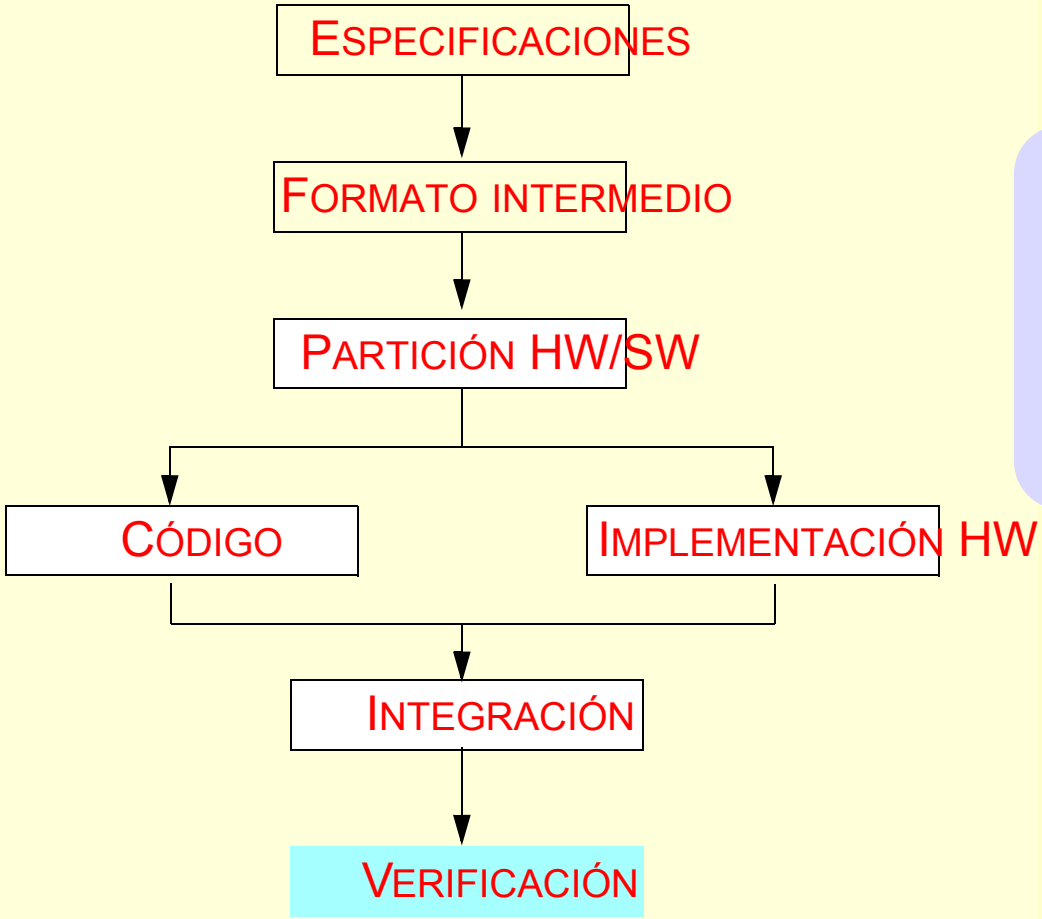
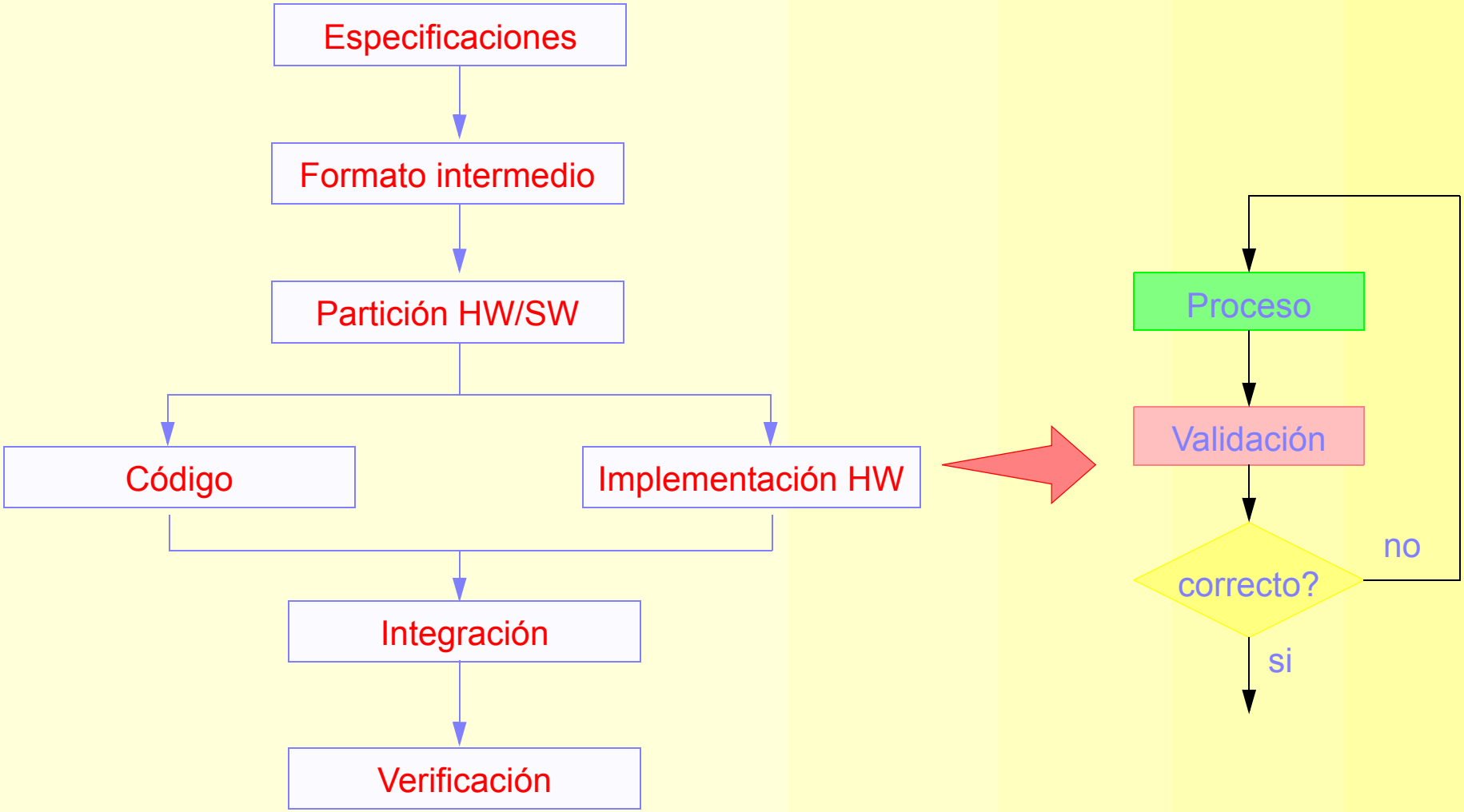
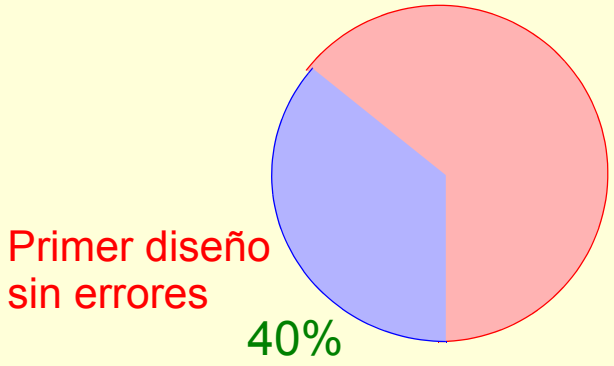
TEMA VII: VERIFICACIÓN



