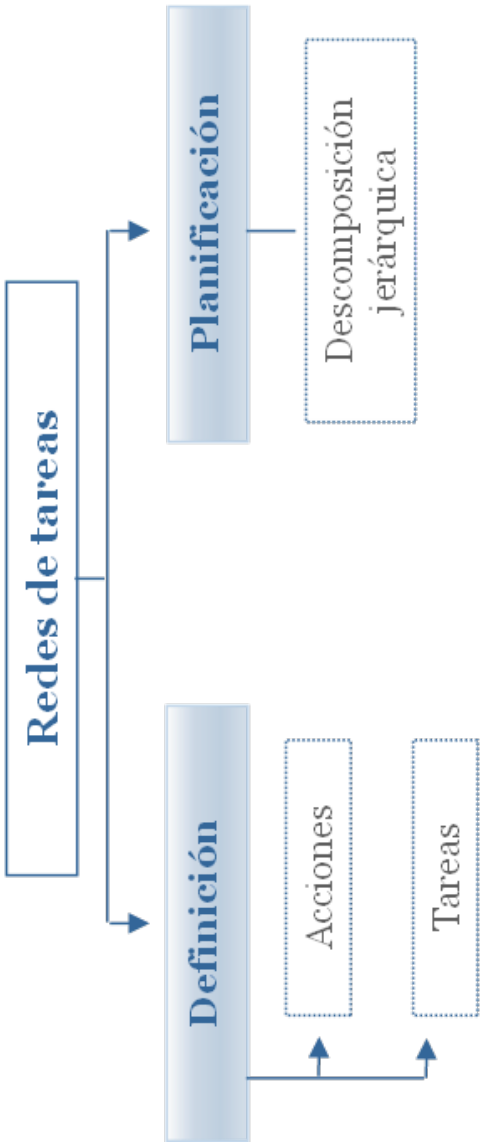
Razonamiento y Planificación Automática

Tema 9. Redes de tareas jerárquicas (HTN)

Índice



9.1. ¿Cómo estudiar este tema?

En este tema presentaremos una estrategia basada en el concepto de «divide y vencerás», por el que plantearemos un problema complejo y lo subdividiremos en tareas más simples. El sistema inteligente se encargará de combinar todas las subtareas disponibles del agente que sean relevantes para el problema, con el objetivo de alcanzar la meta que se le ha planteado.

En el tema trataremos:

- La definición de red de tareas jerárquicas, empleando ejemplos para poder entender cómo se diseñan.
- El mecanismo de planificación que emplean las tareas para construir un plan.

9.2. Definición

En la investigación de planificación de IA, la práctica de planificación (tal como se materializa en los sistemas de planificación implementados) tiende a ir muy por delante de las teorías que explican el comportamiento de esos sistemas. Existe un análisis muy reciente de las propiedades de los sistemas de planificación de orden total y parcial que utilizan operadores de planificación de estilo STRIPS. Los sistemas de planificación de estilo STRIPS, sin embargo, se desarrollaron hace más de veinte años y la mayor parte del trabajo práctico sobre sistemas de planificación de IA durante los últimos quince años se ha basado en la descomposición jerárquica de la red de tareas (HTN).

L a **planificación de la red jerárquica de tareas** (HTA) es una técnica de planificación de la IA que rompe con la tradición de la planificación (Ghallab, 2004). La idea básica detrás de esta técnica incluye:

- Una descripción de estado inicial,
- Una red de tareas inicial como un objetivo a alcanzar
- Y un conocimiento de dominio, que consiste en redes de tareas primitivas y compuestas.

Una **red de tareas** representa una jerarquía de tareas, cada una de las cuales puede ejecutarse, si la tarea es primitiva, o ser descompuesta en subtareas refinadas.

Ideas clave

El **proceso de planificación** comienza descomponiendo la red de tarea inicial y continúa hasta que se descompongan todas las tareas compuestas, es decir, se encuentre una solución. La **solución** es un plan que equivale a un conjunto de tareas primitivas aplicables al estado mundial inicial.

