

LA PLAUSIBILIDAD DE LOS DISEÑOS DE SISTEMAS COMPUTACIONALES

Ulises Agüero A.*

RESUMEN

En este artículo se han estudiado los aspectos más sobresalientes de una versión modificada de la teoría de plausibilidad. La modificación se debe al deseo de este autor de lograr que la teoría sea aplicable a una vasta gama de problemas.

INTRODUCCION

Un *diseño* puede verse como la especificación de un todo conformado por elementos interrelacionados e inventado con el propósito de satisfacer ciertos objetivos (Simon, 1984). Este todo exhibe ciertas propiedades que, en general, no pueden explicarse en términos de las propiedades individuales de los elementos que lo conforman (Checkland, 1984). Además un diseño puede ser descrito en diferentes *niveles de abstracción*, dependiendo de la exhaustividad con que se especifiquen sus elementos y de las interrelaciones correspondientes¹.

Dado el panorama anterior, puede decirse que la *plausibilidad* (o aceptabilidad) de un diseño, descrito en cierto nivel de abstracción, depende de sus propiedades (deseables e indeseables) como un todo, y de las propiedades de sus elementos. El establecimiento de la plausibilidad de un diseño de un sistema computacional (de hardware, software o ambos) es significativo porque puede servir como base para lo siguiente:

- Aumentar (o disminuir) la confianza de un diseñador en su planteamiento del sistema

* Profesor del Departamento de Ingeniería en Computación, Instituto Tecnológico de Costa Rica

- Impeler el proceso de diseño
- Documentar el razonamiento utilizado durante el proceso de diseño
- Evaluar el diseño
- Comparar el diseño con otros productos creados con el propósito de satisfacer los mismos objetivos.

Una teoría puede entenderse como un conjunto de principios usados para explicar un fenómeno. En este sentido, una teoría para explicar la plausibilidad de los diseños de sistemas computacionales fue propuesta recientemente (Agüero 1987, Agüero y Dasgupta, 1987, sin publicar). Esta teoría consta de principios que caracterizan la naturaleza de los diseños plausibles, definen formas en las cuales se pueden desarrollar enunciados de plausibilidad para hacer afirmaciones sobre los méritos y deficiencias de un diseño y sirven como pautas de razonamiento durante el desarrollo de diseños. En este artículo se describirán los aspectos más importantes de una versión modificada de la teoría original. Se espera que esta nueva versión sea más fácil de aplicar a problemas de naturaleza variada, ya que la teoría original fue creada para explicar la plausibilidad de los diseños de arquitecturas de computadoras. La descripción de la nueva versión será general y a menudo informal.

ESTADOS DE PLAUSIBILIDAD

La plausibilidad de un diseño es determinada al establecer la plausibilidad de un número dado de afirmaciones sobre el diseño. En general, estas

afirmaciones describen características relevantes de diseño. En vista de que muchas de las afirmaciones son interdependientes, es común que la plausibilidad de una afirmación dependa de la plausibilidad de otras afirmaciones.

Definiendo la plausibilidad de una afirmación A en términos de la veracidad de A, se describen a continuación cuatro **estados de plausibilidad** por los que A puede transitar²:

- (s1) **Indeterminado**: es desconocido si existe evidencia en contra de la veracidad de A.
- (s2) **Factible**: no existe evidencia en contra de la veracidad de A.
- (s3) **Satisfactorio**: existe evidencia significativa en favor de la veracidad de A y no existe evidencia alguna en contra de ella.
- (s3) **Insatisfactorio**: no existe evidencia en contra de la veracidad de A.

Para simplificar la presentación, se dirá que *una afirmación es indeterminada (factible) (satisfactoria) (insatisfactoria) cuando el estado de plausibilidad de la afirmación sea indeterminado (factible) (satisfactorio) (insatisfactorio)*.

La naturaleza de la evidencia a la que se refieren las definiciones anteriores depende del método empleado para determinar la veracidad de la afirmación. En este artículo, se supondrá que la evidencia es una combinación de los tres tipos siguientes:

- **Precisa y formal**, basada en teorías de deducción bivalentes
- **Aproximada** (y posiblemente formal), basada en métodos heurísticos fundamentados en lógicas menos precisas que las lógicas bivalentes
- **Experimental**, basada en métodos experimentales tales como simulación y emulación.

La variedad de tipos de evidencia incluidos en la teoría de plausibilidad obedece a la tesis de este autor de que no es natural suponer que la veracidad de todas las afirmaciones debe determinarse usando la misma precisión. Si no puede obtenerse evidencia precisa debido a la naturaleza de una afirmación (por ejemplo, si se refiere a rendimiento o a factores humanos) o porque la complejidad computacional es

muy alta, entonces uno se ve forzado a usar evidencia aproximada o experimental. Al respecto, los principios de racionalidad limitada ("*bounded rationality*") (Simon, 1985) e incompatibilidad ("*incompatibility*") (Zadeh, 1975) establecen que una vez que un problema ha excedido cierto nivel de complejidad, un razonamiento (relativamente) preciso que produce soluciones óptimas debe sacrificarse en favor de un razonamiento aproximado que produce soluciones satisfactorias (pero no necesariamente óptimas). Este fenómeno es particularmente frecuente durante el proceso de diseño, donde percepciones y juicios humanos juegan un papel preponderante.

En el resto de esta sección se caracterizarán con más precisión los estados de plausibilidad y se estudiarán sus propiedades lógicas.

El espacio de plausibilidad

Las definiciones (s1)–(s4) de los estados de plausibilidad están relacionados entre sí de tal manera que generan el espacio de plausibilidad mostrado en la Figura No. 1. Basándose en esta figura, pueden establecerse las siguientes leyes de plausibilidad, dada una afirmación A:

- (L1) Si A califica para estar en el estado satisfactorio, entonces A califica para estar en el estado factible.
- (L2) Si A califica para estar en el estado factible, entonces A no califica para estar en el estado insatisfactorio.
- (L3) Si A no califica para estar en el estado insatisfactorio ni en el estado indeterminado, entonces A califica para estar en el estado factible.
- (L4) Si A no califica para estar en el estado insatisfactorio, entonces A puede o no calificar para estar en el estado satisfactorio.
- (L5) Si A no califica para estar en el estado satisfactorio, entonces A puede o no calificar para estar en el estado insatisfactorio.
- (L6) Si A califica para estar en el estado factible, entonces A puede o no calificar para estar en el estado satisfactorio.
- (L7) Si A califica para estar en el estado indeterminado, entonces A no califica para estar en ningún otro estado.

