Universidad de Guadalajara

Computación Tolerante a Fallas

Ciclo 2025a

Notas del Curso - v0.544

Prof. Jesús Fco. Sánchez Blanco
jesus.sblanco@academicos.udg.mx

20 de marzo de 2025

Consideraciones Generales

Materia: Computación Tolerante a Fallas.
Clave: CC411
Profesor: Ing. Jesús Francisco Sánchez Blanco.
Horario: Martes y Jueves 19:00hrs - 21:55hrs

Evaluación

El curso se evaluará de la siguiente forma:

Tareas	20 %
Programas	20 %
Examenes	40 %
Presentaciones	20 %

Reglas generales

- Toma con la mayor seriedad y responsabilidad el curso.
- No hacer trabajos "copy-paste".
- Entregar los trabajos a tiempo y legibles.
- Entregar los trabajos con la suficiente información para identificar el objetivo de estos y a sus autores.
- Intentar agregar una aportación al trabajo. Ser prudentes con el uso de la IA.
- Con 11 faltas en el curso se pierde derecho a examen ordinario.
- El no trabajar durante la clase o no responder cuando se le nombre, causará falta en esa clase.
- No hay justificación de faltas.

Actitud esperada

- Trabaja duro durante el curso. Date la oportunidad de demostrarte que puedes aprender más allá de lo elemental.
- Pon atención a la clase. No pierdas detalle alguno.

- No quedarse con dudas.
- Promover el trabajo en equipo.
- Piensa con proyección a futuro.

1. Confiabilidad y Tolerancia a Fallas

Hoy en día el mundo de la computación ha tenido un amplio desarrollo. Prácticamente no existe actividad del ser humano en la que no se incorpore a la computación como herramienta de ayuda en su desempeño. Así entonces, hoy en día los sistemas son cada vez más grandes y más complejos.

En general, los sistemas tienden a incorporar fallas que pueden contraer defectos. Las fallas pueden tener muchas causas. Las fallas pueden generar errores y estos a su vez, pueden

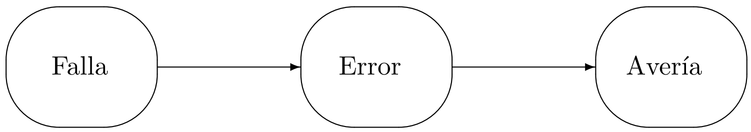

Figura 1: Fallas, errores y averías.

La norma IEEEStd 610.12-1990 [3] define un glosario de la terminología para la Ingeniería de Software. Para los definiciones de falla, error y avería (fallo) tenemos que:

- Falla (Fault): (1) A defect in a hardware device or component. (2) An incorrect step, process, or data definition in a computer program.
- Error (Error): (1) The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically value or condition.
- Avería (Failure): The inability of a system or component to perform its required functions within specified performance requirements.

El error de un componente puede propagarse como una falla a un componente contiguo.

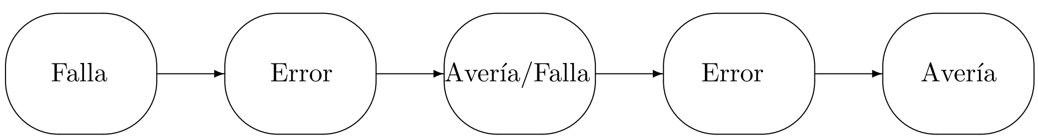

Figura 2: Propagación de una falla.

Definición 1 *Se dice que un sistema es* ***tolerante a fallas*** *(fault-tolerant), si sus programas pueden ser propiamente executed despite the occurrence of logic faults [1].*

Definición 2 *La* ***latencia de un fallo*** *(fault latency) es el tiempo que transcurre desde que se produce la falla (fault) hasta que se manifiesta el error.*

Definición 3 *La* ***latencia de un error*** *es el tiempo que transcurrido entre la aparición de un error y la manifestación de ese error en el exterior del sistema.*

Algunos conceptos pueden confundirse.

Tolerancia a fallas: Significa evitar averías ante la presencia de fallas.

Prevención de fallas: Significa prevenir la ocurrencia o la inserción de fallas.

Eliminación de fallas: Significa reducir el número y la severidad de las fallas.

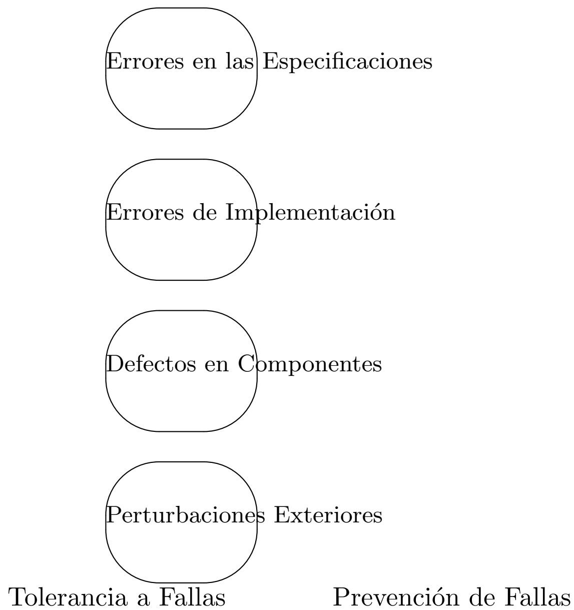

Figura 3: El contexto de la tolerancia a fallas.

Pronóstico de fallas: Significa estimar el número actual, la incidencia futura y las posibles consecuencias de las fallas.

Fechas importantes:

1958 La NASA propone el reto al Laboratorio de Caltech de construir un sistema que soporte fallos para las misiones espaciales.

1967 Definición de los sistemas tolerantes a fallos - Primer artículo.

1971 Primer IEEE International Symposium on Fault-Tolerant Computing (FTCS-1) takes place in Pasadena, CA, USA, with JPL support. (The 42nd,now “DSN” is in Boston this June).

1982 Paper intermedio sobre tolerancia a fallas.

2004 Basic Concepts and Taxonomy of Dependable and Secure Computing.

1.1. Capacidad de Proceso y Especificación de Operación

Un producto se sustenta en un conjunto de requerimientos y especificaciones.
La funcionalidad u operación de un proceso debe ser correctamente especificada. Para sistemas complejos, las especificaciones son incompletas.
Especificaciones Erróneas.
Errores de Implementación.

1.2. Clasificación de Avižienis

2004 Clases de Fallas Elementales [2].

Fase de Creación u Ocurrencia {
- Fallas de Desarrollo - Occur during a) system development, b) maintenance during the use phase
- Fallas Operacionales - Occur during service delivery of the use phase.

Límites del Sistema {
- Fallas Internas
- Fallas Externas

$$\text{Causa Fenomenológica} \begin{cases} \text{Fallas Naturales} \\ \text{Fallas Hechas por Humanos} \end{cases}$$

$$\text{Dimensión} \begin{cases} \text{Fallas de Hardware} \\ \text{Fallas de Software} \end{cases}$$

$$\text{Objetivo} \begin{cases} \text{Fallas Maliciosas} \\ \text{Fallas No Maliciosas} \end{cases}$$

$$\text{Intento} \begin{cases} \text{Fallas Deliberantes} \\ \text{Fallas No Deliberantes} \end{cases}$$

$$\text{Capacidad} \begin{cases} \text{Fallas Accidentales} \\ \text{Fallas de Incompetencia} \end{cases}$$

$$\text{Persistencia} \begin{cases} \text{Fallas Permanentes} \\ \text{Fallas Transitorias} \end{cases}$$

Se proponen tres grupos de fallas:

Fallas de Desarrollo Se incluyen todas las clases de fallas que ocurren durante el desarrollo.

Fallas Físicas Se incluyen todas las clases de fallas que afectan al hardware.

Fallas de Iteracción Se incluyen todas las fallas externas.