EL PROCESO DE VERIFICACIÓN ES EL PROCESO POR EL QUE SE GARANTIZA QUE EL SISTEMA DISEÑADO MUESTRA LAS ESPECIFICACIONES REQUERIDAS


Especificaciones
Formato intermedio
Partición HW/SW
Código
Implementación HW
Integración
Verificación
Proceso
Validación
correcto?
no
si

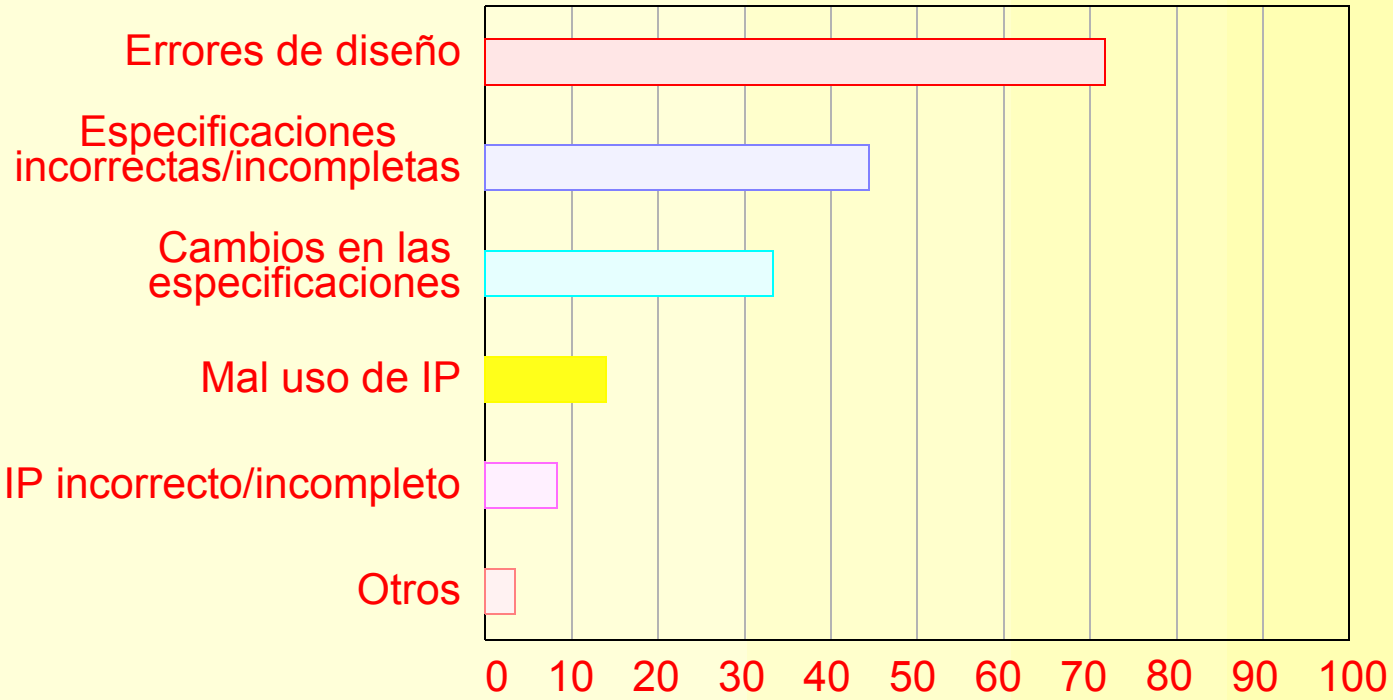
PROTOTIPOS DE DISEÑO



Primer diseño
con errores

PENTIUM

ARIANE 5



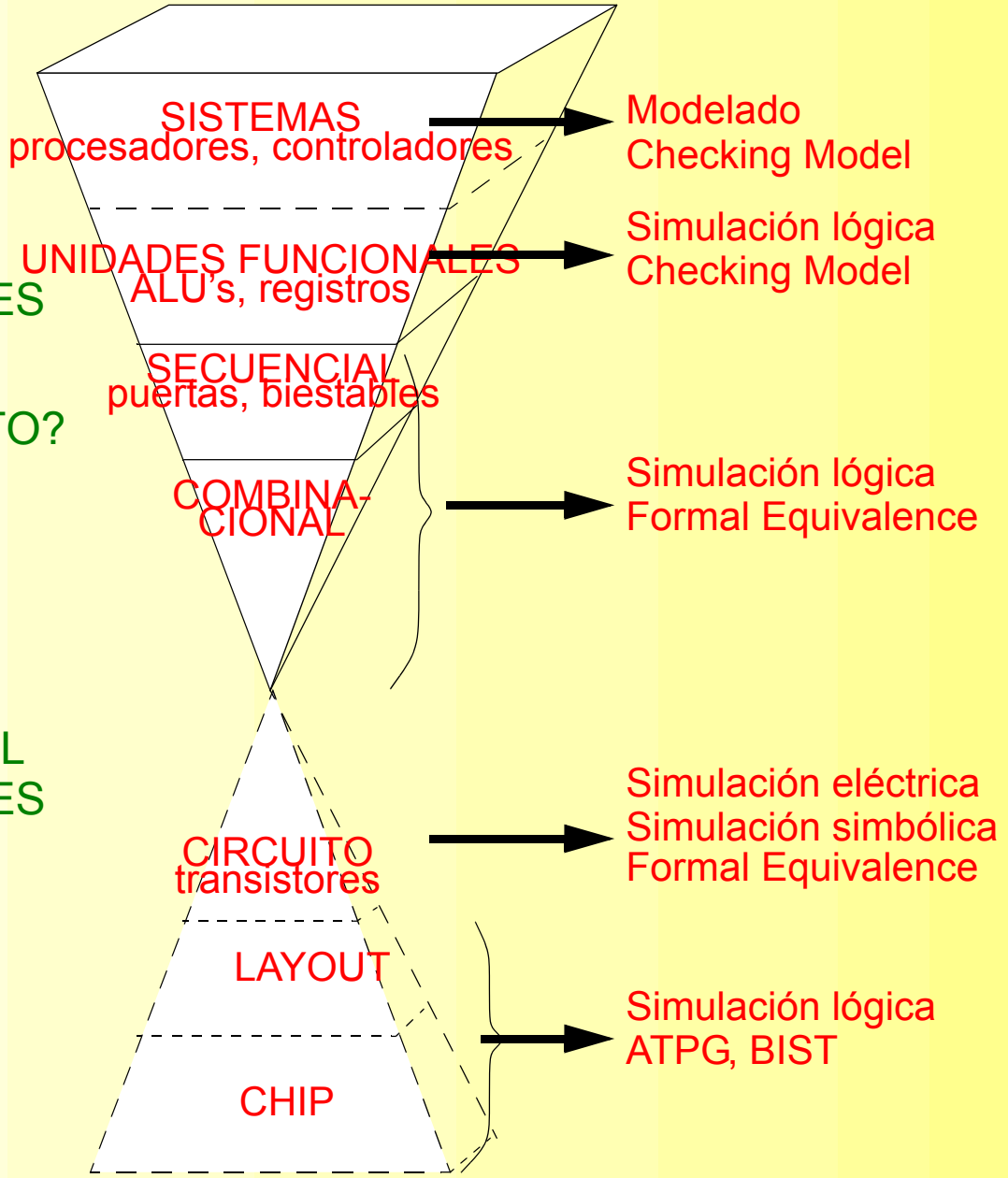
¿Validación o Verificación?

LA VALIDACIÓN ES EL PROCESO POR EL CUAL SE COMPRUEBAN LAS PROPIEDADES EXTRÍNSECAS DEL SISTEMA

¿ESTOY HACIENDO EL SISTEMA CORRECTO?

LA VERIFICACIÓN ES EL PROCESO POR EL CUAL SE COMPRUEBAN LAS PROPIEDADES INTRÍNSECAS DEL SISTEMA