Las leyes (L1-L7) son intuitivamente deseables y pueden ser usadas en cualquier método de diseño para determinar el estado de plausibilidad de las afirmaciones.

(L9) Con respecto a (L8), suponga que E_j contuviera evidencia en contra de la veracidad de A. Esto causaría que A fuera inmediatamente insatisfactoria en vez de satisfactoria.

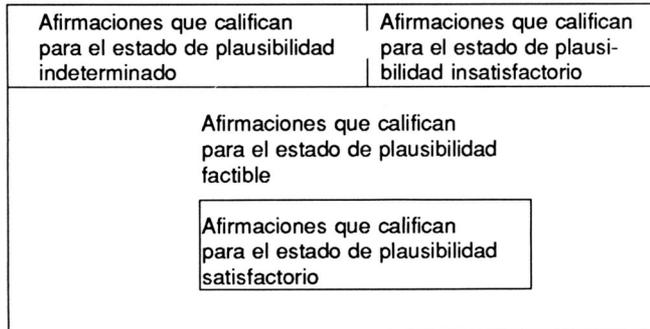


FIGURA No. 1. El espacio de plausibilidad

La naturaleza cambiante de la evidencia

En la mayoría de los problemas de diseño no elementales, muchas decisiones sobre la forma de un diseño son reconsideradas. Esto implica la posibilidad de que el estado de plausibilidad de una afirmación varíe debido a cambios en la evidencia utilizada durante el establecimiento de la veracidad de la afirmación. Aún más, la aparición de nueva evidencia, producto del refinamiento del diseño, puede también ocasionar cambios en los estados de plausibilidad de ciertas afirmaciones, conservando sin alteraciones la evidencia previa.

En el paso k de un método de diseño, el estado de plausibilidad de una afirmación se establece con base en la evidencia E_k de que se dispone. Ahora, supóngase que la expresión $j \geq k$, especifica que el paso de diseño j es el mismo que el paso k, o que j ocurre después de k. Con esta explicación en mente es posible describir las siguientes dos leyes de plausibilidad:

(L8) Suponga que la afirmación A es factible en el paso K. Por definición, esto significa que E_k no contiene evidencia en contra de la veracidad de A. Un diseñador desearía que, eventualmente, A fuera satisfactorio. Esto ocurriría si, en un paso $j \geq k$, E_j no contuviera evidencia en contra de la veracidad de A (que también era cierto con respecto a E_k) y si E_j contuviera evidencia significativa en favor de la veracidad de A (lo cual no era necesariamente cierto con respecto a E_k).

Razonamiento no monotónico en la teoría de plausibilidad

Un modo de razonamiento es llamado monotónico si no permite que conclusión alguna sea retractada en presencia de nueva evidencia. Sin embargo, la mayoría de los tipos de razonamiento creativo son no monotónicos en el sentido de que las conclusiones, derivadas en presencia de cierta evidencia, pueden ser retractadas posteriormente cuando aparezca nueva evidencia (Turner, 1984).

La ley (L8) representa una instancia de comportamiento monotónico, pues el concluir que A califica para estar en el estado satisfactorio no contradice el hecho de que A califique para estar en el estado factible de acuerdo con la ley (L1). Además, la ley (L9) representa una instancia de comportamiento no monotónico, ya que al calificar A para estar en el estado insatisfactorio se contradice el hecho de que A califique para estar en el estado factible (ver la Figura No. 1).

A manera de síntesis, puede decirse entonces que el modo de razonamiento asociado con la teoría de plausibilidad es inherentemente no monotónico, puesto que existe la posibilidad de que algunas conclusiones sean retractadas durante el proceso de diseño.

ENUNCIADOS DE PLAUSIBILIDAD

El establecimiento del estado de plausibilidad de una afirmación debe hacerse de una manera estructurada y sistemática. Para esto se cuenta con una estructura llamada enunciado de plausibilidad, definida como un quíntuple $E = \langle A, P, R, V, S \rangle$, donde:

- A representa la descripción de una afirmación sobre algún aspecto relevante del diseño.
- P representa el estado de plausibilidad en que se desea colocar a A.

- R representa la afirmación que se verificará para determinar si A califica para estar en el estado P. Más específicamente, se asume que R representa una afirmación tal que si R califica para estar en el estado P, A también califica para estar en el estado P.
- V representa la especificación del método que se utilizará para verificar E. Se dice que E ha sido verificado (con buen suceso) cuando, usando V, se logra establecer que R efectivamente califica para estar en el estado P; si R no califica entonces se dice que la verificación de E ha fracasado. Si E se verifica, A se coloca automáticamente en el estado P; si la verificación de D fracasa, el efecto que esto tiene en el estado de A no es definido por la teoría.
- S representa un conjunto de elementos de soporte, relevantes para el proceso de verificación (por ejemplo, otros enunciados y la descripción de programas o arquitecturas).

Ejemplos de enunciados de plausibilidad

Para empezar, considere el diseño de una red de interconexión. Una posible característica deseable de la red es que sea confiable. Obviamente, la existencia o no de esta característica en la red contribuye a la plausibilidad de la misma. Por lo tanto, un enunciado pertinente de plausibilidad sería el siguiente:

(E1)

A: La red de interconexión I es confiable
 P: Satisfactorio
 R: El diagnóstico de fallas y la tolerancia de fallas están eficazmente incorporados en I
 V: Simulación
 S: Descripción de I

El diseñador de (E1) ha decidido que diagnóstico y tolerancia de fallas implica confiabilidad; de ahí la forma de R y A. Posiblemente, otro diseñador habría decidido agregar otras características determinantes de la confiabilidad de I. Por lo tanto, es imprescindible que R sea cuidadosamente especificado por el diseñador.

(E1) establece que su verificación debería determinar si R califica para estar en el estado de plausibi-

lidad satisfactorio, pues se supone que esto implicaría que A también califica para estar en el estado satisfactorio. Nótese que se indica que el método de verificación debe ser de tipo experimental o, más específicamente, simulación. S indica que la descripción de I se utilizará durante la simulación. Una vez que (E1) haya sido verificado (con buen suceso), A se colocará en el estado satisfactorio.