Además de ser un factor que rompe la tradición, la planificación de la HTN también parece ser controvertida. La controversia radica en su requisito de conocimiento del dominio bien concebido y estructurado. Es probable que dicho conocimiento contenga abundante información y orientación sobre cómo resolver un problema de planificación, codificando así más la solución de lo que se previó para las técnicas de planificación clásicas. Este conocimiento estructurado y rico brinda una ventaja principal a los planificadores de HTN en términos de velocidad y escalabilidad cuando se aplica a problemas del mundo real y se compara con sus contrapartes en el mundo clásico (Georgievski, 2014).

La mayor contribución hacia la planificación de HTN ha surgido después de la propuesta del **planificador jerárquico simple (SHOP)** (Nau D. S., 1999) y sus sucesores. SHOP es un planificador basado en HTN que muestra un rendimiento eficiente incluso en problemas complejos, pero a costa de proporcionar un conocimiento de dominio bien escrito y posiblemente algorítmico. La disputa sobre si proporcionar mucho conocimiento a un planificador debería considerarse una trampa en el mundo de la planificación de la inteligencia artificial sigue vigente (Nau D. S., 1999).

9.3. Planificación por medio de red de tareas

Idea principal: muchas tareas en la vida real ya tienen una estructura jerárquica incorporada. Por ejemplo: una tarea computacional, una misión militar o una tarea administrativa.

Sería una pérdida de tiempo construir planes desde los operadores individuales que forman las acciones propias del trabajo a realizar. Usar la jerarquía incorporada en el dominio ayuda a escapar de la explosión exponencial de las posibles combinaciones que tendrían las acciones atómicas.

Ejemplo de aplicación: la actividad de construir una casa consiste en obtener los permisos necesarios, encontrar un constructor, construir el exterior/interior, etc. En el enfoque de las HTN, se utilizan operadores abstractos al igual que operadores primitivos durante la generación del plan.

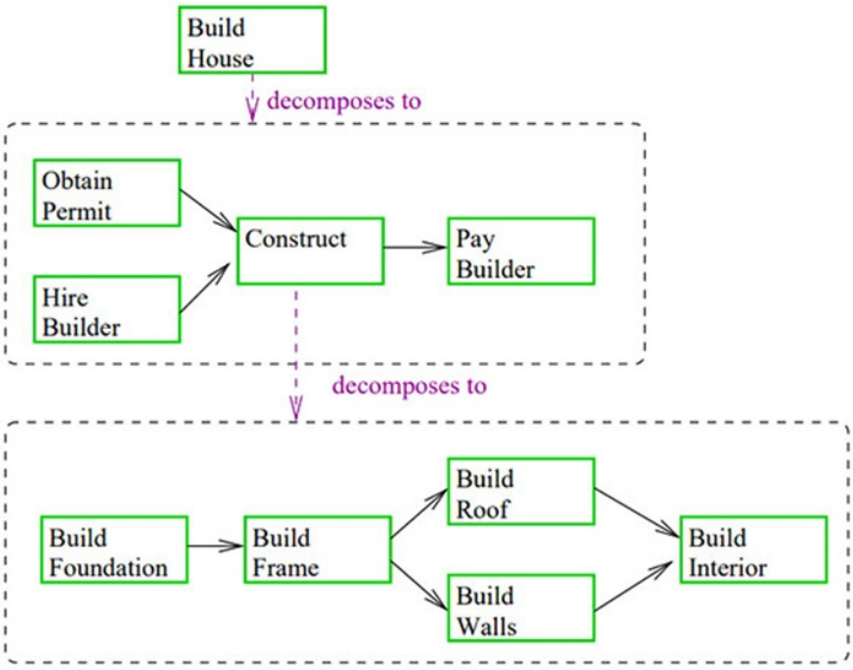


Figura 1. Ejemplo de descomposición de tareas. Fuente: (Onder, 2020).

En general, los componentes de un problema que intentamos resolver por medio de un planificador HTN se basan en las propiedades que define el entorno y el dominio de las tareas que componen las redes que estableceremos para crear los planes. En el caso de los modelos basados en agentes, obtendremos las propiedades del entorno por medio de los sensores del agente y crearemos planes que iremos ejecutando para resolver los problemas a los que este deba hacer frente a lo largo del tiempo.

