

Modelado de un dominio de planificación automática de un robot reforestador de árboles

Modeling of an automatic planning domain of a tree reforestation robot.

Jose David Cano Sierra¹, María José Vargas Pescador²,
Deisy Nataly Ramírez Ruiz³, Juan-Manuel Chaguendo-Benavides⁴

Tipo de Artículo: Investigación científica y tecnológica.
Recibido: 15/12/21 **Aprobado:** 30/05/22 **Publicado:** 18/12/2022

Resumen: La deforestación es uno de los problemas ambientales que tiene Colombia, llevándose millones de ecosistemas a lo largo del tiempo. Afortunadamente, se ha venido haciendo actividades de reforestación de árboles a través de grupos como EPM, Emgesa, Ecopetrol, el Gobierno nacional y Antioquia. Estas actividades requieren de un gran esfuerzo en la planificación y ejecución de cómo se van a sembrar y reforestar los árboles. Por ello, se han propuesto trabajos relacionados con el proceso de reforestación utilizando un robot inteligente. Sin embargo, estas propuestas tienen un enfoque de programación en Arduino y no un enfoque de planificación automática. No obstante, uno de los desafíos al aplicar planificación automática a tareas del mundo real es el modelado de dominios, puesto que especificar las condiciones previas

correctas y los efectos de las acciones es una tarea difícil. En este artículo presentamos la definición de un dominio de planificación automática de un robot reforestador de árboles en lenguaje PDDL que ayuda a representar un problema de siembra de semillas y reforestación. Este dominio también ayuda a obtener un plan que guía a un robot a realizar actividades de reforestación de árboles. Para probar nuestro dominio, creamos, diseñamos e implementamos un problema de reforestación de siete árboles en nueve puntos distribuidos en siete coordenadas. Este problema se codificó en lenguaje de especificación PDDL, teniendo como resultado un plan con 28 acciones o conjunto de estados. Para realizar la plantación de cada árbol se ejecutaron cuatro acciones en el siguiente orden: pickup, go, plant y back.

1 Estudiante de Ingeniería en Software. Institución Universitaria Tecnológico de Antioquia. Colombia, Medellín. jose.cano@correo.tdea.edu.co

2 Estudiante de Ingeniería en Software. Institución Universitaria Tecnológico de Antioquia. Colombia, Medellín. maria.vargas99@correo.tdea.edu.co

3 Estudiante de Ingeniería en Software. Institución Universitaria Tecnológico de Antioquia. Colombia, Medellín. deysi.ramirez@correo.tdea.edu.co

4 Candidato a doctor en Computación. Universidad Politécnica de Cataluña. Investigación financiada por el Ministerio de Ciencia, Innovación y Universidades de España (PRE2018-086544) y cofinanciada por FSE. España, Barcelona. juan.manuel.chaguendo@upc.edu. ORCID: <https://orcid.org/0000-0002-8535-1736>

Palabras clave: Planificación automática, dominio, Lenguaje PDDL, reforestación, deforestación.

Abstract: Deforestation is one of Colombia's environmental problems, taking away millions of ecosystems over time. Fortunately, tree reforestation activities have been carried out through groups such as EPM, Emgesa, Ecopetrol, the national Government and Antioquia. These activities require great effort in your planning and execution of how the trees are to be planted and reforested. For this reason, works related to the reforestation process using smart robots have been proposed. However, these proposals have an Arduino scheduling approach and not an automatic scheduling approach. However, one of the challenges in applying automatic scheduling to real-world tasks is domain modeling, since specifying the correct preconditions and effects of actions is a difficult task. In this article we present the definition of an automatic planning domain of a tree reforestation robot in PDDL language that helps to represent a problem of planting seeds and reforestation. This domain also helps to obtain a plan that guides a robot to carry out tree reforestation activities. To test our mastery, we created, designed and implemented a reforestation problem of 7 trees, in 9 points distributed in 7 coordinates. This problem was coded in the pddl specification language. Resulting in a plan with 28 actions or set of states. To plant each tree, four actions were carried out in the following order: pickup, go, plant and back.

Keywords: Automated planning, domain, PDDL, reforestation, deforestation.