Como segundo ejemplo, considere el siguiente enunciado, donde A y R son iguales:

(E2)

A: La complejidad de tiempo del programa de ordenamiento X es $O(n \log n)$ la complejidad de espacio es $O(n^2)$
 P: Satisfactorio
 R: La complejidad de tiempo de X es $O(n \log n)$ la complejidad de espacio es $O(n^2)$
 V: Análisis de complejidad
 S: Descripción de X, (E3) y (E4)

En este caso, la existencia de A o de R es claramente redundante pero, como lo ilustra (E1), no siempre ocurre esto. El campo S indica que la verificación de (E2) está relacionado con la verificación de (E3) y (E4), los cuales tienen la siguiente forma:

(E3)

A: La complejidad de tiempo de X es $O(n \log n)$
 P: Satisfactorio
 R: X es de la forma $X1; X2$, donde $O(n \log n)$ y $O(\log n)$ son la complejidad de tiempo de $X1$ y $X2$, respectivamente
 V: Análisis de complejidad
 S: Descripción de $X1$ y $X2$

(E4)

A: La complejidad de área de X es $O(n^2)$
 P: Satisfactorio
 R: Las estructuras de datos $Y1$ y $Y2$ son las que demandan más espacio en X y tienen complejidad de espacio $O(n \log n)$ y $O(n^2)$, respectivamente
 V: Análisis de complejidad
 S: Descripción de $Y1$ y $Y2$

Suponga ahora que (E3) y (E4) han sido verificados con buen suceso. De acuerdo con la definición del estado satisfactorio, esto implicaría que R de (E2) califica para estar en tal estado. Estas dependencias entre enunciados de plausibilidad serán estudiadas con más detalle en otra sección.

Lenguaje para la especificación de enunciados

Probablemente el lector ha notado que hasta ahora hemos empleado un lenguaje bastante informal para describir los componentes de los enunciados de plausibilidad, principalmente A y R. Este alto grado de informalismo no es siempre adecuado, especialmente cuando la verificación está parcial o totalmente automatizada, (refiérase a la Figura No. 2). Conforme aumenta el informalismo en los lenguajes del conjunto L_D también aumenta la dificultad de automatizar el proceso de verificación M (que usa los lenguajes de L_v). Además, el informalismo en la especificación de R y A dificulta el establecimiento y el uso de las dependencias entre enunciados de plausibilidad, mencionados en los ejemplos previos.

Lo expuesto en el párrafo anterior no debe interpretarse como una crítica a los lenguajes que modelan el razonamiento impreciso (Zadeh, 1978) pues, como ya se apuntó anteriormente, ciertos problemas requieren ser tratados con cierta imprecisión para conservar la relevancia en las soluciones propuestas. Lo que se apunta en ese párrafo son los problemas causados por la falta de rigidez y la excesiva ambigüedad de las estructuras sintácticas y semánticas de los lenguajes muy informales.

Lo que se desea enfatizar aquí es que la escogencia del lenguaje utilizado para describir los componentes de los enunciados de plausibilidad es de suma importancia. Esta escogencia debe efectuarse prestando atención a los requerimientos establecidos por el método de verificación utilizado. Por ejemplo, si se desea utilizar un probador de teoremas basado en una lógica de primer orden, un lenguaje similar al lenguaje de cálculo de predicados sería una escogencia apropiada (Mendelson, 1987).

Sin embargo, si el probador se basa en una lógica temporal (Moszkowski, 1986), o una lógica difusa ("fuzzy logic") (Zadeh, 1973), otros lenguajes serán más apropiados.

Independientemente del método de verificación empleado, un criterio importante en la escogencia de los lenguajes para la descripción de enunciados es el método de diseño empleado, especialmente si éste se basa en la teoría expuesta aquí. Cada método impone ciertas demandas sobre los niveles de formalismo y precisión requeridos, dependiendo de la naturaleza del diseño que se está desarrollando.

Particionamiento de afirmaciones

El particionamiento controlado de problemas en subproblemas y la subsecuente solución de los subproblemas para facilitar la solución del problema original, es una de las herramientas más frecuentemente utilizadas en el quehacer científico.

La teoría de plausibilidad incorpora este concepto de particionamiento con respecto a los enunciados de plausibilidad. Dado un enunciado $E = \langle A, P, R, V, S \rangle$, R puede particionarse en dos o más afirmaciones que a la vez generan nuevos enunciados asociados a E. El particionamiento de R debe hacerse de tal forma que, conociendo los estados de plausibilidad de las partes, se pueda inferir si R califica para estar en el estado P. Esto impone ciertas demandas en la estructura de R. Por ejemplo, suponga que X1 y X2 son dos afirmaciones y R es de la forma "X1 y X2", donde

X1 \equiv El programa Y es modular y

X2 \equiv Y es confiable

(E4)

A: La complejidad de área de X es $O(n^2)$

P: Satisfactorio

R: Las estructuras de datos Y1 y Y2 son las que demandan más espacio en X y tienen complejidad de espacio $O(n \log n)$ y $O(n^2)$, respectivamente

V: Análisis de complejidad

S: Descripción de Y1 y Y2

SISTEMA DE VERIFICACION

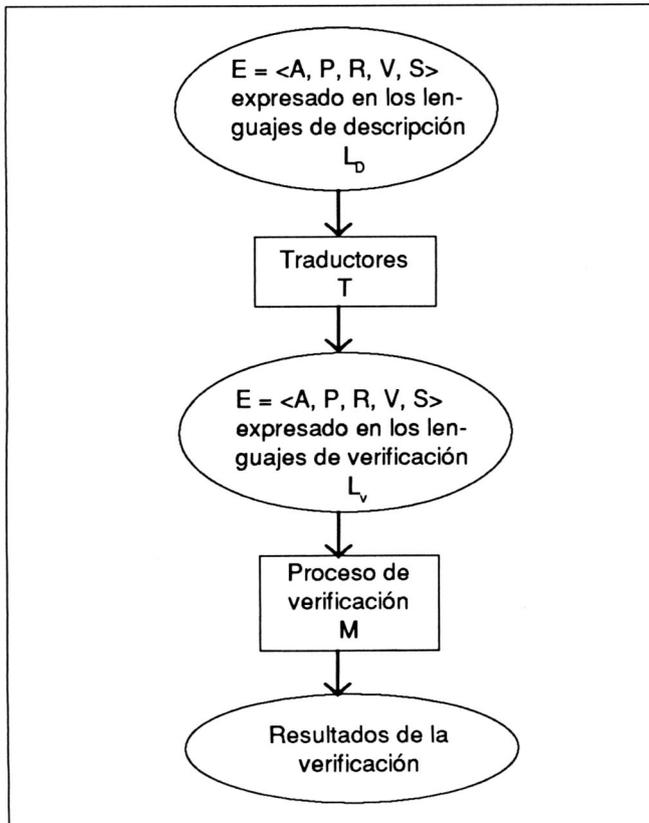


FIGURA No. 2. Organización de un sistema de verificación para enunciados de plausibilidad