Asimismo, tendremos en cuenta las acciones que ejecutemos, de tal modo que apliquemos los efectos de dichas acciones sobre el mundo a la hora de evaluar la viabilidad de un plan.

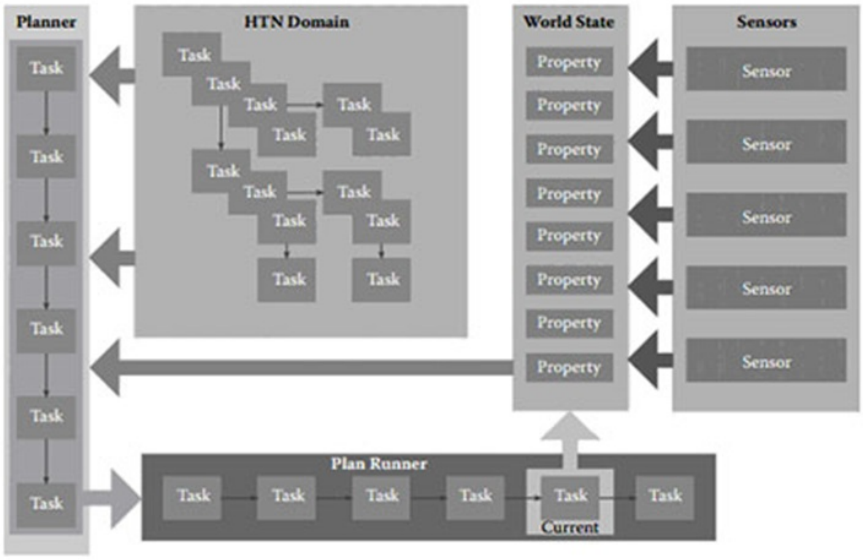


Figura 2. Vista general de un HTN. Fuente: .

9.4. Descomposición jerárquica

HTN es adecuado para dominios donde las tareas se organizan naturalmente en una jerarquía. Para construir estos modelos empleamos operadores abstractos para comenzar un plan. Utilizamos **técnicas de planificación de orden parcial** y **descomposición de acciones** para llegar al plan final.

Una vez que hemos terminado de descomponer el plan, nos encontramos con un conjunto ordenado de operadores primitivos.

No obstante, lo que debe considerarse primitivo es subjetivo: lo que un agente considera primitivo pueden ser los planes de otro agente. Por tanto, la descomposición de un plan puede derivar en acciones complejas que, desde el punto de vista del plan, son consideradas operadores primitivos pero que, a la hora de implementar las acciones finales, debamos descomponerlas en otras por medio de controladores que en la jerarquía del agente se encarguen de detallar esas acciones. Recordemos los primeros modelos sobre controladores apilados para tareas complejas.

A la hora de descomponer las tareas tendremos una **librería del plan** que contendrá tanto las tareas primitivas como las que no lo son. Las no primitivas (o compuestas) tendrán un conjunto de **precondiciones** de ejecución y un conjunto de **efectos** producidos en el entorno.

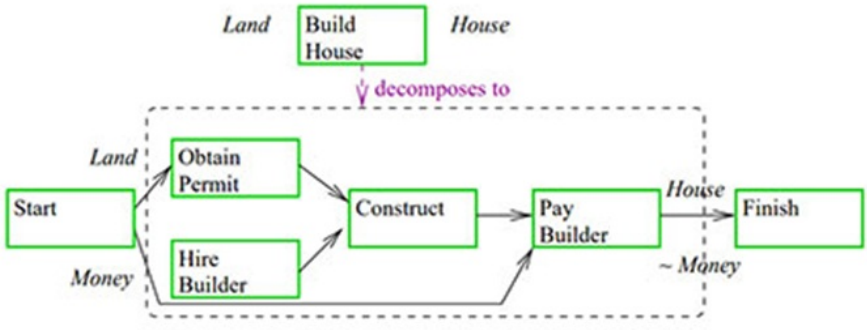


Figura 3. Ejemplo de descomposición de tareas. Fuente: (Onder, 2020).

Pero podemos tener varias tareas que permitan construir un plan, como en la figura anterior. Por ejemplo, podríamos tener otra serie de tareas que nos permitan edificar una casa.

