

CARACTERÍSTICAS DE UN SISTEMA TUTORIAL INTELIGENTE PARA LA RESOLUCIÓN DE PROBLEMAS VERBALES ARITMÉTICO-ALGEBRAICOS⁵⁵

FEATURES OF AN INTELLIGENT TUTORING SYSTEM FOR THE SOLVING OF ARITHMETIC-ALGEBRAIC WORD PROBLEMS

Arnau, D.¹, Arevalillo-Herráez, M.², Puig, L.¹

¹ Departament de Didàctica de la Matemàtica, Universitat de València

² Departament d'Informàtica, Universitat de València

Resumen. Hemos diseñado un sistema tutorial inteligente, llamado *Resolutor de Problemas Basado en Hipergrafos*, que es capaz de supervisar la resolución de problemas verbales aritmético-algebraicos tomando el método cartesiano como modelo de competencia. El programa a) admite resoluciones aritméticas y algebraicas; b) permite el recurso a una o más ecuaciones; y c) determina la validez de las expresiones que se asocian a las cantidades atendiendo a las restricciones del problema. La conjunción de estas características proporciona la flexibilidad necesaria para dar cuenta de múltiples posibles soluciones de un mismo problema. Además, el entorno también permite añadir nuevas colecciones de problemas, lo que posibilitaría su uso en la enseñanza y aprendizaje de la resolución de problemas verbales tanto en primaria como en secundaria.

Palabras clave: sistema tutorial inteligente; problemas verbales; resolución algebraica; resolución aritmética; enseñanza y aprendizaje asistidos por ordenador.

Abstract. We have designed an intelligent tutoring system, called *Hypergraph Based Problem Solver*, which is capable of supervising arithmetic-algebraic word problem solving by taking the cartesian method as a component of formal competence. This software a) supports both arithmetic and algebraic solving; b) allows the use of one or multiple equations; and c) determines the validity of the expressions associated with the quantities attending to the problem's restrictions. The combination of these features provides the flexibility to account for multiple possible solutions to one problem. In addition, the environment allows adding new problems to the repository. This makes it specially suitable as a teaching and learning tool in primary and secondary school.

Key words: intelligent tutoring system; word problems; algebraic solving; arithmetical solving; computer assisted learning and teaching.

⁵⁵ Esta investigación se ha realizado dentro del proyecto EDU2009-10599 concedido por la Dirección General de Investigación Científica y Gestión del Plan Nacional I+D+I del Ministerio de Educación y Ciencia.

ANTECEDENTES

El diseño de entornos informáticos para la enseñanza y aprendizaje de la resolución de problemas verbales ha sido un tema habitual de investigación en la Matemática Educativa. Algunos de estos programas pretenden sustituir el papel del profesor; otros ofrecen entornos en los que el resolutor puede recurrir a distintos sistemas de representación o puede liberarse de tareas rutinarias como el cálculo de operaciones aritméticas. Entre los primeros se incluirían los sistemas expertos y los sistemas tutoriales inteligentes (en adelante, STI) y podríamos citar como ejemplos *AnimalWatch* (Beal y Arroyo, 2002) o *MathCAL* (Chang, Sung y Lin, 2006). Entre los segundos, se encontrarían los programas de propósito general y los bancos de trabajo y podríamos citar como ejemplos *Word Problem Assistant* (Thompson, 1989), *HERON* (Reusser, 1993) y *ANIMATE* (Nathan, 1990) o el uso que se hace de las hojas electrónicas de cálculo en la resolución de problemas verbales (Arnau y Puig, 2005; Sutherland y Rojano, 1993). Los entornos que pretenden sustituir el papel del profesor habitualmente se centran en dos tareas: la tutorización del proceso de resolución y la generación de una secuencia de problemas que se adapte a las características de un estudiante.

Un ejemplo de STI que pretende tutorizar la resolución de un problema verbal sería *MathCAL* (Chang, Sung y Lin, 2006). El programa organiza el proceso de resolución aritmética alrededor de las cuatro fases de trabajo propuestas por Polya (entender el problema; concebir un plan; ejecutar el plan y revisar la solución) para lo que se hace pasar al usuario por cuatro interfaces distintas de manera secuencial. El resolutor genera el plan a partir de una secuencia de pasos prealmacenada que, durante la ejecución del plan, se representará mediante un esquema en forma de árbol (similar al empleado en otros entornos como *Word Problem Assistant* o *HERON*) que permitirá expresar la solución del problema. El programa es capaz de determinar la validez de la solución y de ofrecer una solución correcta en caso que el usuario lo solicite.