Además suponga que se desea saber si R califica para estar en el estado insatisfactorio. Para esto debemos decidir si existe evidencia en contra de la veracidad de "X1 y X2". Nuestro problema es que debemos determinar la veracidad de "X1 y X2" a partir del conocimiento del estado de plausibilidad de X1 y X2, tomando en cuenta el significado de la conectiva "y". Intuitivamente, para que exista evidencia en contra de "X1 y X2" es suficiente que exista evidencia en contra de X1 o evidencia en contra de X2, lo cual equivale a decir que es suficiente que X1 o X2 califique para estar en el estado insatisfactorio.

Ahora, suponga que R es de la forma "X1 o X2" y, como antes, se desea saber si R califica para estar en el estado insatisfactorio. Basándose en el significado intuitivo de la conectiva "o", para que exista evidencia en contra de "X1 o X2" debe existir

evidencia en contra de X1 y evidencia en contra de X2, lo cual equivale a decir que X1 y X2 deben calificar para estar en el estado insatisfactorio.

Para esclarecer un poco más la situación se usará la siguiente notación, dada una afirmación X:

Notación	Significado
ind (X)	X califica para estar en el estado indeterminado
fac (X)	X califica para estar en el estado factible
sat (X)	X califica para estar en el estado satisfactorio
ins (X)	X califica para estar en el estado insatisfactorio

La discusión de los párrafos anteriores puede así resumirse en las leyes de plausibilidad descritas a continuación:

- (L10) ins (X1) y ins (X2) implica ins (X1 o X2)
- (L11) ins (X1) o ins (X2) implica ins (X1 y X2)

Las siguientes son otras leyes que podrían ser de utilidad en el proceso de particionamiento:

- (L12) sat (X1) y sat (X2) implica sat (X1 y X2) y fac (X1 y X2)
- (L13) sat (X1) o sat (X2) implica sat (X1 o X2) y fac (X1 o X2)
- (L14) fac (X1) y sat (X2) implica sat (X1 o X2) y fac (X1 o X2)
- (L15) ins (X1) y sat (X2) implica sat (X1 o X2) y ins (X1 y X2)
- (L16) fac (X1) y fac (X2) implica fac (X1 y X2) y fac (X1 o X2)

Muchas otras leyes similares podrían derivarse usando las definiciones de los estados y el significado intuitivo de las conectivas "y" y "o".

Leyes como las expuestas dependen en gran parte del lenguaje utilizado para describir la afirmación R de un enunciado $E = \langle A, P, R, V, S \rangle$. En particular, (en Agüero, 1987; Agüero y Dasgupta, 1987) se describen una serie de leyes para el caso cuando el lenguaje es el del cálculo de predicados (Mendelson, 1987).

Dependencias entre enunciados

Existen dos tipos principales de dependencias entre enunciados de plausibilidad: dependencias generadas por el proceso de particionamiento y dependencias generadas por afirmaciones sobre características (buenas o malas) de algunos elementos del diseño presentes en otras afirmaciones. Al primer tipo de dependencias se les llamará dependencias primarias y al segundo tipo dependencias secundarias.

Para ilustrar los dos tipos de dependencias, considere el problema de desarrollar un algoritmo para arquitecturas SIMD ("Single Instruction Stream Multiple Data Streams") que ordene n^2 elementos en tiempo menor que $O(n \log n)$, suponiendo que la organización de los procesadores es en forma de malla del tipo descrito por la Figura No. 3 (Hwang, 1984). Considere el siguiente enunciado de plausibilidad³:

(E5)

- A5: Existe un algoritmo de ordenamiento Y para mallas, con tiempo menor que $O(n \log n)$
- P5: Satisfactorio
- R5: Existe un algoritmo de ordenamiento Y para mallas con tiempo $O(n)$ y que requiere un espacio de $O(n^2 \log n)$
- V5: Análisis de complejidad
- S5: Descripción de Y

Para facilitar la verificación de (E5) se puede particionar R5 en dos afirmaciones para generar los siguientes enunciados:

(E6)

- A6: Existe un algoritmo de ordenamiento Y para mallas con tiempo $O(n)$
- P6: Satisfactorio
- R6: La porción Y1 de Y es la que consume más tiempo y requiere un tiempo de $O(n)$
- V6: Análisis de complejidad
- S6: Descripción de Y1 y del resto de Y

(E7)

- A7: Existe un algoritmo de ordenamiento Y para mallas, con espacio $O(n^2 \log n)$
- P7: Satisfactorio
- R7: Las estructuras X1 y X2 de Y son las que demandan más espacio y requieren $O(n^2 \log n)$
- V7: Análisis de complejidad
- S7: Descripción de X1, X2 y las otras estructuras de Y

Con la creación de (E6) y (E7) se introducen dependencias primarias entre (E5) y (E6), de tal manera que (E6) y (E7) se tornan elementos de soporte de (E5), lo cual se debe indicar en S5. La forma en que (E6) y (E7) contribuyen a la verificación de (E5) fue discutida en la sección anterior.

Una dependencia secundaria puede introducirse por medio de un enunciado que justifique el tiempo requerido por Y:

(E5D)

- A5D: Ningún algoritmo de ordenamiento para mallas puede ordenar n^2 elementos en un tiempo menor que $O(n)$
- P5D: Satisfactorio
- R5D: Es una malla de n^2 procesadores, la transposición de dos elementos cargados en esquinas opuestas de la malla requiere $4(n-1)$ transferencias entre procesadores
- V5D: Análisis de complejidad
- S5D: Descripción de la topología de una malla

(E5D) podría afectar la verificación de (E5) en el sentido de que si A5D fuera insatisfactoria, un diseñador podría tomar la decisión de modificar S con el propósito de obtener no sólo un tiempo menor que $O(n \log n)$, como lo especifica A5, sino también el tiempo óptimo. Si tal modificación ocurriera, la verificación de (E5) debería reconsiderarse. Si A5D fuera satisfactoria, la verificación (E5) no se afectaría, pues sólo se estaría confirmando una característica deseable de Y.