¿ESTOY HACIENDO EL SISTEMA DE FORMA CORRECTA?

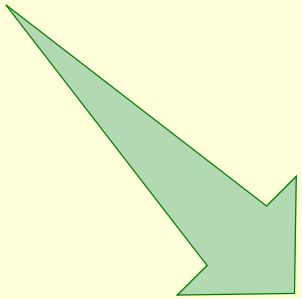


MÉTODOS DE VERIFICAICÓN

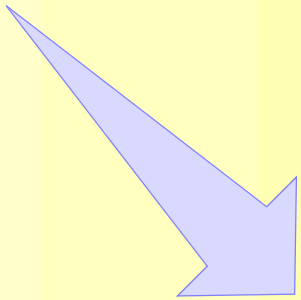
MÉTODOS FORMALES	Determinar analíticamente que las transformaciones realizadas sean correctas
MÉTODOS POR SIMULACIÓN	Determinar a través de simulación que las transformaciones realizadas sean correctas TESTBENCH
MÉTODOS POR EMULACIÓN	Determinar a través de emulación que las transformaciones realizadas sean correctas TESTBENCH

MÉTODOS FORMALES

RAZONAMIENTOS MATEMÁTICOS



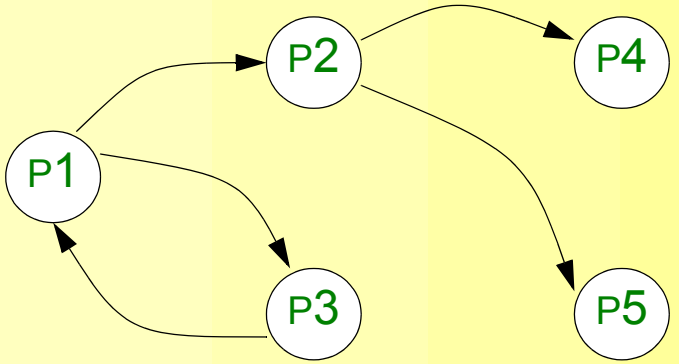
DESCRIPCIÓN FORMAL (LÓGICA TEMPORAL)