1.3. Errores de Diseño en Software

Entre los errores de SW más famosos podemos citar a:

- 1962. Mariner 1 (Julio 28, 1962). Mariner 1, the first American spacecraft intended for Venus,
- 1983. The Vancouver Stock Exchange (1983).
- 1985. THERAC-25. Thorac-25 Radiation Therapy Machine (1985-1987)
- 1987. Rounding error costs DHSS 100 million pounds (1987).
- 1988. Morris Worm (1988): The Teenager Who Brought the Internet to its Knees.
- 1991. American patriot missile failure (February 25, 1991).
- 1993. London Stock Exchange's Taurus Project (1993-1994)
- 1996. ARIANE 5. Ariane 5 Rocket Explosion (June 4, 1996)
- 1998. Mars Climate Orbiter (1998).
- 1999. NASA's Mars Climate Orbiter (1999)
- Y2000 (Y2K). Y2K Bug (1999-2000)
- 2000. Microsoft's Windows ME (Millennium Edition) (2000)
- 2003. Norwegian Tax Administration's ELSA Project (2003-2008):
- 2004. Mars Spirit Rover File System Error (2004)
- 2007. Malaysian man Phone bill for $218 trillion (2007)
- 2008. Heathrow Airport Terminal 5 Opening (2008)

- 2009. Herschel Space Observatory Software Issue (2009)
- 2009. T-Mobile Sidekick Data Loss (2009)
- 2009. Toyota Acceleration Issues (2009-2010)
- 2011. Mt. Gox Bitcoin Exchange Hack (2011)
- 2011. Cybersecurity has become more critical than ever before in recent years.
- 2011. The Fukushima Daiichi Nuclear Disaster (2011)
- 2011. Sony PlayStation Network Outage (2011)
- 2012. Knight Capital Group Trading Loss (2012)
- 2012. SpaceX CRS-1 Mission Failure (2012)
- 2012. Olympic Games Ticketing System (2012)
- 2013. Healthcare.gov Launch Issues (2013)

1.4. Errores de Diseño en Hardware

Entre los errores de HW más famosos podemos citar a:

- Intel Chipset i820.
- 1994. El bug "FDIV"del Intel Pentium. Pentium FDIV Bug (1994)
- El bug "translation-lookaside buffer"del AMD Barcelona.

2. Modelado de Ruido y Fallas Lógicas

Señales – Naturaleza – Medidas.
Naturaleza: Mecánicas, eléctricas, electromagnéticas.

Tipo	Medida
Voltaje	Volts
Corriente Eléctrica	Amperes
Potencia Eléctrica	Watts
Potencia Óptica	OLTs
Intensidad de Radiación	

Cuadro 1: Ejemplo de naturaleza de señales.

Sea x una cantidad mayor a cero ($x > 0$). Esta cantidad puede expresarse en decibeles a través de la conversión:

$$x_{dB} = 10 \log_{10} x. \tag{1}$$

Ejemplo 1 *En Atotonilco el Alto Jalisco, la estación de radio XHHE-FM La Z trasnmite a una frecuencia de 105.5 MHz y a una potencia de 6kW. ¿ Cuál es su potencia de transmisión en decibeles?*
Dado que su potencia de transmisión es de 6×10^3 *Watts, usando la ecuación (1) tenemos que:*

$$10 \log_{10} 6000 = 37{,}7815\, dB.$$

Ejemplo 2 *Una señal de potencia eléctrica de* $0{,}1$ *dB se transmite por un cable coaxial. ¿ Cuál es su potencia de transmisión?*
Si despejamos el valor de x de la ecuación (1) tenemos que:

$$x = 10^{\frac{x_{dB}}{10}}.$$

Sustituyendo $x_{dB} = 0{,}1$ *en la ecuación anterior:*

$$x = 10^{\frac{0{,}1}{10}} = 10^{0{,}01} = 1{,}0233\, Watts.$$

Señales continuas en tiempo. Amplitud. Frecuencia. Fase. Funciones continuas. Espectro de frecuencia.

2.1. El Ruido como una Señal del Sistema

Ruido: Todo aquello que afecta el valor de una señal.
Relación señal-ruido.

Definición 4 *Sea S el valor de una señal y N el valor del ruido. La relación señal-ruido (signal-noise ratio) se define como:*

$$SNR = \frac{S}{N} \tag{2}$$

Ejemplo 3 *Sea una señal de 38 Volts con una señal constante de ruido de 0.5 Volts. Para este caso la relación señal-ruido es:*

$$SNR = \frac{38}{0{,}5} = 76.$$

Ejemplo 4 *Sea una señal de 10 dB con un ruido de 0.1 dB. En este caso para obtener la relación señal-ruido obtenemos la amplitud en unidades reales de la señal, como ésta es de 10 dB, entonces:*

$$10 \log_{10} |S| = 10 \Rightarrow |S| = 10^1 = 10.$$

Similarmente se hace el cálculo para el ruido y como esta es de 0.1 dB, entonces:

$$10 \log_{10} |R| = 0{,}1 \Rightarrow |R| = 10^{0{,}01} = 1{,}0233.$$

De esta forma, la relación señal ruido será:

$$SNR = \frac{10}{1{,}0233} = 9{,}7723.$$

El valor promedio de una señal continua en el tiempo $f(t)$, en un intervalo de tiempo $[t_1, t_2]$, se calcula como:

$$\bar{f}(t) = \frac{1}{(t_2 - t_1)} \int_{t_1}^{t_2} f(t)\, dt. \tag{3}$$

La potencia total promedio para una señal $f(t)$ se define como:

$$P = \int_{-\infty}^{\infty} f(t)\, dt. \tag{4}$$

El valoe eficaz de una señal continua en el tiempo $f(t)$, en un intervalo de tiempo $[t_1, t_2]$, se calcula como:

$$f_{rms}(t) = \sqrt{\frac{1}{(t_2 - t_1)} \int_{t_1}^{t_2} f^2(t)\, dt}. \tag{5}$$

Ruido aditivo

$$f(t) + r(t).$$

Ruido multiplicativo

$$f(t)\, r(t).$$

Ejemplos de ruido.

- Ruido Térmico (Ruido de Johnson)
- Ruido de Disparo
- Ruido Flicker
- Ruido de Ambiente

Técnicas para reducir ruido en hardware

- Aterrizaje y blindaje.
- Amplificadores de Diferencia e Instrumentación
- Filtraje
- Modulación
- Signal Choppin

2.2. Ruido de Disparo (Señales Estrambóticas)

Una fuente de corriente en la cual el pasaje de cada portador de carga es un evento estadísticamente independiente, más que un flujo steady de muchos portadores de carga, necesariamente entrega un "noisy" en corriente, i.e. una corriente que fluctua about an average value. Fluctuations of this kind are called "shot noise". The magnitude of such fluctuations depends on the magnitude of the charges on the individual carriers. Thus a measurement of the fluctuations should, in principle, yield a measure of the magnitude of the charges.

El ruido de disparo aparece cuando los electrones cruzan una unión (en semiconductores).

$$i_{rms} = \sqrt{2I_e \Delta f}. \tag{6}$$

2.3. Ruido Térmico

El ruido térmico es la causa de ruido más importante en los circuitos eléctricos y por consecuencia, está presente en todos los componentes de los sistemas de comunicaciones que incluyen circuitos eléctricos o electrónicos, particularmente en los receptores en que los niveles de señal pueden ser comparables a los de ruido térmico generado en los circuitos del propio receptor. Su origen es el movimiento aleatorio de los electrones libres en los conductores y semiconductores.

La densidad espectral de ruido depende de la temperatura y está dada por:

$$N_0 = kT \quad \text{watt/Hz}. \tag{7}$$

Donde:
T = Temperatura en grados kelvins = Temperatura ambiente en ºC + 273.
k = Constante de Boltzmann = $1{,}38x10^{-23}$ watt/ºK-Hz.

Un ejemplo de ruido térmico son las fluctuaciones de voltaje sobre un resistor eléctrico con resistencia R.

$$N_{TH} = \sqrt{4kTRB}. \tag{8}$$

Donde:
N = Ruido en volts. k = Constante de Boltzmann = $1{,}38 \times 10^{-23}$ watt/ºK-Hz. T = Temperatura en kelvins = Temperatura ambiente en ºC + 273. R = Resistencia en ohms. B = Ancho de banda.

2.4. Ruido de Cuantización

Problema 1 *Encuentre lo siguiente:*

1. *Sea una señal que tiene una amplitud que oscila entre 0 a 12 Volts. Si todo este intervalo se divide en 8 niveles, escriba el valor máximo, el valor mínimo y el valor medio de cada subintervalo obtenido si estos fueran del mismo tamaño.*

2. *Calcule el error de cuantización.*

2.5. Sistemas Estocásticos

Variable aleatoria.

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{9}$$

Ejemplo 5 *Sea una densidad de probabilidad gaussiana con una media $\mu = 0$ y una desviación estándard $\sigma = 1$. Calcular la probabilidad cuando:*

$$P(-\sigma \leq x \leq \sigma).$$

Para esto:

$$P(-\sigma \leq x \leq \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\sigma}^{\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \frac{1}{\sqrt{2\pi}} \int_{-1}^{1} e^{-\frac{x^2}{2}} dx.$$

Calculando este valor por... https://calculadorasonline.com/calculadora-de-distribucion-nomal-campana-de-gauss/

$$P(-\sigma \leq x \leq \sigma) = 0{,}6827.$$

Ejemplo 6 *Proceso aleatorio binario.*

El teorema del Límite Central establece que si muchos procesos aleatorios independientes con funciones de distribución de probabilidad arbitrarias son sumadas, la función de distribución de probabilidad de la suma se aprocima a una distribución normal.