Algunas veces es prudente indicar, por medio de enunciados que generan dependencias secundarias, no sólo aspectos deseables, como en (E5D), sino también aspectos indeseables.

(E51)

- A51: Existen algoritmos de ordenamiento para mallas que requieren un tiempo menor que $O(n \log n)$, pero mayor que el de Y, y un espacio menor que el de Y.
- P51: Satisfactorio
- R51: El algoritmo Y1, que resulta de una modificación de Y, requiere un tiempo menor que $O(n \log n)$ pero mayor que el de Y y un espacio menor que el de Y.
- V51: Análisis de complejidad
- S51: Descripción de Y1 y Y

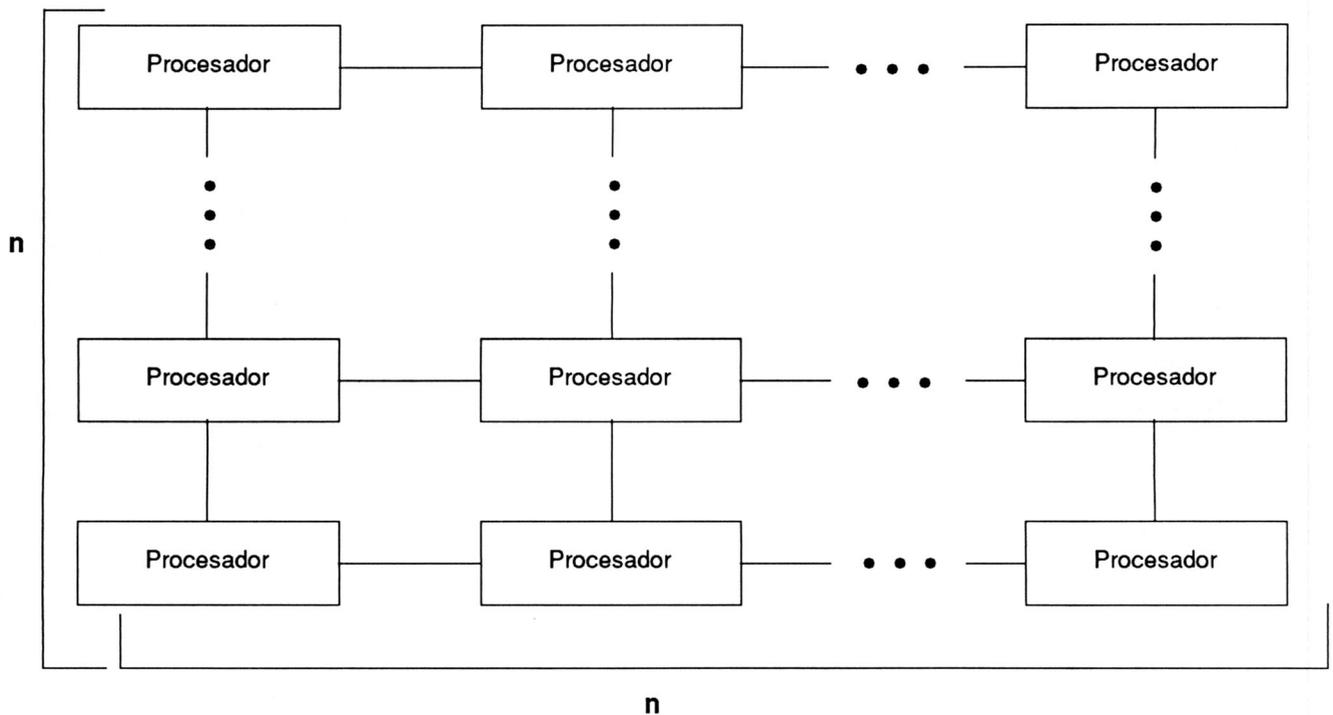


FIGURA No. 3. Una malla de n^2 procesadores

La característica indeseable de Y es que requiera un espacio mayor que otros algoritmos que también satisfacen la especificación descrita en A5, (esto es, que tengan un tiempo menor que $O(n \log n)$). Sin embargo, estos algoritmos posiblemente requieran un tiempo mayor que Y. En este sentido, la decisión de escoger Y se debió a que se consideró que A5D tiene más peso que A51⁴.

(E51) podría afectar la verificación de (E5) si A51 fuera satisfactoria y se decidiera escoger a Y1 en vez de Y, con base en el criterio de que una reducción en el espacio requerido es más deseable que el tiempo óptimo. El reemplazo de Y por Y1 implicaría una redefinición de (E5) y la subsecuente verificación.

Sin embargo, si A51 fuera insatisfactoria, la verificación de (E5) no sería afectada, pues negaría la existencia de una característica indeseable de Y.

El grafo de plausibilidad

Para modelar y procesar las dependencias primarias y secundarias entre enunciados, se introduce ahora el concepto del grafo de plausibilidad.

Un grafo de plausibilidad es un grafo con arcos direccionados y sin ciclos (DAG). Los nodos del grafo representan enunciados o las afirmaciones correspondientes. Los arcos representan dependencias primarias o secundarias. Dados dos nodos E1 y E2, un arco dirigido de E1 hacia E2 indica que E2 puede afectar la verificación de E1, de acuerdo con el tipo de dependencia (primarias y secundarias) y al menos dos tipos de dependencias secundarias (deseables e indeseables), se utiliza un tipo distinto de arco para modelar cada tipo de dependencia. De esta manera, las dependencias descritas en la sección anterior podrían modelarse con el grafo de la Figura No. 4. Si, con respecto a (E6) y (E7), R6 y R7 se particionaran en dos afirmaciones cada una, el grafo de la Figura No. 4 crecería según se muestra en la Figura No. 5. El particionamiento podría hacerse de la siguiente manera:

EL DESARROLLO DE DISEÑOS PLAUSIBLES

En las secciones previas se ha discutido brevemente la forma en que las consideraciones de plausibilidad pueden impedir el proceso de diseño.

- (E8)
A8: La porción Y1 de Y es la que consume más tiempo
•
•
•
- (E9)
A9: La porción Y1 de Y requiere un tiempo de $O(n)$
•
•
•
- (E10)
A10: Las estructuras X1 y X2 de Y son las que requieren más espacio.
•
•
•
- (E11)
A11: Las estructuras X1 y X2 de Y requieren un espacio de $O(n^2 \log n)$
•
•
•

Este asunto se ha tratado extensamente en otros escritos (Agüero, 1987, Agüero y Dasgupta, 1987, Hooton, 1987). En esta sección discutiremos no el proceso de diseño en sí, sino más bien las herramientas (total o parcialmente computarizadas) que deberían ponerse a disposición del diseñador.