DIAGRAMAS DE ESTADO

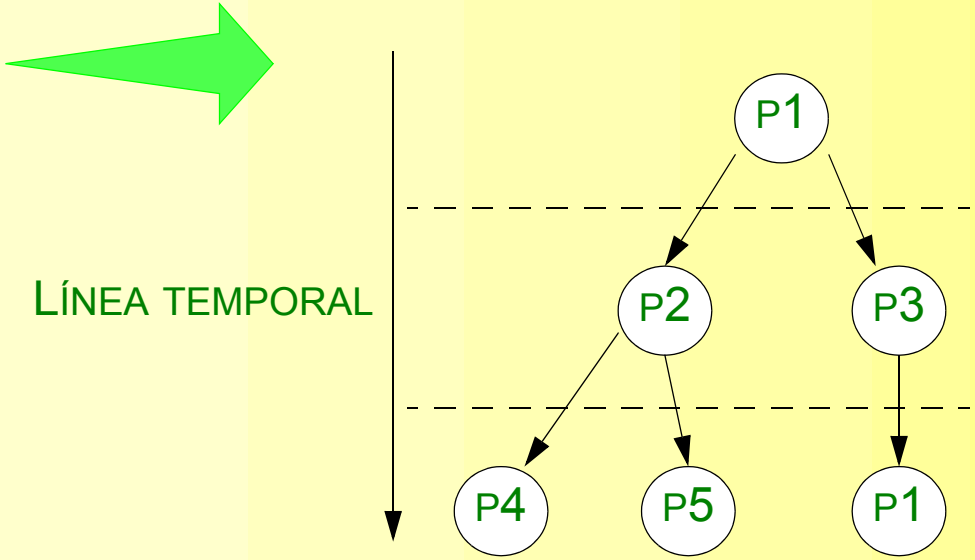
MÉTODOS FORMALES (LÓGICA TEMPORAL)

LA LÓGICA TEMPORAL PRETENDE VERIFICAR PROPIEDADES QUE CAMBIAN CON EL TIEMPO



LÓGICA DE ÁRBOLES DE COMPUTACIÓN (CTL)

Modelo discreto de tiempos con mas de un posible futuro desde un mismo estado inicial



MÉTODOS FORMALES (LÓGICA TEMPORAL)

M (DESCRIPCIÓN DEL SISTEMA)

- S (CONJUNTO DE ESTADOS)
- R (RELACIÓN DE PRÓXIMO ESTADO)
- L (CONJUNTO DE PROPOSICIONES CIERTAS EN UN ESTADO)

Cualquier proposición es una fórumla CTL

Si f1 y f2 son fórmulas CTL

- $\neg$f1
- f1 $\wedge$ f2
- f1 $\vee$ f2
- f1 => f2
- AX f1
- EX f1
- A [f1 $\cup$ f2]
- E [f1 $\cup$ f2]

Otros operadores útiles

- AF f = A [true $\cup$ f]
- EF f = E [true $\cup$ f]
- AG f = $\neg$EF$\neg$f
- EG f = $\neg$AF$\neg$f

MÉTODOS FORMALES (LÓGICA TEMPORAL)

SI CONSIDERAMOS UNA FÓRMULA CTL F, UNA ESTRUCTURA M=(S, R, L) Y UN ESTADO $S_0 \in S$, LA RELACIÓN |= DEFINE UNA VERDAD FORMAL Y ES DEFINIDA RECURSIVAMENTE DE LA SIGUIENTE FORMA

$M,s_0 \models f \Leftrightarrow f \in L(s_0)$

$M,s_0 \models \neg f \Leftrightarrow f \notin L(s_0)$

$M,s_0 \models f1 \wedge f2 \Leftrightarrow M,s_0 \models f1$ y $M,s_0 \models f2$

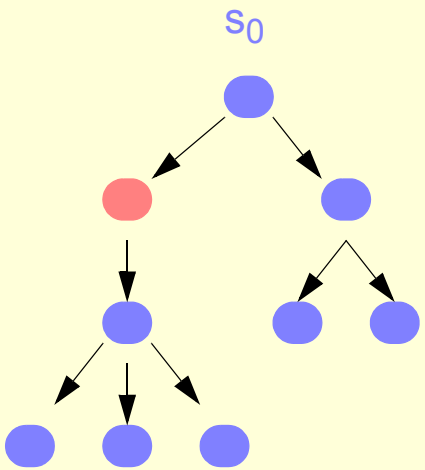
$M,s_0 \models AX\ f \Leftrightarrow$ para todo $t \in S \mid (s_0, t) \in R \Rightarrow M,t \models f$

$M,s_0 \models EX\ f \Leftrightarrow$ existe $t \in S \mid (s_0, t) \in R \Rightarrow M,t \models f$

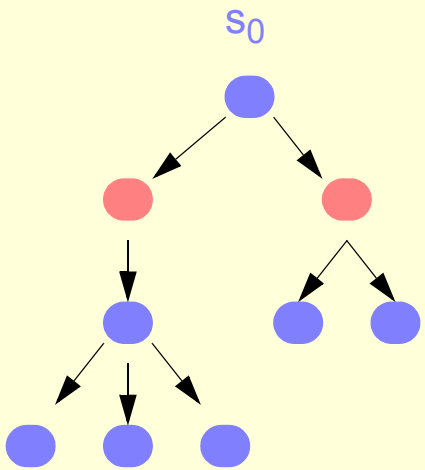
$M,s_0 \models A\ [f1 \cup f2] \Leftrightarrow$ existe $i \geq 0 \mid M,s_i \models f2 \wedge$ para todo $j = 0, ..., i-1\ M,s_j \models f1$ en todos los caminos