El programa *AnimalWatch* (Beal y Arroyo, 2002) sería un ejemplo de STI que dirige la atención a determinar el modelo de estudiante y a proponer una secuencia de problemas que se adapte a sus conocimientos matemáticos. El programa plantea problemas sobre animales en peligro de extinción (osos panda, ballenas, etc.) que deben resolverse de manera aritmética y está dirigido a estudiantes de últimos cursos de primaria y primeros de secundaria. A diferencia de *MathCAL*, *AnimalWatch* pone el énfasis en el resultado del problema, en lugar de hacerlo en el proceso de resolución.

OBJETIVOS Y MARCO TEÓRICO

Los entornos interactivos de aprendizaje para la resolución de problemas verbales que se han diseñado hasta el momento han debido decidir entre permitir flexibilidad al resolutor en sus decisiones y poder tutorizar sus acciones. Entre aquellos que tutorizaban realmente el proceso de resolución, la falta de flexibilidad se reflejaba en: un problema se asocia a una solución; una cantidad se asocia a una expresión; y cada problema se asocia a unos mensajes de error.

Hemos construido un sistema tutorial inteligente que es capaz de supervisar la resolución de aquellos problemas verbales aritmético-algebraicos que puedan reducirse a un conjunto de relaciones ternarias entre cantidades. Para dotarlo de flexibilidad en la respuesta hemos exigido que el programa cumpliera la siguientes condiciones:

Características de un sistema tutorial inteligente para la resolución de problemas verbales aritmético algebraicos

- *Independencia respecto al método de resolución.* El problema permite resolver problemas de manera aritmética y de manera algebraica.
- *Independencia respecto al uso de una o más ecuaciones.* Cuando se resuelve de manera algebraica, es posible el uso de una o más letras y el consiguiente recurso a una o más ecuaciones.
- *Independencia entre cantidad y su representación.* El programa no supone una asignación predeterminada entre una cantidad y su representación, sino que comprueba la validez de una expresión atendiendo a las restricciones del problema y a las decisiones del resolutor.
- *Independencia de funcionamiento del STI respecto del problema.* La verificación de la validez de las expresiones matemáticas que se introducen y los mensajes (básicos) de error y ayuda que proporcionará el STI son independientes del problema concreto que se está resolviendo.

Como el programa pretende tutorizar la actuación de sujetos reales cuando resuelven problemas, su diseño se fundamentará sobre un modelo de competencia en la resolución de problemas verbales que le servirá de referente. Hemos tomado como modelo de competencia la división en pasos ideales del método cartesiano (en adelante, MC) que describe las acciones de un sujeto ideal cuando resuelve un problema verbal de manera algebraica (Filloy, Puig y Rojano, 2008; Puig, 2003). De manera resumida, los pasos del MC, hasta el planteamiento de la ecuación, son: 1) lectura analítica del problema; 2) asignación de letra (o letras) a una (o varias) cantidad conocida; 3) representación del resto de cantidades desconocidas; 4) construcción de la ecuación (o de tantas ecuaciones como letras se empleen). Tomar el MC como modelo de competencia también nos permite tutorizar la resolución aritmética de problemas, ya que, desde un punto de vista algorítmico (es decir, dejando de lado los procesos cognitivos y centrándonos en las acciones que se realizan), podríamos decir que la resolución aritmética es un caso particular de la resolución mediante el MC en el que sólo se ejecutan los pasos primero y tercero.

EL FUNCIONAMIENTO DEL TUTOR

A continuación, ofreceremos ejemplos del funcionamiento del programa, cuando se resuelven los problemas *Conejos y gallinas* y *Camisas y pantalones*, que pondrán de manifiesto cómo se han integrado las distintas exigencias en un entorno informático.

Conejos y gallinas

En una granja, entre gallinas y conejos hay 20 cabezas y 52 patas. ¿Cuántos conejos y cuántas gallinas hay en la granja?

Camisas y pantalones

En las rebajas compré tres camisas y dos pantalones por 95 €. Recuerdo que el precio de un pantalón era de 25 €. ¿Cuál era el precio de cada camisa?