Ambientes para diseños plausibles

Un diseño plausible se considera compuesto por tres elementos:

<especificación, garantía, historia>
donde:

- **Especificación** se refiere a los elementos que definen el diseño, independientemente de los enunciados asociados.
- **Garantía** se refiere al conjunto de enunciados cuyas afirmaciones están en estado satisfactorio o factible, y al grafo de plausibilidad correspondiente.
- **Historia** se refiere al registro de al menos los siguientes aspectos del diseño:

- la evolución del grafo de plausibilidad
- los enunciados cuyas afirmaciones han sido insatisfactorias, y las consecuencias que esto tuvo en el diseño.

Dada esta definición, un ambiente para el desarrollo de diseños plausibles tendría la forma general presentada en la Figura No. 6. Los sistemas para modelar diseños plausibles son los encargados de desarrollar cada uno de los elementos constituidos ya mencionados <especificación, garantía, historia>.

Los sistemas para la administración de la base de datos y conocimiento son componentes críticos de un ambiente que muy posiblemente se constituyan en el cuello de botella de cualquier ambiente que se implante. Tales sistemas de administración requieren una integración profunda de inteligencia artificial y tecnologías para bases de datos (Brodie, 1985).

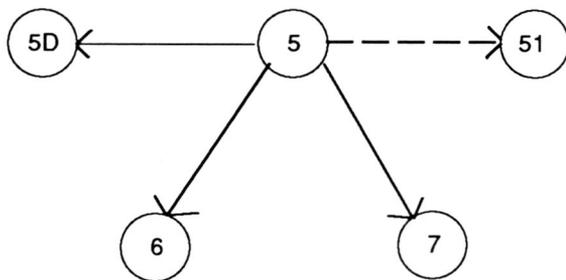


FIGURA No. 4. Un grafo de plausibilidad

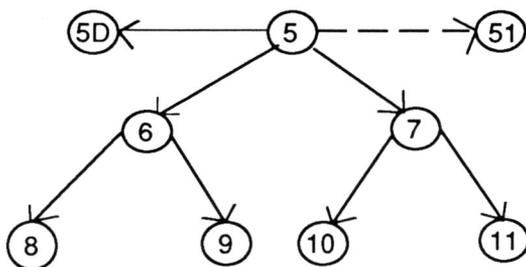


FIGURA No. 5. Un grafo de plausibilidad

Otro aspecto relevante al ambiente de la Figura No. 6 es la integración de las herramientas de diseño con los lenguajes que se empleen. Las ventajas de esta integración han sido reconocidas por un buen número de investigadores (Brown, 1983; Gilman, 1986) e incluyen las siguientes:

- Facilitar la comunicación de resultados de una etapa de diseño hacia la siguiente,
- Proveer compatibilidad entre diferentes niveles de abstracción y
- Cooperar con el diseñador en la administración de la complejidad inherente en la mayoría de las actividades de diseño.

Para finalizar, se describirán a continuación los lineamientos generales de un ambiente para el diseño de programas plausibles, el cual se está desarrollando en el Departamento de Computación y el Centro de Investigaciones en Computación del Instituto Tecnológico de Costa Rica.

Plausible-CAD: Un ambiente para el diseño de Programas Plausibles

Plausible-CAD está siendo desarrollado en el Instituto Tecnológico de Costa Rica por Juan Carlos Gómez, Franco Quirós, Vicente Gómez, Sergio Cascante, Rodolfo Soto, Jaime Casasa y el autor de este artículo. Paralelamente, Carlos Loría se encuentra investigando la posibilidad de desarrollar un sistema axiomático de plausibilidad.

Plausible-CAD⁵ es un ambiente para el apoyo computarizado del diseño de programas. La organización general del ambiente se muestra en la Figura No. 7 y los componentes se describen a continuación:

- **Núcleo de Plausible-CAD:** es común a todos los ambientes para diseños plausibles, independientemente de la aplicación. Por ejemplo, un ambiente para el diseño de arquitecturas usaría el mismo núcleo.
- **Configuración:** permite adaptar la interfaz del núcleo a una aplicación determinada, incluyendo el idioma en que se entablará el diálogo con el diseñador.

- **Asistencia:** sugiere al diseñador formas de proceder durante el diseño (usando la Base de Conocimiento y un método de diseño dado) y provee información sobre diseños plausibles que hayan sido catalogados en la Base de Estilos de Diseño.
- **Verificación:** se encarga de la verificación de los enunciados de plausibilidad usando el método escogido por el diseñador (o sugerido por el sistema).
- **Administración de diseños:** coordina las actividades de los diseñadores, cuando el trabajo se haga en grupo (Kezner, 1984). Además, registra la evolución de los tres componentes de un diseño plausible y atiende consultas al respecto. Finalmente, permite que se retroceda a un punto dado de la evolución y se rediseñe a partir de ahí.