2.6. El Ruido Blanco

Los valores de la señal en dos tiempos diferentes no guardan correlación estadística. El ruido blanco es un caso especial de un proceso estocástico WSS en el cual las variables aleatorias que lo forman no están correlacionadas

El ruido blanco Gaussiano se refiere a un ruido con espectro de frecuencia (o de potencia espectral) uniforme y amplitud con distribución Gaussiana.

3. Verificación en Sistemas Digitales

Sea $P(x_i)$ la probabilidad de ocurrencia de un evento x_i. Su cantidad de información está definida como:

$$I_i = \log_n \frac{1}{P_i} \quad \text{n-arias.} \tag{10}$$

Ejemplo 7 *Sea un registro de 10 bits que tiene los siguientes datos* $r = 0000100101_b$. *La probabilidad de que un bit sea cero será de* $7/10$ *y por lo tanto su cantidad de información será de:*

$$I(r_i = 0) = \log_2 \frac{1}{\left(\frac{7}{10}\right)} = \log_2 \frac{10}{7} = 0{,}5146\ bits.$$

Analogamente, la cantidad de información del símbolo 1 en el registro será de:

$$I(r_i = 1) = \log_2 \frac{1}{\left(\frac{3}{10}\right)} = \log_2 \frac{10}{3} = 1{,}7370\ bits.$$

Problema 2 *De un texto a transmitir, un mensaje* m_1 *aparece 4 de cada 10 veces. ¿Cuál es la cantidad de información de* m_1*?*

Definición 5 (Entropía) *Sean* $P_1, P_2, \ldots P_k$ *las probabilidades asociadas con* k *eventos independientes* $X = x_1, x_2, \cdots, x_k$. *La entropía* $H(X)$ *de dichos eventos se define como el promedio de su cantidad de información. Por lo tanto:*

$$H(x) = \sum_{i=1}^{k} P_i I_i = \sum_{i=1}^{k} P_i \log_n \frac{1}{P_i} \quad \textit{n-arias.} \tag{11}$$

Ejemplo 8 *Para el ejemplo anterior la entropía de los datos del registro* r *será de:*

$$H(r) = \sum_{i=0}^{1} P_i I_i = P(0) I_0 + P(1) I_1 = 0{,}70\,(0{,}5146) + 0{,}30\,(1{,}7370) = 0{,}3602 + 0{,}5211 = 0{,}8813\ bits.$$

Problema 3 *Dos registros de 8 bits tiene el siguiente valor* $r_1 = 00110101_b$ *y* $r_2 = 10000100_b$. *Cuál de los dos registros anteriores tiene más información?*

Para resolver el problema anterior, calculemos primero la entorpiía (información promedio) de cada registro.

$$H(r_1) = P(r_{1_i} = 0)I_i + P(r_{1_i} = 1)I = \frac{4}{8}\log_2 \frac{8}{4} + \frac{4}{8}\log_2 \frac{8}{4} = 1{,}0000\ bits.$$

$$H(r_2) = P(r_{2_i} = 0)I_i + P(r_{2_i} = 1)I = \frac{6}{8}\log_2 \frac{8}{6} + \frac{2}{8}\log_2 \frac{8}{2} = ...;bits$$

3.1. Definición de Canal Binario Simétrico

En un sistema, la información se transfiere de un componente a otro a través de un canal (ver la siguiente figura).

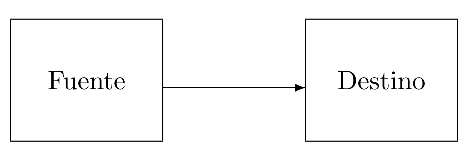

Figura 4: Esquema de Comunicación de Información

Así entonces, una fuente puede transmitir una cantidad de símbolos $X = \{x_0, x_1, \ldots .x_{n-1}\}$ codficados en unidades n-arias y con una probabilidad de aparición $P(x_0), P(x_1), \ldots, P(x_{n-1})$. Y en general, puede recibir una cantidad de símbolos $Y = \{y_0, y_1, \ldots .y_{m-1}\}$ codificados en unidades m-arias, cuya probabilidad de cada uno de los símbolos será entonces $P(y_0), P(y_1), \ldots, P(y_{m-1})$. Salvo casos excepcionales se tiene que $m = n$.

Ejemplo 9 *Una fuente trasmite el siguiente mensaje:* $m = aaeeiioouuaiu$*. Para este caso, la fuente emite 5 símbolos* $X = \{a, e, i, o, u\}$ *y sus probabilidades serán* $P(a) = P(i) = P(u) = \frac{3}{13}$ *y* $P(e) = P(o) = \frac{2}{13}$.

Al transmitirse un símbolo x_i idealmente se espera que se reciba el símbolo y_i, lo que implicaría que no existe error en la transmisión. Desafortunadamente eso sólo existe en un caso ideal. En la práctica si se transmite un símbolo x_i, existe una probabilidad $P(x_i|y_j)$ (probablilidad condicional) de que se reciba el símbolo y_j.

Un canal binario se representa como lo muestra la figura. En este caso la fuente (origen de los datos) puede transmitir dos tipos de símbolos x_0 y x_1, y puede recibir también dos tipos de símbolos y_0 y y_1.

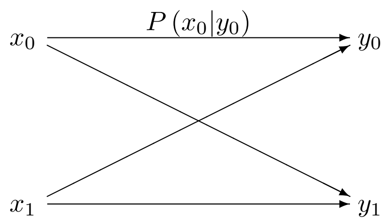

Figura 5: Canal Binario.

Cuando el símbolo x_0 se transmite, la fuente puede recibir el símbolo y_0 si la transmisión no tuviese error y y_1 si lo hubiera. De igual forma, pasa con el símbolo x_1. Para tales casos asociamos una probabilidad condicional; por lo que tenemos que: $P(x_0|y_0)$ es la probabilidad de que dado que se recibió y_0 se envió x_0.

$$P(X|Y) = \begin{pmatrix} P(x_0|y_0) & P(x_0|y_1) \\ P(x_1|y_0) & P(x_1|y_1) \end{pmatrix}.$$

Sea p una probabilidad de falla. Si $P(x_0|y_0) = P(x_1|y_1) = 1 - p$ entonces el canal binario se dice ser un canal binario simétrico.

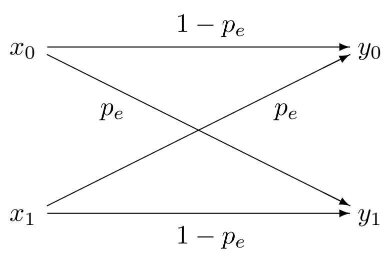

Figura 6: Canal Binario Simétrico.

Problema 4 *Una fuente* X *emite 4 mensajes con probabilidades* $1/2$*,* $1/4$*,* $1/8$ *y* $1/8$*. Encuentre su entropía* $H(X)$.

3.2. Tasa de Errores

En un canal sin errores no existe pérdida de información.

Definimos la capacidad de un canal como la cantidad de información que el canal puede soportar de manera fiable. Se sabe que la capacidad del canal es igual a la información mútua entre la fuente y el destinatario y está dada como:

$$I(X;Y) = H(X) - H(X|Y).$$

En donde $H(X)$ es la entropía de la fuente y $H(X|Y)$ es la pérdida promedio cuando se reciben todos los símbolos y tal que se mandaron los símbolos x, las cuales están dadas por:

$$H(X) = \sum_{i=0}^{n-1} P(x_i) \log_n \frac{1}{P(x_i)}$$

$$H(X|Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P(x_i|y_j) \log_n \frac{1}{P(x_i|y_j)}$$

Por lo tanto:

$$I(X;Y) = \sum_{i=0}^{n-1} P(x_i) \log_n \frac{1}{P(x_i)} - \sum_{j=0}^{n-1} \sum_{j} P(x_i|y_j) \log_n \frac{1}{P(x_i|y_j)}.$$

Desarrollando esta expresión tenemos entonces que:

$$I(X;Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P(x_i) P(y_j|x_i) \log_n \frac{P(y_j|x_i)}{\Sigma P(x_i) P(y_j|x_i)}. \tag{12}$$

Problema 5 *Para un canal simétrico binario, encontrar una expresión para $I(X;Y)$ en términos de las probabilidades de la fuente y del canal.*

Solución 1 *La expresión*

$$I(X;Y) = H(X) - H(X|Y),$$

puede escribirse como:

$$I(X;Y) = \sum_{i=0}^{1} P(x_i) \log_2 \frac{1}{P(x_i)} - \sum_{i=0}^{1} \sum_{j=0} P(x_i|y_j) \log_2 \frac{1}{P(x_i|y_j)} = ...$$

...