Podemos resumir los pasos segundo y tercero del MC como la definición de las cantidades que participarán en la resolución del problema. La definición de una cantidad conocida exigirá la asignación de un valor numérico; mientras que la definición de una cantidad desconocida supondrá la asignación de una letra o de una expresión que

materialice una relación que tiene esta cantidad con otras cantidades. Atendiendo a esto, una vez se selecciona un problema, el programa (ver Figura 2) muestra el enunciado y únicamente da al resolutor la posibilidad de pulsar el botón *Definir nueva cantidad*.

Una vez se acciona el botón, se activa el panel de declaración de cantidades, el usuario debe seleccionar el nombre de una cantidad de las que aparecen en un menú desplegable (ver Figura 3) y elegir entre tres posibles acciones: *Asignar número*; *Asignar letra* o *Asignar expresión*. Estas tres opciones responderían, respectivamente, a: dar un valor a una cantidad conocida, asignar una letra a una cantidad desconocida y asignar una expresión aritmética o algebraica a una cantidad desconocida. En caso que la asignación fuera incorrecta, el programa muestra un mensaje de error (ver Figura 3). Una vez se define una cantidad de manera correcta, se representa el nombre de la cantidad y la expresión que se le ha asociado en el cuadro *Cantidades definidas* (ver Figura 4).

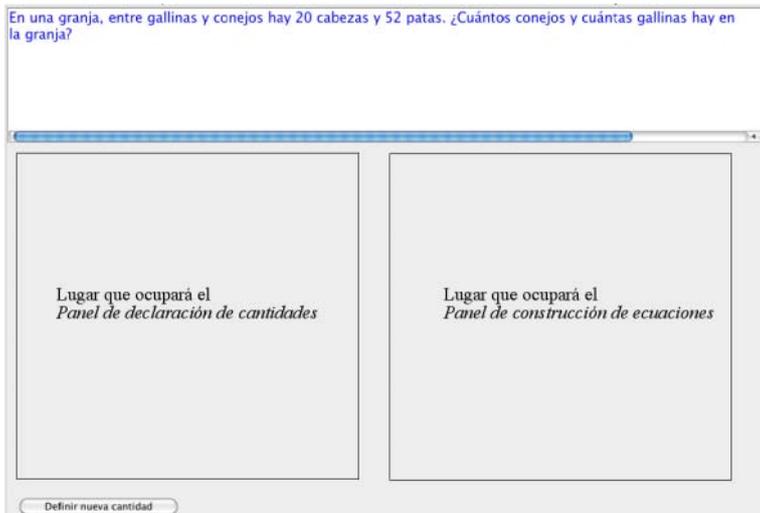


Figura 2

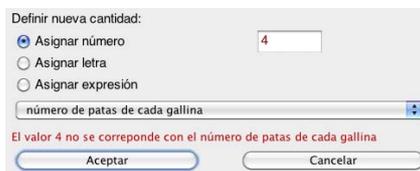


Figura 3

El usuario elige el orden en que declara las cantidades a las que se asigna un número o una letra y tiene libertad para decidir a qué cantidad desconocida asignará una letra y el número de letras que va a emplear. Así, por ejemplo, en la Figura 4, tras haber declarado que la cantidad conocida *número de patas de cada conejo* vale 4, se asigna la letra x a la cantidad desconocida *número de gallinas* cuando aún quedan cantidades conocidas por declarar.

Características de un sistema tutorial inteligente para la resolución de problemas verbales aritmético algebraicos



Figura 4

La definición de una cantidad a la que se le asigna una expresión aritmética o algebraica supone recurrir a una relación en la que aparece esta cantidad y exige que se hayan definido previamente las otras cantidades que participan en la relación. Así, si se declara la cantidad *número de conejos* basándose en el hecho de que la suma del número de conejos y de gallinas es igual al número de animales, deberán definirse previamente las cantidades *número de gallinas* y *número de cabezas*. Cuando se elige la opción *Asignar expresión*, el programa genera unos botones (ver Figura 6) que representan los signos de las cuatro operaciones básicas y la expresión de las distintas cantidades declaradas hasta el momento. En la Figura 5, se muestra la secuencia de botones que es necesario pulsar para asignar la expresión $20 - x$ a la cantidad *número de conejos*.