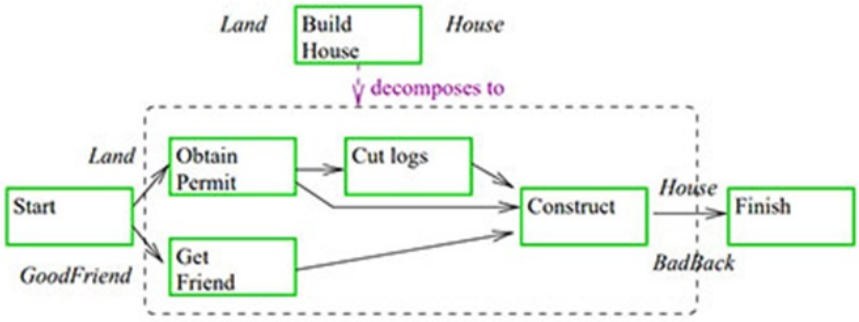


Figura 4. Ejemplo de descomposición de tareas. Fuente: (Onder, 2020).

Ambos conjuntos de tareas estarían a disposición del planificador para poder crear distintos planes, teniendo en cuenta que ambos medios de creación del plan tienen distintos efectos en el entorno (tener menos dinero o tener la espalda mal) al margen de obtener el efecto deseado (tener la casa).

Definición de las tareas

Así, cuando definamos las tareas, tendremos que establecer dos elementos principales: las **precondiciones** necesarias que deben darse en el entorno para poder ejecutar una tarea y los **efectos** que se crean en el entorno.

Por ejemplo:

Action(*BuyLand*, PRECOND:*Money*, EFFECT: Land ∧¬ *Money*)
Action(*GetLoan*, PRECOND:*GoodCredit*, EFFECT: Money ∧ *Mortgage*)
Action(*BuildHouse*, PRECOND:*Land*, EFFECT: *House*)

Action(*GetPermit*, PRECOND:*Land*, EFFECT: *Permit*)
Action(*HireBuilder*, EFFECT: *Contract*)
Action(*Construct*, PRECOND:*Permit* ∧ *Contract*,
EFFECT: *HouseBuilt* ∧¬ *Permit*)
Action(*PayBuilder*, PRECOND:*Money* ∧ *HouseBuilt*,
EFFECT: ¬ *Money* ∧ *House* ∧¬ *Contract*)

Figura 5. Ejemplo de precondiciones y efectos. Fuente: (Onder, 2020).

Cuando tengamos que desarrollar un plan, descompondremos las tareas de modo que empleemos las acciones del sistema. Por ejemplo, para el primer caso tendríamos que la descomposición del plan se produce de la siguiente manera:

Plan	Pasos (P1:GetPermit, P2: HireBuilder, P3: Construction, P4:PayBuilder)	
	Ordenación:	Inicio < P1 < P2 < P3 <P4 < Final Inicio < P2 < P3
	Con las propiedades del estado que se desprenden de los distinto efectos.	

Tabla 1. Descomposición de un plan.

De esta manera, podremos descomponer varios planes que empleen distintas acciones en distintos contextos para obtener planes posibles que resuelvan el problema.

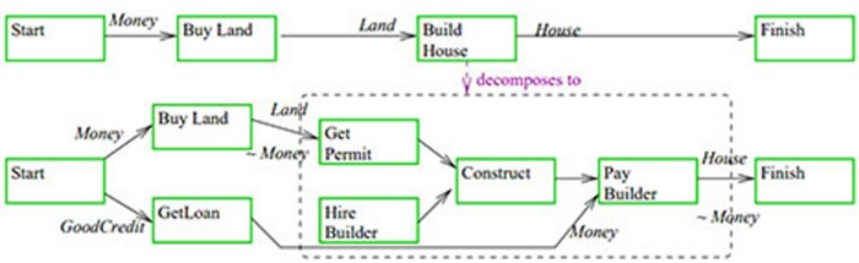


Figura 6. Ejemplo de descomposición de tareas. Fuente: (Onder, 2020).

9.5. Referencias bibliográficas

Georgievski, I. y. (2014). *An Overview of Hierarchical Task Network Planning.*

Groningen: Johann Bernoulli Institute for Mathematics and Computer Science University of Groningen. Obtenido de https://arxiv.org/pdf/1403.7426v1.pdf

Ghallab, M. N. (2004). *Automated Planning: theory and practice.* San Francisco: Elsevier.