$$I(X;Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} P(x_i) P(y_j|x_i) \log \frac{P(y_j|x_i)}{\sum_{k=0} P(x_i) P(y_j|x_i)}.$$

La entropía de la fuente en un canal binario es:

$$H(X) = \sum_{i=0}^{1} P(x_i) \log_2 \left(\frac{1}{P(x_i)}\right) = P(x_0) \log_2 \left(\frac{1}{P(x_0)}\right) + P(x_1) \log_2 \left(\frac{1}{P(x_1)}\right).$$

Como $P(x_1) = 1 - P(x_0)$, entonces:

$$H(X) = P(x_0) \log_2 \left(\frac{1}{P(x_0)}\right) + \left[(1 - P(x_0)) \log_2 \left(\frac{1}{(1 - P(x_0))}\right)\right].$$

La pérdida de información se da por:

$$H(X1Y) = \sum_{i} \sum_{j} P(x_i|y_j) \log_2 \left(\frac{1}{P(x_i|y_j)}\right).$$

Para un canal binario simétrico se tiene que:

$$I(X;Y) = \alpha\bar{P}_e \log_2 \left[\frac{\bar{P}_e}{\alpha\bar{P}_e + \bar{\alpha}P_e}\right] + \alpha P_e \log_2 \left[\frac{P_e}{\alpha P_e + \bar{\alpha}\bar{P}_e}\right] + \bar{\alpha}P_e \log_2 \left[\frac{P_e}{\alpha\bar{P}_e + \alpha\alpha P_e}\right] + \bar{\alpha}\bar{P}_e \log_2 \left[\frac{P_e}{\alpha P_e + \bar{\alpha}\bar{P}_e}\right] =$$

$$= \left(\alpha P_e + \bar{\alpha}\bar{P}_e\right) \log_2 \left[\frac{1}{\alpha P_e + \bar{\alpha}\bar{P}_e}\right] + \left(\alpha\bar{P}_e + \bar{\alpha}P_e\right) \log_2 \left[\frac{1}{\alpha\bar{P}_e + \bar{\alpha}P_e}\right] - \left(P_e \log_2 \left(\frac{1}{P_e}\right) + \bar{P}_e \log_2 \left(\frac{1}{\bar{P}_e}\right)\right).$$

Definición 6 *La máxima capacidad del canal se alcanza cuando la informción mútua es máxima y por lo tanto:*

$$C =_{max} I\left(X;Y\right). \tag{13}$$

Resuelva lo siguiente relacionado con cantidad de información.

1. Sea un canal binario con una matriz de canal

$$P\left(Y|X\right) = \begin{pmatrix} 0{,}8 & 0{,}2 \\ 0{,}2 & 0{,}8 \end{pmatrix}.$$

   Calcule la pérdida de información si $P\left(0\right) = 1/5$ y $P\left(1\right) = 4/5$.

2. Para el problema anterior calcule la máxima capacidad del canal.

Para un canal binario se tiene entonces que la información mútua es máxima cuando... ...

3.3. Capacidad de Shannon

Sea B el ancho de banda de un canal exprezado en Hz. Sea S/N la media cuadrada (mean-square) de la relación señal-ruido (no en dB). La capacidad del canal está dada por:

$$C = B\log_2\left(1 + S/N\right). \tag{14}$$

Esta expresión implica que una transmisión libre de error es posible si no se envía información a un rate”mayor que la capacidad del canal.

Ejemplo 10 *Si $B = 3$ kHz and S/N se mantiene en 30 dB para un canal de una línea telefónica típica, la capacidad del canal C es aproximadamente 30 kbits/s.*

Dado que un canal nunca estará libre de errores, entonces es necesario agregar información (códigos).

3.4. Clasificación de Códigos Correctores

Detección de errores vs corrección de errores.

Sea $C = \{00, 01, 10, 11\}$. El tamaño de un código es la cantidad de elementos del código. En este caso el tamaño de C es 4.

Según la relación entre mensajes, los códigos se clasifican en:

- Códigos de Bloque: Se forma la palabra de cdígo m_i añadiendo información K_i en función de cada mensaje m_i.
- Códigos Convolucionales: Se opera sobre el flujo continuo de la información.

Según la estructura del código, la clasificación es la siguiente:

Ejemplos de códigos de bloques son: Reeds-Solomon, Hamming, Hadamard, Expander, Golay y Reed-Muller.

Según la disposición, los códigos se clasifican en:

- Códigos Sistemáticos: m de los n bits de la palabra de código N coinciden con los bits del mensaje M..
- Códigos No Sistemáticos: No se pueden identificar los bits de M dentro de N.

Según sus propiedades algebraícas, los códigos se clasifican en:

- Códigos Lineales : Dados dos mensajes cualesquiera m_i y m_j y sus palabras de código N_i y N_j, el mensaje $m_i + m_j$ se codifica como $N_i + N_j$.
- Códigos No Lineales: No cumplen la propiedad de la linealidad.

Las familias más conocidas de estos códigos son

$$\text{Códigos}\left\{\begin{array}{l}\text{Bloque Fijo}\left\{\begin{array}{l}\text{Lineales}\left\{\begin{array}{l}\text{Cíclicos}\\ \text{Convolución}\\ \text{Otros}\end{array}\right.\\ \text{No Lineales}\left\{\begin{array}{l}\text{Sobre Anillos}\\ \text{Otros}\end{array}\right.\end{array}\right.\\ \text{Longitud Variable}\left\{\begin{array}{l}\text{Huffman}\\ \text{Otros}\end{array}\right.\end{array}\right.$$

Figura 7: Tipos de códigos.

- – Códigos lineales: de Hamming, de Hamming extendidos, simplex, de Golay, de Reed-Muller, de Goppa geométricos.
- – Códigos cíclicos: BCH, de Reed-Solomon, de residuos cuadráticos, de Goppa clásicos.
- – Códigos no-lineales: Hadamard, Kerdock, Justesen, Preparata.

Dada una palabra codificada en binario (ejemplo 0010100_b), es posible agregarle un bit adicional según su cantidad de ceros o de unos. Así entonces tenemos estos posibles casos:

1. Agregar un 0 si la cantidad de ceros es par o un 1 si es impar.
2. Agregar un 1 si la cantidad de ceros es par o un 0 si es impar.
3. Agregar un 0 si la cantidad de unos es par o un 1 si es impar.
4. Agregar un 1 si la cantidad de unos es par o un 0 si es impar.

Nota: El cero es considerado como una cantidad par.

La siguiente tabla ilustra que bit agregar para cada uno de los casos, p_0 para el caso 1, p_1 para el caso 2, p_2 para el caso 3 (que es el caso más común) y p_3 para el caso 4.

x_1	x_0	p_3	p_2	p_1	p_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	0	1	0	1
1	1	1	0	1	0

Si la información es $D = \{00_b, 01_b, 10_b, 11_b\}$ entonces su código de paridad es $C = \{000_b, 011_b, 101_b, 110_b\}$.

Definición 7 *Sean m_i y m_j dos palabras del mismo tamaño. La distancia Hamming entre m_i y m_j se define como la cantidad de símbolos que cambian en la misma posición en m_i y m_j. Para un código de tamaño de palabra constante C, la distancia Hamming del código se define como la distancia Hamming mínima entre todos sus elementos. Sean m_i, m_j y m_k tres palabras. Los siguiente se cumple:*

1. *$d(m_i, m_j) = 0$ sý sólo si $m_i = m_j$.*
2. *$d(m_i, m_j) = d(m_j, m_i)$.*
3. *$d(m_i, m_j) + d(m_j, m_k) \geq d(m_i, m_k)$.*

Ejemplo 11 *Sea $m_i = 0101010_b$ y $m_j = 1101011_b$ dos palabra. La distancia Hamming entre m_i y m_j es dos.*

$$d(m_i, m_j) = 2.$$

Para el código $C = \{00_b, 01_b, 10_b, 11_b\}$, la distancia Hamming es $d(C) = 1$. Si ahora consideramos $C = \{000_b, 011_b, 101_b, 110_b\}$, la distancia Hamming de C es:

$$d(C) = 2,$$

ya que $d(000_b, 011_b) = 2$, $d(000_b, 101_b) = 2$, $d(000_b, 110_b) = 2$, $d(011_b, 101_b) = 2$, $d(011_b, 110_b) = 2$ y $d(101_b, 110_b) = 2$ y 2 es la distancia Hamming mínima entre los componentes de C.

Problema 6 *Para el siguiente código* $C = \{10101_b, 01010_b\}$. *Defina su longitud, su tamaño y su distancia Hamming.*

Definición 8 *Sea* C *un código de tamaño fijo y con una distancia Hamming* $d(C)$. C *es s-detectable si*

$$d = s + 1.$$

Un código s-detectable puede detectar a lo más s errores.