Figura 5

Para recordar a qué cantidad corresponde cada expresión se puede consultar la ventana en la que aparecen las cantidades que ya se han declarado. Además cuando se sitúa el cursor sobre uno de los botones que representa una cantidad, aparece un globo de ayuda con el nombre de la cantidad (ver Figura 6).

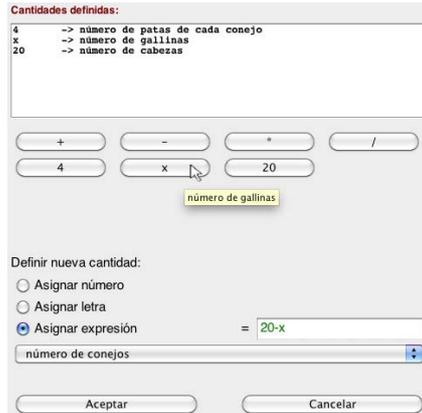


Figura 6

El programa gestiona la necesidad de utilizar paréntesis en el proceso de construcción de las expresiones para que, de esta forma, el usuario pueda centrarse en el proceso de resolución y no deba preocuparse de las dificultades propias del uso del lenguaje algebraico⁵⁶. Así, por ejemplo, mientras se construye la expresión para la cantidad *número de patas de todos los conejos*, el programa (ver Figura 7) no pone 4 entre paréntesis; pero sí, $20 - x$.

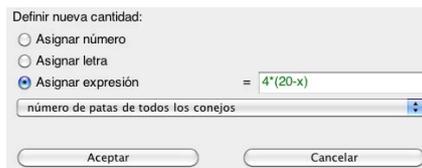


Figura 7

Cuando se hayan declarado todas cantidades necesarias para resolver el problema (lo que supondría haber concluido el paso tercero del MC), el programa activará el panel de construcción de ecuaciones. La construcción de una ecuación supondrá la asignación de una nueva expresión a una cantidad a la que se le había asignado otra previamente. Por esta razón, el término izquierdo de la ecuación se obtendrá del listado de expresiones construidas que aparecerán en un menú desplegable; mientras que el término derecho supondrá la construcción, mediante los botones, de una expresión que dará cuenta de la

⁵⁶ De acuerdo con Filloy, Puig y Rojano (2008), la competencia en la resolución algebraica de problemas verbales aritmético-algebraicos depende básicamente de tres componentes: la competencia en el lenguaje natural en el que está escrito el enunciado; la competencia en el lenguaje del álgebra en el que se representará la ecuación y la competencia en el proceso de conversión del texto expresado en lenguaje natural a lenguaje algebraico. Al tomar la decisión de permitir que el programa gestione la necesidad del uso de paréntesis, estamos soslayando la exigencia de la competencia en el lenguaje del álgebra. Somos conscientes de que esta competencia será necesaria si queremos que los estudiantes puedan utilizar de manera autónoma el MC; pero al reducir esta exigencia, pretendemos que puedan centrar su atención en el proceso de conversión del enunciado del problema al lenguaje del álgebra."

Características de un sistema tutorial inteligente para la resolución de problemas verbales aritmético algebraicos

relación que queda por emplear⁵⁷. Una vez se introduce la ecuación, el programa comprueba su validez y proporciona, en su caso, un mensaje en el que informa que el planteamiento es correcto.

En el caso que nos ocupa (ver Figura 8), únicamente queda por emplearse la relación que establece que el *número de patas* (52) es igual a *número de patas de todos los conejos* ($4(20 - x)$) más *número de patas de todas las gallinas* ($2x$). Así, si tomamos la decisión de escribir de dos formas distintas la cantidad *número de patas de todas las gallinas*, construiríamos la ecuación $2x = 52 - 4(20 - x)$.