$M,s_0 \models E\ [f1 \cup f2] \Leftrightarrow$ existe $i \geq 0 \mid M,s_i \models f2 \wedge$ para todo $j = 0, ..., i-1\ M,s_j \models f1$ en algún camino

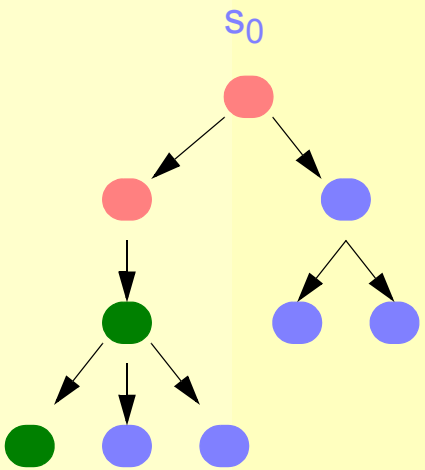
MÉTODOS FORMALES (LÓGICA TEMPORAL)



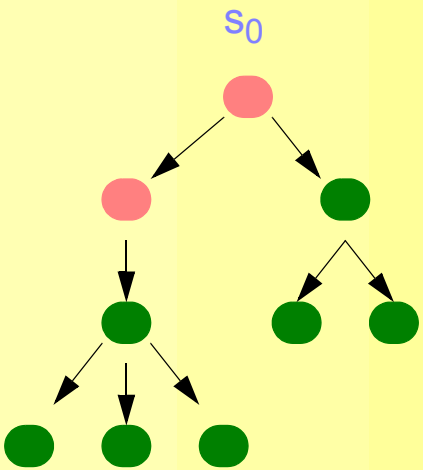
$M, s_0 \models EX\ f1$



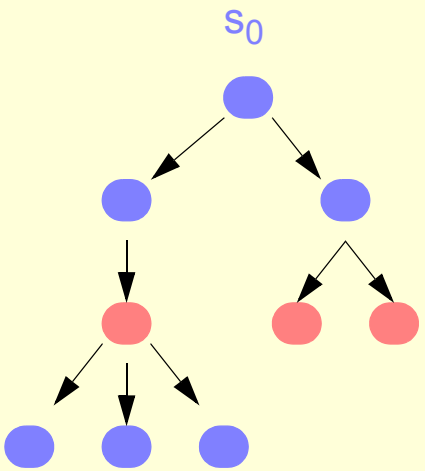
$M, s_0 \models AX\ f1$



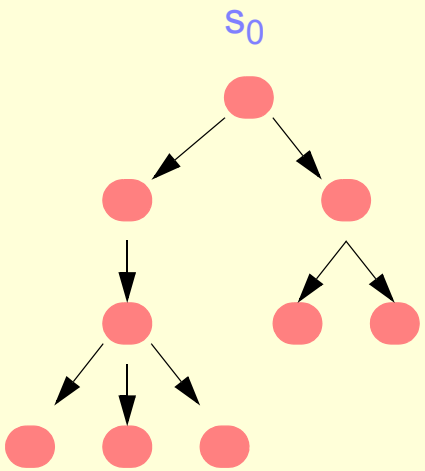
$M, s_0 \models E\ [f1 \cup f2]$



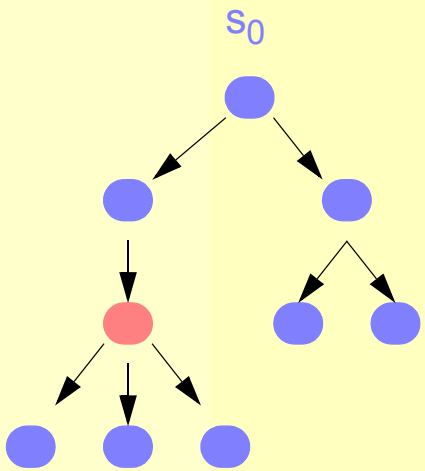
$M, s_0 \models A\ [f1 \cup f2]$



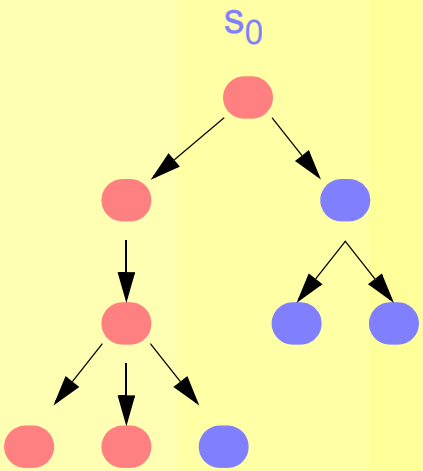
$M, s_0 \models AF\ f1$
$= A\ [true \cup f1]$



$M, s_0 \models AG\ f1$



$M, s_0 \models EF\ f1$
$= E\ [true \cup f1]$



$M, s_0 \models EG\ f1$

Satisface f1

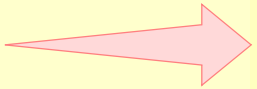
Satisface f2

CHEQUEO DE MODELO (CHECKING MODEL)

EL CHEQUEO DE MODELO CONSISTE EN DETERMINAR LAS PROPIEDADES QUE CUMPLEN UN SISTEMA DETERMINADO

CHEQUEO DE MODELO (CHECKING MODEL)

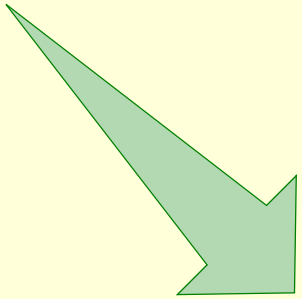
EL CHEQUEO DE MODELO CONSISTE EN DETERMINAR LAS PROPIEDADES QUE CUMPLEN UN SISTEMA DETERMINADO