Ejemplo 12 *El código* $C = \{00_b, 01_b, 10_b, 11_b\}$ *tiene una distancia Hamming* $d(C) = 1$, *por lo tanto no puede detectar ningún error ya que* $s = 0$ *en:*

$$s = d - 1 = 0.$$

Definición 9 *Un código* C *de tamaño fijo, se dice ser* t *corregible si...*

$$d = 2t + 1.$$

Un código t-corregible puede corregir a lo más t errores.

Ejemplo 13 *El código* $C = \{000_b, 011_b, 101_b, 110_b\}$ *tiene una distancia Hamming* $d(C) = 2$. *Dado que* $d = 2t + 1$, *entonces:*

$$t = \frac{d-1}{2} = \frac{2-1}{2} = 0.$$

Este código sólo puede detectar un error y no puede corregir ninguno.

Problema 7 *Escriba un código de tamaño 4, longitud 8 y que corrija dos errores.*

Ejemplo 14 *El código de Hamming (7,4). Propuesto por Hamming en 1950. Consta de agregar 3 bits de paridad* p_0, p_1 *y* p_2, *a una palabra de 4 bits* $d = d_3d_2d_1d_0$, *de tal forma que cada bit de paridad tiene una dependencia de los bits de datos tal y como como se muestra en la siguiente tabla.*

1	2	3	4	5	6	7
p_0	p_1	d_0	p_2	d_1	d_2	d_3
×		×		×		×
	×	×			×	×
			×	×	×	×

Para calcular cada bit de paridad, entonces:

$$\begin{aligned} p_0 &= d_0 + d_1 + d_3 \\ p_1 &= d_0 + d_2 + d_3 \\ p_2 &= d_1 + d_2 + d_3 \end{aligned}$$

Si fuese $d = 1011_b$ *entonces* $d_0 = 1$, $d_1 = 1$, $d_2 = 0$ *y* $d_3 = 1$ *y los bits de paridad serían:*

$$\begin{aligned} p_0 &= 1 + 1 + 1 &= 1 \\ p_1 &= 1 + 0 + 1 &= 0 \\ p_2 &= 1 + 0 + 1 &= 0 \end{aligned}$$

Entonces el mensaje a transmitir, de bit de mayor a menor prioridad será:

$$m = d_3d_2d_1p_2d_0p_1p_0 = 1010101_b.$$

Supongamos ahora que el mensaje que se recibe fuera el mismo $m = 1010101_b$, *calculamos entonces:*

$$\begin{aligned} p_0' &= p_0 + d_0 + d_1 + d_3 \\ p_1' &= p_1 + d_0 + d_2 + d_3 \\ p_2' &= p_2 + d_1 + d_2 + d_3 \end{aligned}$$

Y obtenemos que:

$$\begin{array}{lclcl} p_0' & = & 1+1+1+1 & = & 0 \\ p_1' & = & 0+1+0+1 & = & 0 \\ p_2' & = & 0+1+0+1 & = & 0 \end{array}$$

lo que significa que no hay error. Si se hubiese recibido $m = 1011101_b$ *tenemos que:*

$$\begin{array}{lclcl} p_0' & = & 1+1+1+1 & = & 0 \\ p_1' & = & 0+1+0+1 & = & 0 \\ p_2' & = & 1+1+0+1 & = & 1 \end{array}$$

lo que significa que se detectó un error en la posición 4 de $m = 1011110_b$.

Problema 8 *Sea una palabra a transmitir* 1110110_2. *Agregue los bits necesarios para codificar la palabra anterior usando el código de Hamming.*

3.5. Generador Lineal Homogéneo

Sea un conjunto de datos $D = \{d_1, \ldots, d_n\}$. Un código lineal puede generarse a través de una matriz $\mathbf{G}$ de dimensión $m \times n$ con $m > n$ y con entradas $\{0,1\}$ tal que la palabra de código generada es:

$$m_i = \mathbf{G} d_i.$$

Ejemplo 15 *Paridad es un código lineal homogéneo. Si* $D = \{00_b, 01_b, 10_b, 11_b\}$ *y la matriz generadora sea*

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix},$$

entonces el código a generar es;

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \ y \ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

Por lo que el código generado por $\mathbf{G}$ *sobre* D *es* $C = \{000_b, 011_b, 101_b, 110_b\}$.

Problema 9

1. *Sea* $I = \{(x_1, x_2, x_3) \mid x_i \in \{0,1\}\}$. *Sea* $C = \{(y_1, y_2, y_3, y_4) \mid y_i \in \{0,1\}\}$. *Obtenga la matriz de codificación* T, *tal que:* $y_1 = x_1$, $y_2 = x_2$, $y_3 = x_1 + x_2$ *y* $y_4 = x_1 + x_3$.

2. *Determine la distancia Hamming del código generado por* T.

3. *Sea* m *un mensaje de longitud 8. Si le agregamos dos bits de paridad, ¿cuántos errores se pueden detectar y cuántos se pueden corregir? Justifique su respuesta.*

Problema 10 *Encuentre la matriz generadora para el código de Hamming* $(7,4)$.

Código de Reed-Muller

3.6. Generador Redundante

Una palabra de $n+1$ bits puede representarse por un polinómio:

$$P(x) = v_n x^n + v_{n-1} x^{n-1} + \cdots + v_2 x^2 + v_1 x^1 + v_0 x^0,$$

tal que $v_i \in \{0,1\}$. Por ejemplo si un byte es $m_i = 01001010_b$ entonces dicho byte se representa por el polinomio

$$P(x) = 0 \cdot x^7 + 1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0 = x^6 + x^3 + x.$$

Para generar un código sobre un conjunto de datos se debe definir un polinomio generador $G(x)$ de forma tal que si la información está dada por el polinomio $P(x)$, el código de redundancia se obtiene del residuo $R(x)$ de tal manera que

$$T(x) = x^n D(x) + \frac{x^n D(x)}{G(x)}.$$

Ejemplo 16 *Sea la información de 6 bits $m_i = 110101_b$ representada por el polinomio $D(x) = 1 \cdot x^5 + 1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0 = x^5 + x^4 + x^2 + 1$. Sea $G(x) = x^2 + 1$ el polinomio generador. Para obtener el CRC primero tenemos que multiplicar $D(x)$ por el grado del polinomio generador y tenemos que*

$$T(x) = x^2 \cdot D(x) = x^2 \cdot \left(x^5 + x^4 + x^2 + 1\right) = x^7 + x^6 + x^4 + x^2.$$

Se calcula el residuo entre $T(x)$ y $G(x)$.

$$\begin{array}{cccccccc}
1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & & & & & \\
\hline
 & 1 & 1 & 1 & & & & \\
 & 1 & 0 & 1 & & & & \\
\hline
 & & 1 & 0 & 0 & & & \\
 & & 1 & 0 & 1 & & & \\
\hline
 & & & & 1 & 1 & 0 & \\
 & & & & 1 & 0 & 1 & \\
\hline
 & & & & & 1 & 1 & 0 \\
 & & & & & 1 & 0 & 1 \\
\hline
 & & & & & & 1 & 1
\end{array}$$

Entonces el CRC $= 11$. El código a transmitir será entonces $T(x) +$ CRC: 11010111_b.

Código de Reducndancia Cíclica

1. Calcule el CRC de 01000100_2 con un polinomio generador $x^3 + x^2 + 1$.
2. ¿El código de Hamming para palabras de tamaño de 4 bits es lineal? Justifique su respuesta.

3.7. Matriz Correctora

Sea **G** una matriz generadora. Una matriz **H** es una matriz verificadora/correctora de errores si

$$\mathbf{H}\mathbf{G}^\top = 0.$$

Ejemplo 17 *La matriz generadora para el código de paridad se muestra enn el ejemplo 15. La matriz correctora correspondiente será una matriz H tal que:*

$$\mathbf{H} \begin{bmatrix} x_1 \\ x_0 \\ p \end{bmatrix} = \begin{bmatrix} x_1 \\ x_0 \\ p' \end{bmatrix};$$

en donde $p' = x_1 + x_0 + p$. Por lo tanto:

$$\mathbf{H} =$$

Ejemplo 18 *Sea la matriz generadora*

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

La matriz verificadora será entonces:

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Si los posibles datos se dan por toda la combinación de tres variables, entonces todo el código generado por $\mathbb{G}$ *se muestra en la siguiente tabla.*

d_2	d_1	d_0	x_5	x_4	x_3	x_2	x_1	x_0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	0	1
0	1	0	0	1	0	1	1	0
0	1	1	0	1	1	0	1	1
1	0	0	1	0	0	0	1	1
1	0	1	1	0	1	1	1	0
1	1	0	1	1	0	1	0	1
1	1	1	1	1	1	0	0	0

Si se reciben las palabras de código $m_0 = 100011_b$*,* $m_1 = 100111_b$ *y* $m_3 = 100010_b$*, entonces se multiplican a través de la matriz correctora, por lo que:*

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

3.8. Verificación Combinatoria

Dada una palabra codificada m_i de n-bits con 2 bits de CRC.
Self checking circuits.