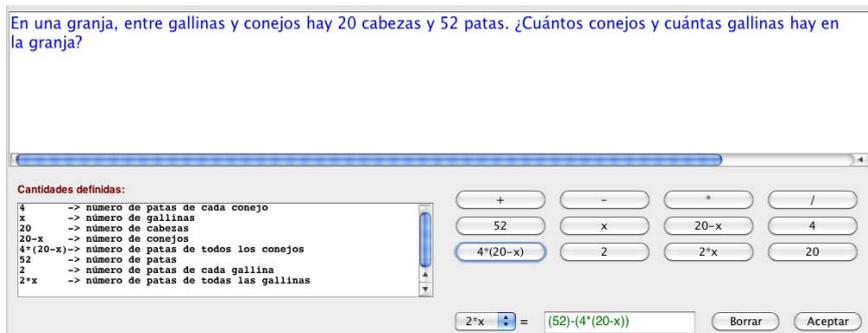


Figura 8

El programa también acepta planteamiento de sistemas de ecuaciones cuando se resuelven problemas de manera algebraica. Para ello, no se necesita dar ninguna instrucción, ni es necesario introducir una nueva información sobre el problema, sencillamente se debe asignar más de una letra en el proceso de definición de cantidades. En la Figura 9, se muestra el instante en que, en una nueva resolución del problema *Conejos y gallinas*, se asigna la letra *y* a la cantidad *número de gallinas* después de haber asignado la letra *x* a la cantidad *número de conejos*.

⁵⁷ La dificultad para expresar una misma cantidad de dos maneras diferentes cuando se construye una ecuación es inherente a la resolución algebraica de problemas. No podemos decir que el entorno ayude a evitar esta dificultad, pero sí que hace explícita la necesidad de igualar una expresión existente de una cantidad (la que aparece en el menú desplegable a la izquierda del signo igual) con una nueva expresión de esta cantidad (que se deberá construir en el cuadro de texto situado a la derecha del signo igual) obtenida al emplear una relación que no se había utilizado en el paso 3 del MC.

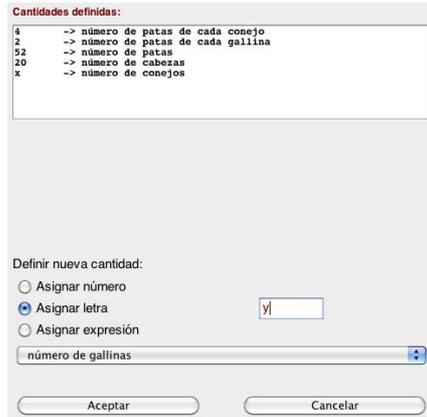


Figura 9

En este caso, cuando finaliza la definición de cantidades, el programa crea espacio para la construcción de dos ecuaciones y supervisa su construcción (ver Figura 10).

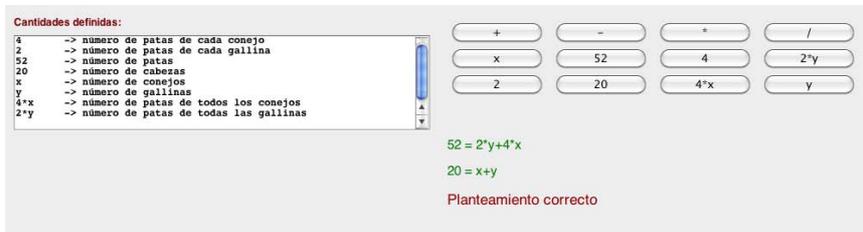


Figura 10

Si las características del problema lo permiten, el programa también es capaz de verificar una resolución aritmética. En este caso, el usuario se limitará a declarar las cantidades conocidas y a construir expresiones aritméticas para calcular las cantidades desconocidas. En la Figura 11, se ofrece el instante de la resolución aritmética del problema *Camisas y pantalones* en el que se construye la expresión que determinará la cantidad *precio de una camisa* (aquella por la que se pregunta en el enunciado). Conviene señalar que hemos decidido mostrar las expresiones aritméticas sin evaluar para hacer explícito el funcionamiento del programa.

Características de un sistema tutorial inteligente para la resolución de problemas verbales aritmético algebraicos

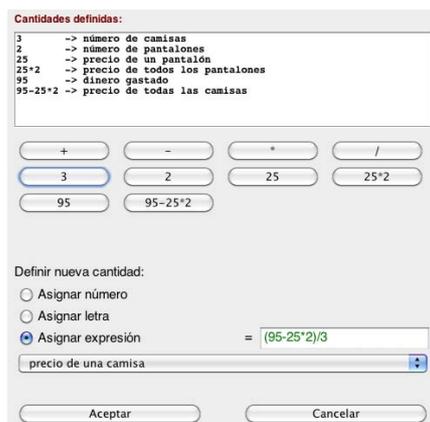


Figura 11

Sin embargo, el hecho de que realicemos una lectura aritmética de un problema no supone que no podamos resolverlo siguiendo los pasos del MC. Por ejemplo, si para la lectura anterior del problema *Camisas y pantalones*, asignamos la letra x a la cantidad *precio de una camisa*, el problema también calificaría de correcto el planteamiento de la ecuación $95 = 3x + 2 \cdot 25$ (ver Figura 12).