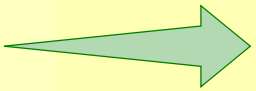
MODELADO Y ESPECIFICAICÓN DE SISTEMA  Diagrama de estados o modal

CHEQUEO DE MODELO (CHECKING MODEL)

EL CHEQUEO DE MODELO CONSISTE EN DETERMINAR LAS PROPIEDADES QUE CUMPLEN UN SISTEMA DETERMINADO

MODELADO Y ESPECIFICAICÓN DE SISTEMA

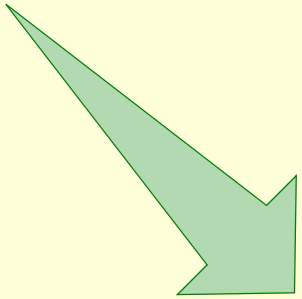


ESPECIFICACIÓN DE LAS PROPIEDADES  Fórmulas CTL

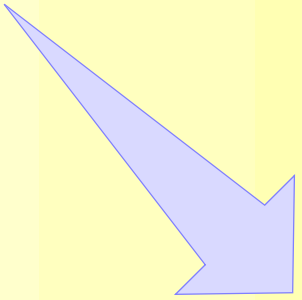
CHEQUEO DE MODELO (CHECKING MODEL)

EL CHEQUEO DE MODELO CONSISTE EN DETERMINAR LAS PROPIEDADES QUE CUMPLEN UN SISTEMA DETERMINADO

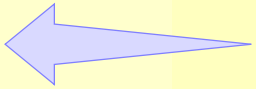
MODELADO Y ESPECIFICAICÓN DE SISTEMA



ESPECIFICACIÓN DE LAS PROPIEDADES



VERIFICACIÓN

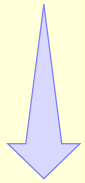


Dados una estructura M={S, R, L} y una fórmula lógica temporal f, determinar el conjunto de estados que satisfacen f ({s Œ S : M,s |= f})

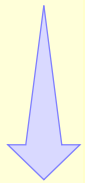
CHEQUEO DE MODELO (CHECKING MODEL)

PROCESO DE VERIFIACIÓN

DIVISIÓN DE LA FÓRMULA EN LAS DIFERENTES SUBFÓRMULAS DE JERARQUÍA INFERIOR

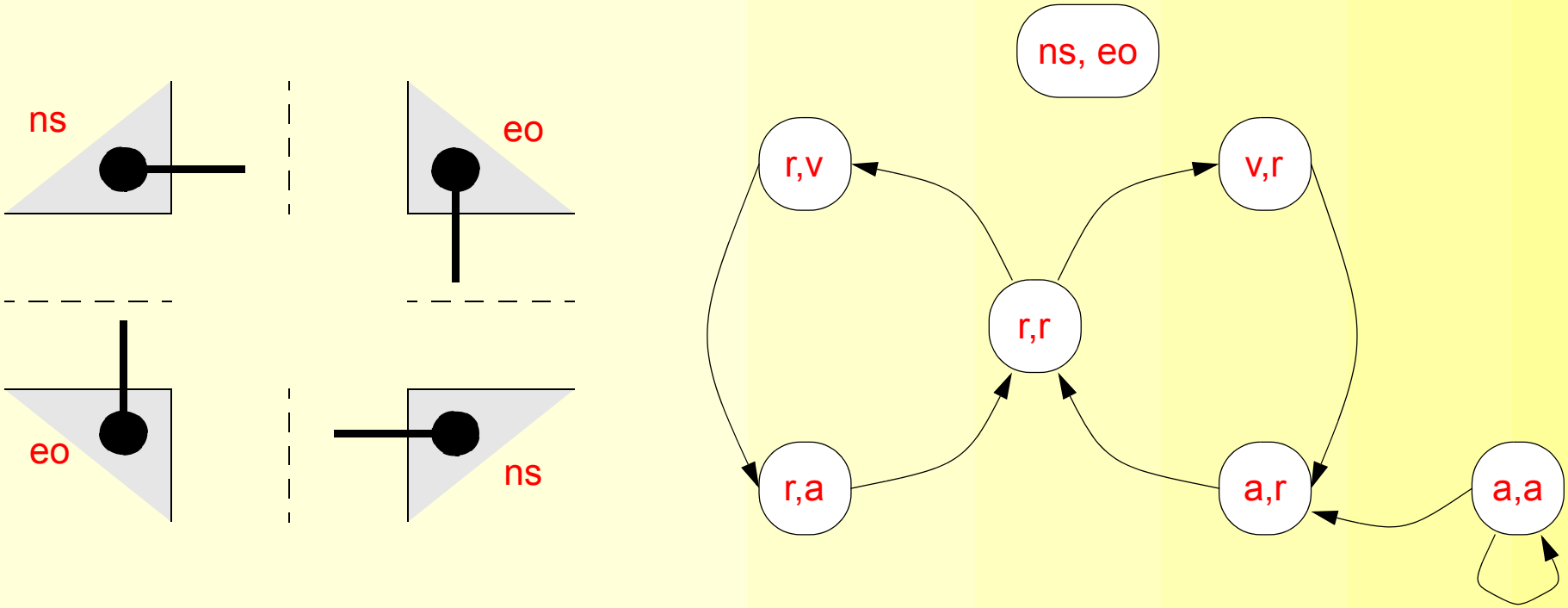


ETIQUETADO DE LOS ESTADOS QUE CUMPLEN LAS SUBFÓRMULAS ANTERIORES



IDENTIFICAR LOS ESTADOS CON LA ETIQUETA DE LA FÓRMULA ORIGINAL

CHEQUEO DE MODELO (CHECKING MODEL)



