land \chi)$
11. $\varphi \lor (\psi \land \chi) \equiv (\varphi \lor \psi) \land (\varphi \lor \chi)$

Lógica de predicados.

1. $\forall x\varphi \equiv \varphi$ si x no está libre en φ.
2. $\exists x\varphi \equiv \varphi$ si x no está libre en φ.
3. $\neg\forall x\varphi \equiv \exists x\neg\varphi$.
4. $\neg\exists x\varphi \equiv \forall x\neg\varphi$.
5. $\forall x\varphi \equiv \forall y\varphi(x|y)$ si y ni ningún cuantificador de y aparecen en φ.
6. $\exists x\varphi \equiv \exists y\varphi(x|y)$ si y ni ningún cuantificador de y aparecen en φ.
7. $\forall x\varphi \to \psi \equiv \exists x(\varphi \to \psi)$ si x no está libre en ψ.
8. $\exists x\varphi \to \psi \equiv \forall x(\varphi \to \psi)$ si x no está libre en ψ.
9. $\varphi \to \forall x\psi \equiv \forall x(\varphi \to \psi)$ si x no está libre en φ.
10. $\varphi \to \exists x\psi \equiv \exists x(\varphi \to \psi)$ si x no está libre en φ.
11. $\forall x\varphi \lor \psi \equiv \forall x(\varphi \lor \psi)$ si x no está libre en ψ.
12. $\forall x\varphi \land \psi \equiv \forall x(\varphi \land \psi)$ si x no está libre en ψ.
13. $\exists x\varphi \lor \psi \equiv \exists x(\varphi \lor \psi)$ si x no está libre en ψ.
14. $\exists x\varphi \land \psi \equiv \exists x(\varphi \land \psi)$ si x no está libre en ψ.
15. $\forall x\varphi \to \exists x\psi \equiv \exists x(\varphi \to \psi)$.
16. $\forall x\varphi \land \forall x\psi \equiv \forall x(\varphi \land \psi)$.
17. $\exists x\varphi \lor \exists x\psi \equiv \exists x(\varphi \lor \psi)$.