Humphrey, T. (2015). Exploring HTN Planners through Example. *Game AIPro 2*, (pp. 149-167.

Nau, D. S. (1999). SHOP: Simple Hierarchical Ordered Planner. *IJCAI*, (págs. 968-975).

Nau, D. S. (2007). Current Trends in Automated Planning. *AI Magazine*, 28(4), 43-58.

Onder, N. (. (09 de 05 de 2020). *Computer Science Course*. Obtenido de Michigan Technological University: https://pages.mtu.edu/~nilufer/classes/cs5811/2010-fall/lecture-slides-3e/cs5811-ch11b-htn

Aplicación y desarrollo de planes por medio de HTN

Molina, M. (2006). Planificación jerárquica HTN. En Molina, M (Ed.), *Métodos de resolución de problemas* (pp. 251-307). Madrid: Fundación General Universidad Politécnica de Madrid.

El capítulo seis de este libro presenta varios ejemplos de aplicación y desarrollo de planes por medio de HTN.

HTN

Lección de la Universidad de Edimburgo que presenta de modo sencillo las HTN.

Accede al vídeo a través del aula virtual o desde la siguiente dirección web:https://www.youtube.com/watch?v=7rHi4FwlJw4

Ejemplo de creación de tareas jerárquicas

Humphreys, T. (2013). Exploring HTN Planners through Example. En Rabin, S (Ed.), *Game AI Pro* (pp. 149-167). Florida: CRC Press. Recuperado de: http://www.gameaipro.com/GameAIPro/GameAIPro_Chapter12_Exploring_HTN_Planners_through_Example.pdf

Aquí podemos ver un ejemplo aplicado de creación de tareas jerárquicas.

Principles of Artificial Intelligence

Accede a la página web a través del aula virtual o desde la siguiente dirección web: https://www.csee.umbc.edu/courses/671/fall09/

Curso de la universidad de Maryland a cargo del profesor Des Jardins.

Bibliografía adicional

Molina, M. (2006). *Métodos de resolución de problemas*. Madrid: Fundación General Universidad Politécnica de Madrid.

Test

1. ¿Qué afirmación es verdadera?

   A. HTN es un ejemplo de STRIPS.

   B. HTN es distinto de STRIPS.

   C. HTN es admitido por todo el mundo.

2. HTN se basa en:

   A. Planes de orden completo.

   B. Tareas y subtareas.

   C. Valores aleatorios de planes.

3. Una tarea…

   A. Puede ser primitiva o no.

   B. No se puede descomponer en subtareas.

   C. Todas se descomponen en tareas.

4. Una tarea primitiva está compuesta de:

   A. Precondiciones y valores.

   B. Precondiciones y efectos.

   C. Efectos y utilidad.

5. Los efectos de una tarea:

   A. Afectan a valores de propiedades del mundo.

   B. Forman todos parte del resultado objetivo de la búsqueda.

   C. Se descomponen en subefectos.

6. Dado un dominio de HTN:

A. Solo existe un plan que es solución al problema.

B. Nunca produce el mismo resultado para una misma búsqueda.

C. Puede crear varios planes que sean solución al problema y dependerá de las propiedades de cada instante el que se creen unos u otros.

7. Los HTN:

A. Requieren de mucho conocimiento a priori de las tareas posibles.

B. No requieren de tareas.

C. Crea todas las tareas y subtareas por búsqueda.

8. Las tareas pueden tener acciones que no se implementen directamente el entorno:

A. Sí, porque pueden ser parte de una jerarquía de controladores.

B. No, todas las tareas deben ejecutar acciones concretas del mundo.

C. No, porque las tareas pueden descomponerse siempre en subtareas.

9. Las HTN:

A. Emplean planificación de orden total.

B. Planificación de orden parcial.

C. No emplean planificación.

10. En una HTN:

A. Da igual el orden de aplicación de todas las tareas.

B. Todas las tareas tienen que ejecutarse en un orden determinado.

C. Algunas tareas pueden ejecutarse en órdenes no establecidos mientras que guarden el orden parcial.