3.9. Verificación Secuencial

Alternativamente al método combinatorio, es posible realizar una verificación secuencial de una palabra de código con un registro de corrimiento.

Ejemplo 19

3.10. Verificación Polinomial

Para determinar si un mensaje trae error, el mensaje se divide con el mismo polinomio con el que fue generado.

Ejemplo 20 *Sea el mensaje que se generó en el ejemplo anterior* $m = 11010111_b$ *con el polinomio*

Periodo del Reloj	Entrada $c(x)$	Estado del Registro s_2	s_1	s_0	Salida $d(x)$
0		0	0	0	
1	1	0	0	1	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	0	1	1	1
5	1	1	1	1	0
6	0	1	0	1	1
7	1	0	0	0	1

Cuadro 2: Secuencia de un registro de corrimiento [1010001].

generador $G(x) = x^2 + 1$. Checando su residuo se obtiene:

$$\begin{array}{cccccccc}
1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & & & & & \\
\hline
 & 1 & 1 & 1 & & & & \\
 & 1 & 0 & 1 & & & & \\
\hline
 & & 1 & 0 & 0 & & & \\
 & & 1 & 0 & 1 & & & \\
\hline
 & & & & 1 & 0 & 1 & \\
 & & & & 1 & 0 & 1 & \\
\hline
 & & & & & 1 & 0 & 1 \\
 & & & & & 1 & 0 & 1 \\
\hline
 & & & & & & 0 & 0
\end{array}$$

Como el residuo es 00_b se concluye que el mensaje recibido no tiene error. Si ahora el mensaje recibido fuera $m = 10010111_b$ (error en el bit 7), se obtiene que:

$$\begin{array}{cccccccc}
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & & & & & \\
\hline
 & & 1 & 1 & 0 & & & \\
 & & 1 & 0 & 1 & & & \\
\hline
 & & & 1 & 1 & 1 & & \\
 & & & 1 & 0 & 1 & & \\
\hline
 & & & & 1 & 0 & 1 & \\
 & & & & 1 & 0 & 1 & \\
\hline
 & & & & & & 0 & 1
\end{array}$$

Como el residuo es 01_b entonces se concluye que hay error.

Convolutional Codes

4. Confiabilidad del Entorno

La fiabilidad (reliability) de un sistema es una medida de su conformidad con una especificación autorizada de su comportamiento.

Definición 10 *Fiabilidad es la probabilidad de que un dispositivo realice adecuadamente su función prevista a lo largo del tiempo, cuando opera en el entorno para el que ha sido diseñado [6].*

$$MTBF = MTTF + MTTR. \tag{15}$$

MTBF: Tiempo medio entre fallas.
MTTF: Tiempo medio a fallar.
MTTR: Tiempo medio a reparar.

La confiabilidad (dependability) es una propiedad de los sistemas que permite confiar justificadamente en el servicio que proporcionan.

Sistemas en Serie: En una configuración en serie el fallo de cualquiera de sus componentes provoca el fallo del sistema. Sea X_i la probabilidad de que funcione el i-ésimo componente, la función de estructura del sistema es:

$$X_S = X_1 \cdots X_n = \Pi_{i=1}^{n} X_i.$$

La fiabilidad de un sistema en serie es:

$$F = \Pi R_i t \tag{16}$$

Sistemas en Paralelo: En una configuración en paralelo el funcionamiento de cualquiera de sus componentes provoca el funcionamiento del sistema. Sea X_i la probabilidad de que funcione el i-ésimo componente, la función de estructura del sistema es:

$$X_S = 1 - [(1 - X_1) \cdots (1 - X_n)] = 1 - \left[\Pi_{i=1}^{n} (1 - X_i)\right].$$

Fiabilidad (Reliability) Es la probabilidad that a system continues to operate correctly during a particular time interval given that it was operational at the beginning of the interval.

Availability Es la probabilidad que un sistema esté operando correctamente en un instante de tiempo dado.

Seguridad (Safety) Es la probabilidad that the system will perform in a non-hazardous way. A hazard is defined as .ª state or condition of a system that, together with other conditions in the environment of the system, will lead inevitably to an accident"[Leveson 95].

Performability Es la probabilidad that the system performance will be equal to or greater than some particular level at a given instant of time.

Mantenimiento (Maintainability) Es la probabilidad that a failed system will be returned to operation within a particular time period. Maintainability measures the ease with which a system can be repaired.

Testability Es una medida de the ability to characterize a system through testing. Testability includes the ease of test development (i.e., controllability) and effect observation (i.e., observability).

The main direct concern for fault tolerant designs is the ability to continue delivery of services
Distribución de Weibull

$$f(t) = \frac{\beta}{\eta}\left(\right). \tag{17}$$

Función de fiabilidad de Weibull.

$$f(t) = \frac{\beta}{\eta}\left(\right). \tag{18}$$

4.1. Modos de Fallos

Las fallas de funcionamiento de un sistema pueden tener su origen en:

- Una especificación inadecuada
- Errores de diseño del software
- Averías en el hardware
- Interferencias transitorias o permanentes en las comunicaciones

Tipos de Averías

```
Nunca se avería
Avería
    |
    | ------ Valor
    |            | ------- Error de Intervalo
    |            | ------- Error de Valor
    |
    | -----  Tiempo
    |              | ------- Pronto
    |              | ------- Nunca (omisión)
    |              |                | ----- Avería Silenciosa
    |              |                | ----- Parada Segura
    |              |                | ----- Avería Controlada
    |              |
    |              | ------- Avería por Retraso
    |
    | -----  Arbitrario
                    | ------ Avería Incontrolada
```

Caracterización de las Averías { ...

Figura 8: Tipos de averías.

4.2. Prevención de Fallos

Hay dos formas de aumentar la fiabilidad de un sistema:

- Prevención de fallos: Se trata de evitar que se introduzcan fallos en el sistema antes de que entre en funcionamiento
- Tolerancia de fallos: Se trata de conseguir que el sistema continúe funcionando aunque se produzcan fallos u En ambos casos el objetivo es desarrollar sistemas con tipos de averías bien definidos

Se realiza en dos etapas:

- Evitación de fallos » Se trata de impedir que se introduzcan fallos durante la construcción del sistema
- Eliminación de fallos » Consiste en encontrar y eliminar los fallos que se producen en el sistema una vez construido

Técnicas de evitación de fallos

- Hardware
  - Utilización de componentes fiables

- Técnicas rigurosas de montaje de subsistemas
- Apantallamiento de hardware

- Software
  - Especificación de requisitos rigurosa o formal
  - Métodos de diseño comprobados
  - Lenguajes con abstracción de datos y modularidad
  - Utilización de entornos de desarrollo con computador (CASE) adecuados para gestionar los componentes

Técnicas de eliminación de fallos

- Comprobaciones
  - Revisiones de diseño
  - Verificación de programas
  - Inspección de código
- Pruebas (tests) – Son necesarias, pero tienen problemas:
  - » no pueden ser nunca exhaustivas
  - » sólo sirven para mostrar que hay errores, no que no los hay
  - » a menudo es imposible reproducir las condiciones reales
  - » los errores de especificación no se detectan

4.3. Tolerancia a Fallos

Limitaciones de la prevención de fallos

- u Los componentes de hardware fallan, a pesar de las técnicas de prevención – La prevención es insuficiente si » la frecuencia o la duración de las reparaciones es inaceptable » no se puede detener el sistema para efectuar operaciones de mantenimiento
- u Ejemplo: naves espaciales no tripuladas
- u La alternativa es utilizar técnicas de tolerancia de fallos

Grados de tolerancia de fallos

- u Tolerancia completa (fail operational) – El sistema sigue funcionando, al menos durante un tiempo, sin perder funcionalidad ni prestaciones
- u Degradación aceptable (fail soft, graceful degradation) – El sistema sigue funcionando con una pérdida parcial de funcionalidad o prestaciones hasta la reparación del fallo
- u Parada segura (fail safe) – El sistema se detiene en un estado que asegura la integridad del entorno hasta que se repare el fallo

El grado de tolerancia de fallos necesario depende de la aplicación

```
Tolerancia a Fallas
              |
              |--- Detección de Errores
              |                    |
              |                    |--- Detección Concurrente
              |                    |
              |                    |--- Preemptive Detection
              |
```

```
|--- Recovery
           |
           |--- Error Handling
           |              |
           |              |--- Rollback
           |              |--- Rollforward
           |              |--- Compensation
           |
           |--- Fault Handling
                            |
                            |--- Diagnosis
                            |--- Isolation
                            |--- Reconfiguration
                            |--- Reinitialization
```

Redundancia

- La tolerancia de fallos se basa en la redundancia
- Se utilizan componentes adicionales para detectar los fallos y recuperar el comportamiento correcto
- Esto aumenta la complejidad del sistema y puede introducir fallos adicionales
- Es mejor separar los componentes tolerantes del resto del sistema

Técnicas de tolerancia a fallos en SW versión simple:

- Partitioning.
- Detección de errores.
- Atrapar excepciones.
- Checkpoint y restart.
- Procesos pares.
- Diversidad de datos.