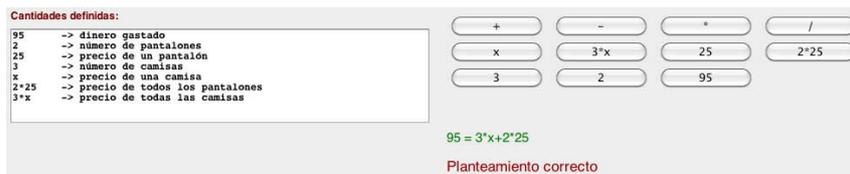


Figura 12

A MODO DE CONCLUSIÓN

El recurso al método cartesiano como modelo de competencia nos ha permitido construir un STI que examina la resolución de problemas verbales y es flexible respecto al método de resolución a emplear y la asignación de expresiones a las cantidades desconocidas. El entorno admite el recurso a soluciones aritméticas y algebraicas y permite la incorporación de nuevas colecciones de problemas. En consecuencia, podría ser posible utilizar el programa como un instrumento para el aprendizaje de la resolución de problemas en los niveles de primaria y secundaria. Por otro lado, también podemos suponer que un estudiante mínimamente instruido en su uso podrá aprovechar las características del tutorización del programa sin la presencia de un tutor humano.

A diferencia de otros entornos diseñados con intenciones similares, hemos construido un STI que es flexible con las decisiones del resolutor y tutoriza de manera detallada el proceso de resolución. Esto nos permitirá en un futuro incorporar al entorno la posibilidad de hacer un diagnóstico fino de las dificultades de un estudiante para proponer una secuencia de problemas que se adapte a sus posibilidades.

Referencias

- Arnau, D. y Puig, L. (2005). Análisis de las actuaciones de los estudiantes de secundaria cuando resuelven problemas verbales en el entorno de la hoja de cálculo. En A. Maz, B. Gómez y M. Torralbo (Eds.), *Investigación en Educación Matemática. Actas del IX Simposio de la SEIEM* (pp. 153-162). Córdoba, España: Servicio de Publicaciones de la Universidad de Córdoba y la SEIEM.
- Beal, C. R. y Arroyo, I. (2002). The AnimalWatch project: Creating an intelligent computer mathematics tutor. En S. Calvert, A. Jordan y R. Cocking (Eds.), *Children in the digital age: Influences of Electronic Media on Development* (pp. 183-198). Westport, CT: Praeger.
- Chang, K. E., Sung, Y. T. y Lin, S. F. (2006). Computer-assisted learning for mathematical problem solving. *Computers & Education*, 46(2), 140-151.
- Filloy, E., Puig, L. y Rojano, T. (2008). El estudio teórico local del desarrollo de competencias algebraicas. *Enseñanza de las Ciencias*, 26(3), 327-342.
- Nathan, M. J. (1990). Empowering the student: prospects for an unintelligent tutoring system. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People* (pp. 407-414). New York: ACM.
- Puig, L. (2003). Historia de las ideas algebraicas: componentes y preguntas de investigación desde el punto de vista de la matemática educativa. En E. Castro, P. Flores, T. Ortega, L. Rico y A. Vallecillos (Eds.), *Investigación en Educación Matemática. Actas del VII Simposio de la SEIEM* (pp. 97-108). Granada: Universidad de Granada.
- Reusser, K. (1993). Tutoring Systems and Pedagogical Theory: Representational Tools for Understanding, Planning, and Reflection in Problem Solving. En S. P. Lajoie y S. J. Derry (Eds.), *Computers as Cognitive Tools*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Sutherland, R. y Rojano, T. (1993). A Spreadsheet Approach to Solving Algebra Problems. *Journal of Mathematical Behavior*, 12(4), 353-383.
- Thompson, P. W. (1989). Artificial Intelligence, Advanced Technology, and Learning and Teaching Algebra. En S. Wagner y C. Kieran (Eds.), *Research Issues in the Learning and Teaching of Algebra* (pp. 135-161). Reston, VA: Lawrence Erlbaum Associates and National Council of Teachers of Mathematics.