AMBIENTE PARA DISEÑOS PLAUSIBLES

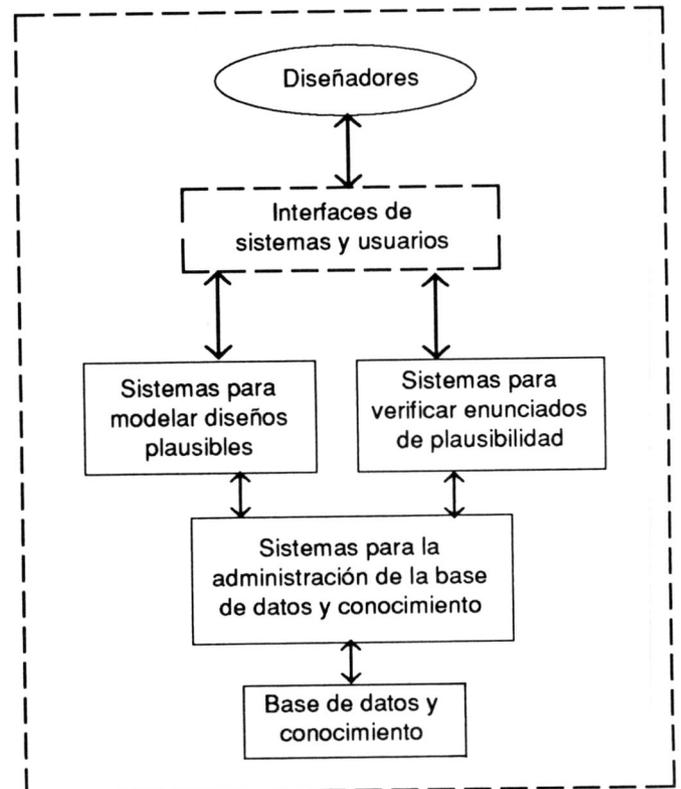


FIGURA No. 6. Componentes principales de un ambiente para diseños plausibles

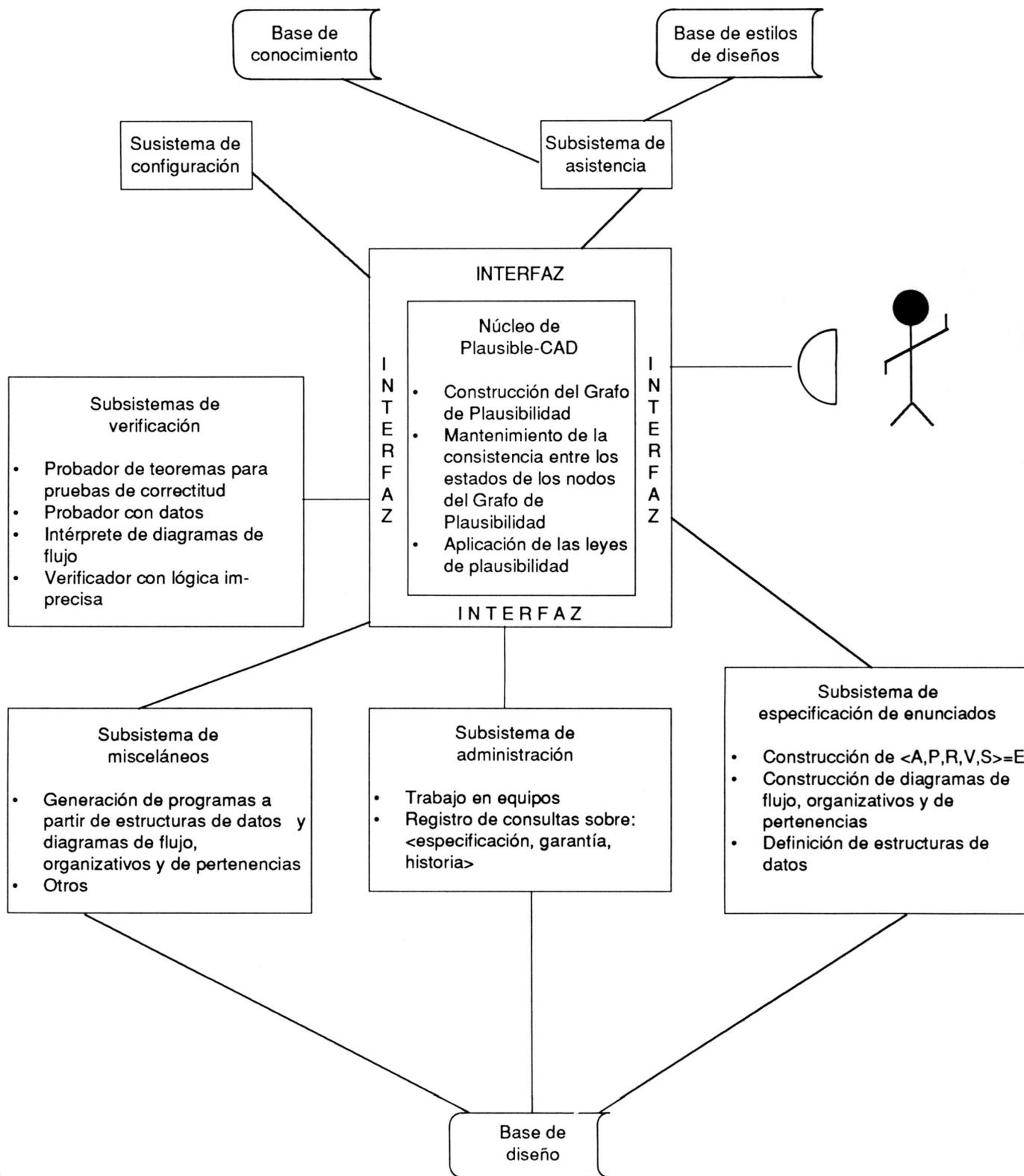


FIGURA No. 7. Organización general de Plausible-CAD

Especificación de enunciados: este sistema integra los lenguajes de especificación (o descripción) escogidos para la aplicación con la especificación de los componentes de los enunciados de plausibilidad. En vista de que Plausible-CAD está siendo desarrollado para el diseño de programas, se han escogido los diagramas de flujo estructurados descritos (en Agüero, 1985) para la especificación de los algoritmos. Estos diagramas son acompañados por una especificación abstracta de las estructuras de datos y por diagramas que definen la organización jerárquica de los diagramas de flujo vistos como módulos y las relaciones de alcance entre las variables de cada módulo (referirse a [Agüero, 1985] para más detalles)⁶.

Misceláneos: por el momento, sólo se ha pensado en asignar a este subsistema la labor de sintetizar programas en un lenguaje como Pascal, tomando como punto de partida los diagramas y las estructuras de datos correspondientes.

NOTAS

1. Nótese que el término abstracción se usa aquí en el sentido de eliminación selectiva de detalles, con el propósito de que se concentre la atención en sólo los aspectos relevantes.
2. Naturalmente, A sólo puede estar en un estado a la vez.
3. La malla a la que se refieren los enunciados de esta sección es como la descrita en la Figura N° 3.
4. Enunciados que indican propiedades indeseables fueron empleados por primera vez en (9), el cual describe un caso de diseño basado en la teoría de plausibilidad.
5. CAD: Computer Aided Design.
6. Vale mencionar que los diagramas y las estructuras de datos serán especificados interactivamente.