PROPIEDAD DE SAFETY

M |= ([ns = r] ^ [eo = r]) => AG ([ns = r] $\vee$ [eo = r])

CHEQUEO DE MODELO (CHECKING MODEL)

[ns = r] = p1

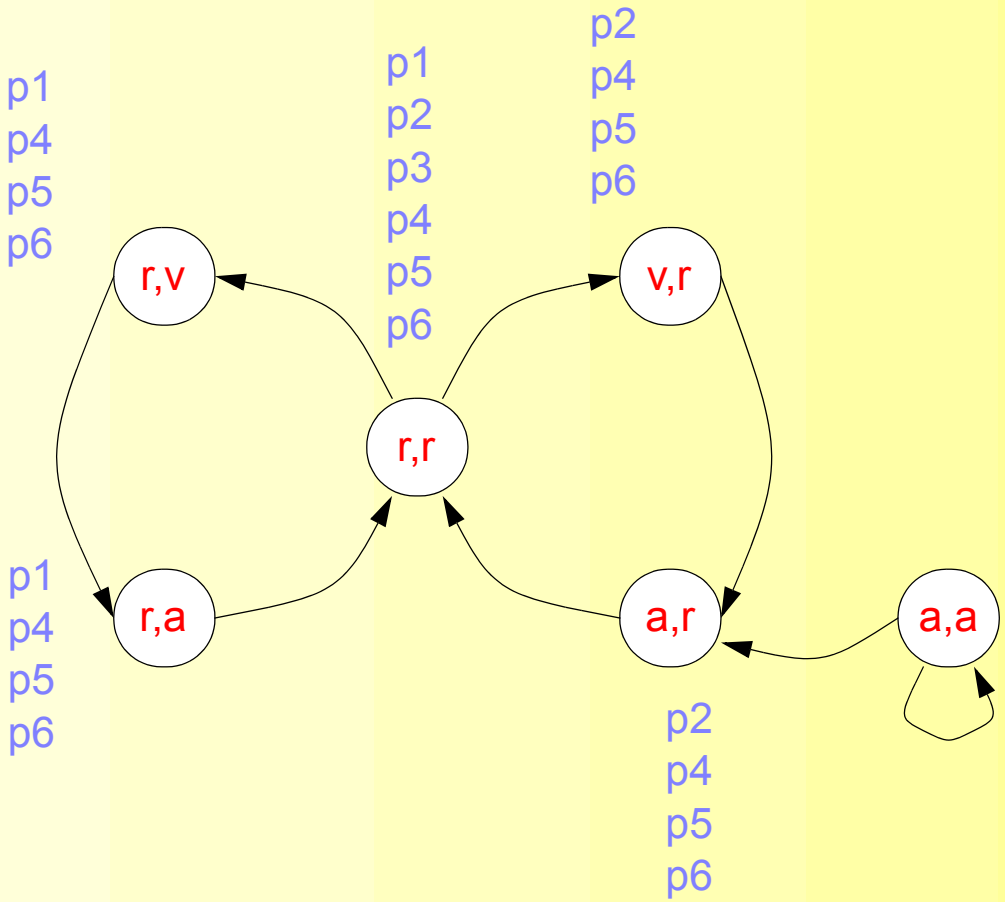
[eo = r] = p2

p1 ^ p2 = p3

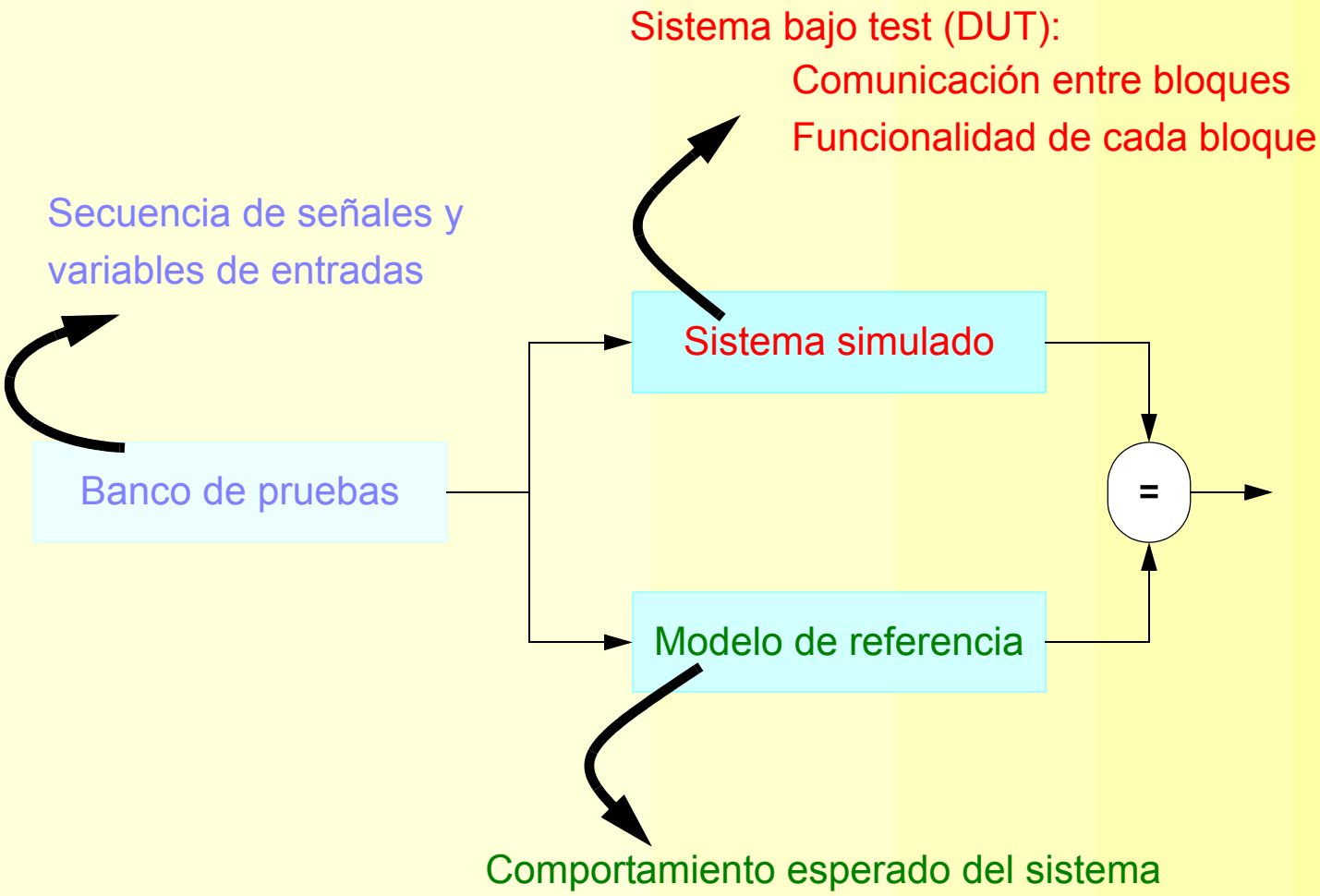
p1 ∨ p2 = p4

AG (p4) = p5

p3 => p5 = p6



MÉTODOS BASADOS EN LA SIMULACIÓN Y EMULACIÓN



MÉTODOS BASADOS EN LA SIMULACIÓN Y EMULACIÓN

Banco de pruebas

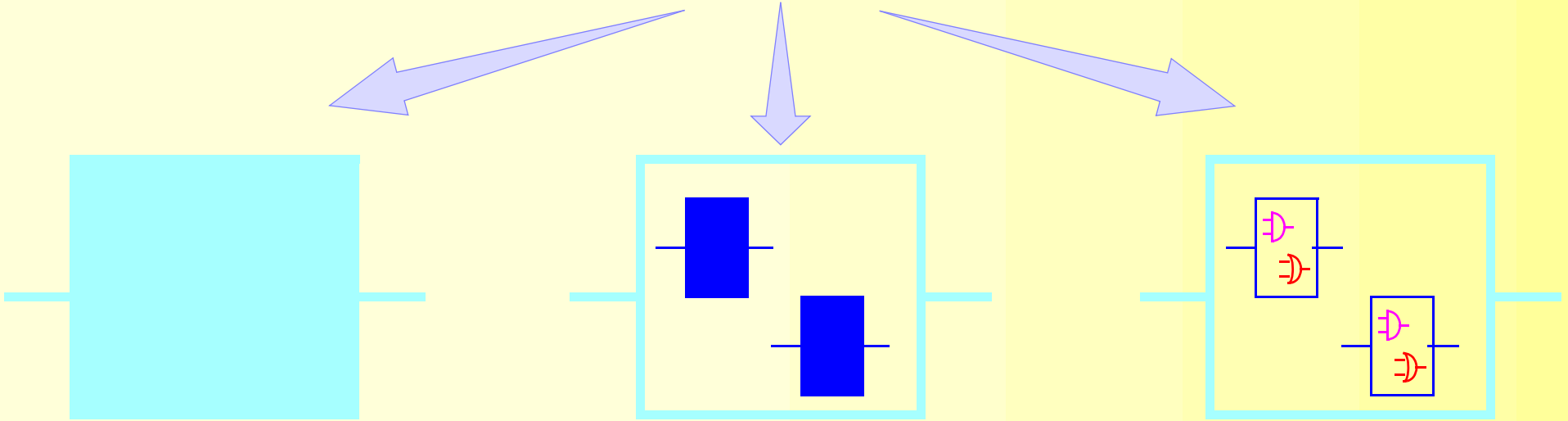
Transacciones: buses y circuitos de comunicaciones

Instrucciones: típica de las arquitecturas ISA

Errores atípicos: secuencias cuya aparición no es típica

Comportamiento funcional: verificación limitada

DUT



CAJA NEGRA

CAJA GRIS

CAJA BLANCA

TOLERANCIA A FALLOS

UN SISTEMA TOLERANTE A FALLOS ES AQUEL QUE MUESTRA UN COMPORTAMIENTO DE FALLO BIEN DEFINIDO CUANDO SUS COMPONENTES FALLAN, O ENMASCARAN LOS FALLOS DE LOS COMPONENTES AL USUARIO

Un servicio especifica una colección de operaciones cuya ejecución puede ser disparada por los usuarios del servicio o por el paso del tiempo

Un servidor es el componente que implementa un servicio sin exponer a los usuarios la representación del estado interno del servicio y los detalles de su implementación

Un servidor diseñado para proveer un determinado servicio es correcto si, en respuesta a sus entradas, se comporta de manera consistente con las especificaciones del servicio

Tolerancia a fallos

Un fallo de omisión es aquel en el que el servidor omite responder a una entrada

Un fallo de temporización se produce cuando la respuesta del servidor es funcionalmente correcta pero se produce fuera de tiempo

- Fallo de temporización temprano
- Fallo de temporización tarde

Un fallo de respueta se produce cuando el servidor responde de manera incorrecta, ya sea por valor o por transición de estado

- Fallo de valor
- Fallo de transición de estado

Un fallo de crash se produce cuando el servidor muestra un fallo de omisión a todas las entradas, desde el primer fallo de omisión hasta que se reinicie

- Fallo de amnesia
- Fallo de amnesia parcial
- Fallo de pausa
- Fallo de halting

TOLERANCIA A FALLOS

Propagación jerárquica de fallos

LA PROPAGACIÓN DE UN FALLO A TRAVÉS DE LA JERARQUÍA PUEDE PROVOCAR EL CAMBIO DEL TIPO DE FALLO