Técnicas de tolerancia a fallos en SW versión múltiple:

- Bloques de recuperación.
- Programación de N versiones.
- Programación N self-checking.
- Bloques de recuperación por consenso.
- $t/(n-1)$-variant programming.

Bloques de recuperación: ...
Redundancia en hardware

- Redundancia estática
  - – Los componentes redundantes están siempre activos
  - – Se utilizan para enmascarar los fallos
  - – Ejemplo: » Redundancia modular triple (ó N), TMR/NMR
- Redundancia dinámica
  - – Los componentes redundantes se activan cuando se detecta un fallo
  - – Se basa en la detección y posterior recuperación de los fallos
  - – Ejemplos: » sumas de comprobación » bits de paridad

5. Procesadores Tolerantes

5.1. Fallas en el Procesador

TBD

5.2. Redundancia Estática y Dinámica

Redundancia estática: programación con N versiones.
Redundancia dinámica: Consta de dos etapas, detección y recuperación.

- - Bloques de recuperación: Proporcionan recuperación hacia atrás.
- - Excepciones: Proporcionan recuperación hacia adelante.

Tolerancia de fallos de software

- Técnicas para detectar y corregir errores de diseño
- Redundancia estática – Programación con N versiones
- Redundancia dinámica
  - - Dos etapas: detección y recuperación de fallos
  - – Bloques de recuperación » Proporcionan recuperación hacia atrás
  - – Excepciones » Proporcionan recuperación hacia adelante

Programación con N versiones

- u Diversidad de diseño – N programas desarrollados independientemente con la misma especificación – sin interacciones entre los equipos de desarrollo
- u Ejecución concurrente
  - – proceso coordinador (driver) » intercambia datos con los procesos que ejecutan las versiones
  - – todos los programas tienen las mismas entradas
  - – las salidas se comparan
  - – si hay discrepancia se realiza una votación

5.3. Modelado de la Dependencia de Procesadores y su Confiabilidad

TBD

5.4. Acuerdo Bizantino

Fallas Bizantinas

La tolerancia a faltas bizantinas (BFT) es la resistencia de un sistema informático tolerante a faltas, en particular los sistemas informáticos distribuidos, a fallas de componentes electrónicos donde hay información imperfecta sobre si un componente falla. En una "falla bizantina", un componente, como un servidor, puede aparecer de manera incoherente, fallando y funcionando para sistemas de detección de fallas, presentando diferentes síntomas a diferentes observadores. Es difícil para los otros componentes declarar que falló y cerrarlo fuera de la red, ya que primero necesitan llegar a un consenso sobre qué componente falla. El término se deriva del problema de los generales bizantinos[1] donde los actores deben acordar una estrategia concertada para evitar una falla catastrófica del sistema, pero algunos de los actores no son confiables. También se ha hecho referencia a la tolerancia a faltas bizantinas con las frases consistencia interactiva o congruencia de fuente, avalancha de error, problema de acuerdo bizantino, problema de generales bizantinos y falla bizantina.

6. RAID

Resilient Disk Systems

Fundamentos Matemáticos

Enteros modulo n

Sea $\mathbb{Z} = \{\ldots -2, -1, 0, 1, 2, \ldots\}$ el conjunto de los números enteros, y sea $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ el conjunto de los números enteros no negativos. Sean $x \in \mathbb{Z}$ y $n \in \mathbb{Z}_+$, se define $m = x \mod n$ como $m \in \mathbb{Z}_+$ tal que $\exists y \in \mathbb{Z}$ que $x = ny + m$.

Ejemplo 21 $8 \mod 2 = 0$, $3 \mod 2 = 1$, *etc.*

Serie de Fourier Trogonométrica

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \operatorname{sen}(k\omega_0 t) + b_k \cos(k\omega_0 t), \tag{19}$$

en donde $\omega_0 = \frac{2\pi}{T}$,

$$a_0 = \frac{2}{T}\int_{-T/2}^{T/2} f(t)\,dt, \quad a_k = \frac{2}{T}\int_{-T/2}^{T/2} f(t)\cos(k\omega_0 t)\,dt \quad y \quad b_k = \frac{2}{T}\int_{-T/2}^{T/2} f(t)\operatorname{sen}(k\omega_0 t)\,dt.$$

Ejemplo 22 *Sea una señal cuadrada periódica*

$$f(t) : \begin{cases} 5 & 0 \leq t < T/2 \\ 0 & T/2 \leq t < T \end{cases}$$

que tiene una frecuencia $f = 1MHz$.
Dado que $T = \frac{1}{f} = \frac{1}{1\times 10^6}s = 1\times 10^{-6}s$. Calculamos ahora a_0 como:

$$a_0 = \frac{2}{T}\int_0^T f(t)\,dt = \frac{2}{T}\left[\int_0^{T/2} 5dt + \int_{T/2}^{T} 0dt\right] = \frac{2}{T}\int_0^{T/2} 5dt = \frac{2}{T}\left[5t|_0^{T/2}\right] =$$

$$= \frac{2(5)}{T}\left[\frac{T}{2} - 0\right] = 5.$$

Calculamos ahora a_k y b_k para cuatro términos.

$$a_1 = \frac{2}{T}\int_{-T/2}^{T/2} f(t)\cos(\omega_0 t)\,dt = \frac{2}{T}\left[\int_0^{T/2} f(t)\cos(\omega_0 t)\,dt + \int_{T/2}^{T} f(t)\cos(\omega_0 t)\,dt\right] =$$

$$= \frac{2}{T}\left[\int_0^{T/2} 5\cos(\omega_0 t)\,dt + \int_{T/2}^{T} 0\cos(\omega_0 t)\,dt\right] = \frac{2}{T}\int_0^{T/2} 5\cos(\omega_0 t)\,dt = \frac{10}{T}\int_0^{T/2}\cos(\omega_0 t)\,dt =$$

$$= \frac{10}{T\omega_0}\operatorname{sen}(\omega_0 t)|_0^{T/2} = \frac{10}{2\pi}\operatorname{sen}\left(\omega_0\frac{T}{2}\right) = \frac{5}{\pi}\operatorname{sen}\left(\frac{2\pi}{T}\frac{T}{2}\right) = \frac{5}{\pi}\operatorname{sen}(\pi) = 0.$$

$$b_1 = \frac{2}{T}\int_{-T/2}^{T/2} f(t)\operatorname{sen}(\omega_0 t)\,dt = \frac{2}{T}\left[\int_0^{T/2} f(t)\operatorname{sen}(\omega_0 t)\,dt + \int_{T/2}^{T} f(t)\operatorname{sen}(\omega_0 t)\,dt\right] =$$

$$= \frac{2}{T}\left[\int_0^{T/2} 5\operatorname{sen}(\omega_0 t)\,dt + \int_{T/2}^{T} 0\operatorname{sen}(\omega_0 t)\,dt\right] = \frac{2}{T}\int_0^{T/2} 5\operatorname{sen}(\omega_0 t)\,dt = \frac{10}{T}\int_0^{T/2}\operatorname{sen}(\omega_0 t)\,dt = ...$$

$$a_2 = \frac{2}{T}\int_{-T/2}^{T/2} f(t)\cos(2\omega_0 t)\,dt = \frac{2}{T}\left[\int_0^{T/2} f(t)\cos(2\omega_0 t)\,dt + \int_{T/2}^{T} f(t)\cos(2\omega_0 t)\,dt\right] = ...$$

$$b_2 = \frac{2}{T}\int_{-T/2}^{T/2} f\left(t\right)\operatorname{sen}\left(2\omega_0 t\right)dt = \frac{2}{T}\left[\int_{0}^{T/2} f\left(t\right)\operatorname{sen}\left(2\omega_0 t\right)dt + \int_{T/2}^{T} f\left(t\right)\operatorname{sen}\left(2\omega_0 t\right)dt\right] = ...$$

$$a_3 = \frac{2}{T}\int_{-T/2}^{T/2} f\left(t\right)\cos\left(3\omega_0 t\right)dt = \frac{2}{T}\left[\int_{0}^{T/2} f\left(t\right)\cos\left(3\omega_0 t\right)dt + \int_{T/2}^{T} f\left(t\right)\cos\left(3\omega_0 t\right)dt\right] = ...$$

$$f(t) = 2{,}5 + ...$$

Probabilidad

Definición de probabilidad. $P :\to [0,1]$.