REFERENCIAS

- 1– Agüero, U. **A theory of plausibility for computer architecture designs**. Disertación de PhD. Center for Advanced Computer Studies, University of Southwestern, Louisiana, Abril, 1987.
- 2– Agüero, U. y Dasgupta, S. *A plausibility-driven approach to computer architecture design*. **Communications ACM**. 3(11): 922-932.
- 3– Agüero, U. y Dasgupta, S. *A plausibility-driven approach to program development*. **The Science of Computer Programming** (enviado para que se considere su publicación).
- 4– Agüero, U. **Programación con diagramas estructurados**. Cartago: Editorial Tecnológica de Costa Rica, 1985.
- 5– Brodie, M.L. y Mylopoulos, J. *Knowledge bases and databases: Semantic vs. computational theories of information*. En Ariav, G y Clifford, J. (Eds.) **New Directions for Database Systems**. Ablex Publishing Company, 1985.
- 6– Brown, H., Tong, C. y Foyster, G. *Palladio: An exploratory environment for circuit design*. **Computer**. 41-56, Diciembre, 1983.
- 7– Checkland, P.B. **Systems thinking, systems practice**. John Wiley & Sons, 1984.
- 8– Gilman, A.S. *VHL-The designer environment*. **IEEE Design and Test**. 3(2): 42-47, Abril, 1986.
- 9– Hooton, A.R. **The plausible design of CASE: a computer architecture simulation engine**. Tesis de MSc. Center for Advanced Computer Studies, University of Southwestern, Louisiana. Abril, 1987.
- 10– Hwang, K. y Briggs, F.A. **Computer architecture and parallel processing**. Sección 5.3.2. McGraw-Hill, 1984.
- 11– Kerzner, H. **Project management**. 2ed. Van Nostrand, 1984.
- 12– Mendelson, E. **Introduction to mathematical logic**. 2ed. Belmont: Wadsworth, 1987.
- 13– Moszkowski, B. *Executing temporal logic programs*. **Technical Report No. 71**. Computer Laboratory. University of Cambridge, 1986.
- 14– Simon, H.A. **The sciences of the artificial**. 2ed. MIT Press, 1984.
- 15– Turner, R. **Logics for artificial intelligence**. John Wiley & Sons, 1984.
- 16– Zadeh, L.A. *The concept of a linguistic variable and its applications to approximate reasoning-I*. **Information Sciences**. 8: 199-249, 1975.

17- Zadeh, L.A. *PRUF-a meaning representation language for natural languages. International Journal of Man-Machine Studies.* 10: 395-46, 1978.

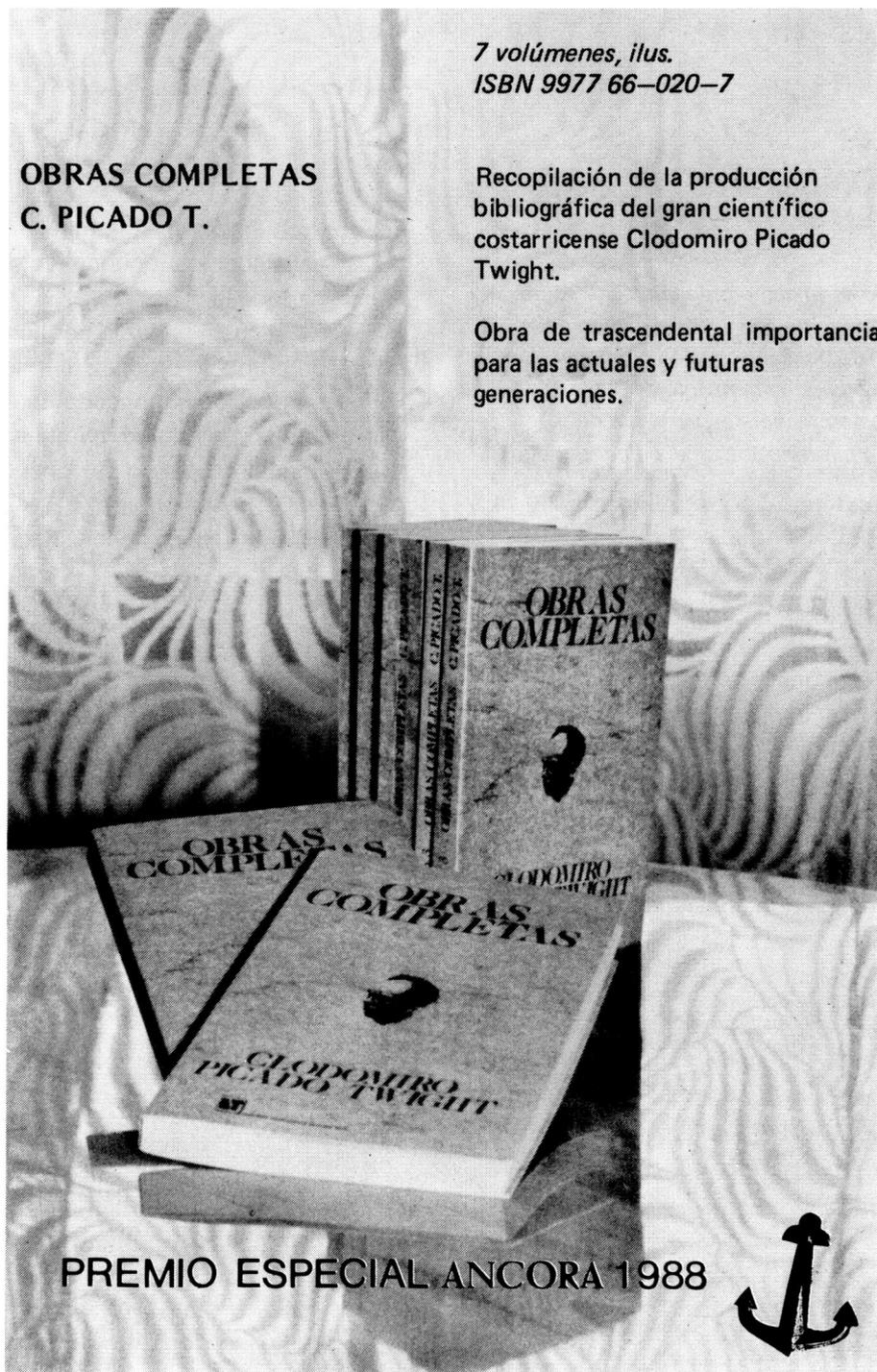
18- Zadeh, L.A. *Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems Man and Cybernetics.* SMC-3 (1): 28-44, enero, 1973.

**OBRAS COMPLETAS
C. PICADO T.**

*7 volúmenes,ilus.
ISBN 9977 66-020-7*

Recopilación de la producción bibliográfica del gran científico costarricense Clodomiro Picado Twight.

Obra de trascendental importancia para las actuales y futuras generaciones.



PREMIO ESPECIAL ANCORÁ 1988