Definición 11 *Sean A y B dos eventos con probabilidades asociadas $P\left(\right)$ $yP\left(\right)$ respectivamente. A y B se dicen independientes si:*

$$AB = \emptyset$$

Densidad de Probabilidad.

$$\int_{-\infty}^{\infty} f\left(x\right)dx = 1.$$

Teorema 1 *Teorema de Bayes.*

$$P\left(A|B\right) = \frac{P\left(B|A\right)P\left(A\right)}{P\left(B\right)}$$

Tareas

1. Por entregar – Averías Históricas en Computación.

Consideraciones generales para todas las tareas.

1. Incluir en la tarea el nombre de los autores del trabajo.
2. En caso de que dos o más trabajos sean iguales o casi iguales, estos serán anulados y tendrán calificación de 0.

Tarea 1: Averías Históricas en Computación

(**Nuevo.**)

Lean los siguientes dos artículos:

`https://listverse.com/2019/07/30/top-10-most-catastrophic-computer-failures-in-history/`
`https://uwaterloo.ca/news/news/researchers-discover-cause-catastrophic-computer-failures`

Realice lo siguiente.

- Definan un criterio para determinar cuando una avería es más crítica que otra.
- Ordenen las 10 averías de mayor a menor importancia según el criterio definido, expuestas en el primer artículo. Justifique su orden.
- Escojan una de las averías expuestas en el artículo.
- Profundicen la información de la avería seleccionada realizando la investigación correspondiente. Intuyan que falla pudo haber provocado la avería; justifiquen su afirmación.
- Plasmen todo lo anterior en un reporte, que además incluya un breve resumen del segundo artículo y se destaquen 3 cosas del mismo.

Para la entrega de la tarea considerar lo siguiente:

- Fecha límite de entrega: Martes 25 de marzo, 2025. 14:30 hrs.
- Tarea por equipos de 2 o 3 integrantes.
- El archivo que incluya la tarea debe estar en formato pdf.
- La tarea será entregada por correo.
- El archivo que se entregue debe tener el siguiente nombre: Tarea 1 - Apellido1, Apellido2, Apellido3.pdf, en donde ApellidoX son los primeros apellidos de los integrantes del equipo.

Nota: Los siguiente artículos pueden ser de interés para ahondar en el tema:

`https://www.cigniti.com/blog/37-software-failures-inadequate-software-testing/`
`https://www.worksoft.com/corporate-blog/top-software-failures`

Programas

1. Entregado – Señales continuas en el tiempo.
2. Por entregar – Señal + Ruido.

Consideraciones generales para todos los programas.

1. Los programas deben codificarse en lenguaje PYTHON. Deben poder ejecutarse en la v3.4.
2. En caso de que dos o más trabajos sean iguales o casi iguales, estos serán anulados y tendrán calificación de 0.

Programa 1: Señales continuas en el tiempo.

Sean $f(t) = A \operatorname{sen}(\omega_0 t + \phi)$, $g(t) = A \operatorname{sen}(2\omega_0 t)$ y $h(t) = A \operatorname{sen}(3\omega_0 t)$ tres funciones continuas en el tiempo. Realizar lo siguiente:

1. En una primera gráfica, grafique dos periodos de la señal $f(t)$ con $\phi = 0$ y dos periodos de la señal $f(t)$ con $\phi = \pi/4$.
2. En una segunda gráfica, grafique dos periodos de la señal $f(t)$ con $\phi = 0$, cuatro periodos de la señal $g(t)$ y seis periodos de la señal $h(t)$. Utilice colores diferentes para cada función. Es importante que se muestren los valores numéricos en cada eje.
3. Por separado grafique un periodo de la señal $f(t)$ con $\phi = 0$, junto con su valor eficaz; este último debe verse como una línea horizontal de diferente color a $f(t)$.
4. Por separado grafique dos periodos de la señal $|f(t)|$ (valor absoluto) con $\phi = \pi/2$.
5. Por separado grafique dos periodos de la señal $f^2(t)$ (valor cuadrático) con $\phi = \pi$.
6. Por separado grafique dos periodos de la señal $f^2(t)$ en decibeles con $\phi = 0$.

Consideraciones generales:

- La amplitud A debe ser propuesta y debe ser mayor a 1.
- La frecuencia f tambén debe ser propuesta para obtener la cantidad de ciclos que quieren graficarse. Recordar que $\omega_0 = 2\pi f$.
- Fecha límite de entrega – Jueves 13 de marzo, 2025. antes de las 13:30 hrs .
- El programa debe realizarse en equipos de 3 o 4 integrantes.
- El programa y el reporte deben quedar en un mismo archivo zip, el cual debe ser entregado por correo.
- El archivo zip que se entregue debe tener el siguiente nombre: Programa 1 - Apellido1, Apellido2, Apellido3, Apellido4.zip, en donde ApellidoX son los primeros apellidos de los integrantes del equipo. El mismo nombre debe utilizarse para el programa y el reporte (con la extensión correspondiente).

Programa 2: Señal + Ruido

(**Nuevo.**)

Sean $g(t) = A \operatorname{sen}(\omega_0 t)$, $r_1(t) = \frac{A}{6} \operatorname{sen}(8\omega_0 t)$ y $r_2(t) = \frac{A}{10} \operatorname{sen}(16\omega_0 t)$, tres funciones continuas en el tiempo. Realizar lo siguiente:

1. Grafiquen dos periodos de $g(t)$.
2. En una segunda gráfica, grafiquen 10 periodos de $r_1(t)$ y 20 periodos de la señal $r_2(t)$.
3. Por separado grafiquen $g(t) + r_1(t)$ usando el mismo intervalo de tiempo usado en la gráfica del item 1.
4. Por separado grafiquen $g(t) + r_2(t)$ usando el mismo intervalo de tiempo usado en la gráfica del item 1.
5. En otra gráfica, grafiquen $g(t) + r_1(t) + r_2(t)$ usando el mismo intervalo de tiempo usado en la gráfica del item 1.
6. En otra gráfica, grafiquen $g(t)\, r_2(t)$ usando el mismo intervalo de tiempo usado en la gráfica del item 1.
7. En otra gráfica más, grafiquen $(g(t) + r_1(t) + r_2(t))^2$ en decibeles usando el mismo intervalo de tiempo usado en la gráfica del item 1.

8. Finalmente, en otra gráfica, grafiquen $g^2(t)+{r_1}^2(t)+{r_2}^2(t)$ en decibeles usando el mismo intervalo de tiempo usado en la gráfica del item 1.

Consideraciones generales:

- La amplitud A debe ser propuesta y debe ser mayor a 1.
- La frecuencia f tambén debe ser propuesta para obtener la cantidad de ciclos que quieren graficarse. Recordar que $\omega_0 = 2\pi f$.
- Fecha límite de entrega – Miércoles 26 de marzo, 2025, 23:00hrs.
- El programa debe realizarse en equipos de 3 o 4 integrantes.
- El programa y el reporte deben quedar en un mismo archivo zip, el cual debe ser entregado por correo.
- El archivo zip que se entregue debe tener el siguiente nombre: Programa 2 - Apellido1, Apellido2, Apellido3, Apellido4.zip, en donde ApellidoX son los primeros apellidos de los integrantes del equipo. El mismo nombre debe utilizarse para el programa y el reporte (con la extensión correspondiente).

Referencias

[1] Algirdas Avizienis, " Design of fault-tolerant computers", *Fall Joint Computer Conference*, –, —, x-y month 1967, pp. 733 – 743.

[2] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", *IEEE Transactions on Dependable ......*

[3] IEEEStandardsBoard IEEE Standard Glossary of Software Engineering Terminology, 1990

[4] Israel Koren &C. Mani Krishna, " Fault-Tolerant Systems", Morgan Kaufmann, 2010

[5] Wilfredo Torres-Pomales, Software Fault Tolerance: A Tutorial. 2000.

[6] Roger S. Pressman, Software Engineering: A Practitioner's Approach, The McGraw-Hill Companies, Inc., 1997.

[7] Thomas Herault, Yves Robert, "Fault-Tolerance Techniques for High-Performance Computing", Springer, 2016

[8] Michael Butler, Cliff B. Jones, Alexander Romanovsky, "Methods, Models and Tools for Fault Tolerance", Springer Science & Business Media, 2009

[9] Basic Concepts and Taxonomy of Dependable and Secure Computing.

[10] bibliographic information