I. Introducción

En Colombia, durante el tercer trimestre del 2019, se identificaron alrededor de 3.000 hectáreas deforestadas en los departamentos de Caquetá, Meta y Guaviare, sin contar los demás departamentos del país afectados por la misma problemática [1], [2]. Por ello, la reforestación cumple un papel fundamental en el mantenimiento de los ecosistemas terrestres, contribuyendo con la recuperación de las cuencas hidrográficas, la estructura ecológica, la conservación de la

biodiversidad y la seguridad alimentaria [2]. Afortunadamente, en Colombia, durante los últimos años, grupos como EPM, Emgesa [3] y Ecopetrol tienen programas que fomentan la reforestación y realizan sembrado de millones de árboles cada año. También el Gobierno nacional y Antioquia [4] se han sumado a estas iniciativas para la conservación de los bosques de nuestro país. Sin embargo, llevar a cabo un proceso de siembra o reforestación requiere de un gran esfuerzo en su planificación y ejecución, es decir, la hoja de ruta de dónde se empieza y dónde se termina, por lo cual se hace necesaria la implementación de las nuevas tecnologías de la información en busca de una solución a la problemática de la deforestación. Es así que se han propuesto trabajos relacionados con el proceso de siembra de semillas o reforestación utilizando robots inteligentes [5], [6], [7], [8], [9], [10]. No obstante, estas propuestas presentan un enfoque de programación utilizando Arduino y no un enfoque de planificación automática que pretende abordar el problema de generar comportamientos autónomos a partir de un modelo que describe cómo funcionan las acciones en un dominio de interés y cuál es el estado inicial y el estado final a alcanzar. En planificación automática, especificar las condiciones previas correctas y los efectos de las acciones es una tarea difícil, especialmente cuando el dominio de interés requiere de atención detallada.

Este artículo presenta la definición de un dominio de planificación automática de un robot reforestador de árboles en lenguaje PDDL, permitiendo representar un problema de siembra de semillas y reforestación que se asemeja a uno real, que también ayudará a obtener un plan (conjunto de estados) que guiará a un robot o agente inteligente en la realización de actividades de reforestación de árboles. Para ello, se siguieron cuatro fases del proceso de investigación: fase 1: definición del objeto de investigación; fase 2: formulación de la propuesta de investigación; fase 3: ejecución de la investigación, y fase 4: síntesis de la investigación. Además, para probar nuestro dominio, creamos, diseñamos e implementamos un problema supuesto de reforestación de árboles, enmarcado en el plan de desarrollo destinado para el año 2022 en el territorio antioqueño, que

tiene como finalidad desplegar un proyecto de reforestación en Caucasia. El proyecto consiste en que en 2 hectáreas del territorio se siembren siete árboles de Guayacán en nueve puntos distribuidos en siete coordenadas. Este problema se codificó en lenguaje PDDL.

Como producto de la investigación se tiene un dominio diseñado, desarrollado e implementado en lenguaje PDDL, conformado por cuatro acciones: *pickup*, *go*, *plant* y *back*. Este dominio puede utilizarse de manera simple para representar un problema de un robot reforestador basado en planificación automática en el contexto de siembra y reforestación de árboles. También en la ejecución del dominio con el problema en la herramienta web PDDL Editor se tuvo como resultado un plan conformado por 28 acciones o conjunto de estados. En este caso, el robot realizó cuatro acciones por cada árbol a sembrar; por consiguiente, hizo su cometido de plantar siete árboles mediante 28 acciones.

El presente artículo está organizado de la siguiente manera: En la Sección II presentamos el marco teórico. En la sección III describimos los antecedentes, junto con el planteamiento del problema. En la sección IV se definen los objetivos. Luego, en la sección V se muestra la metodología de investigación utilizada. En la sección VI están los trabajos relacionados. En la sección VII se definen el dominio y el problema en PDDL. Después, en la sección VIII, se muestran los resultados. En la sección IX están las conclusiones y trabajo futuro. Y, finalmente, las referencias en la sección X.

II. Marco teórico

La planificación automática aborda el problema de generar comportamientos autónomos (planes) a partir de un modelo que describe cómo funcionan las acciones en un dominio de interés y cuál es el estado inicial y el estado final a alcanzar. Para ello, la planificación automática opera sobre representaciones explícitas de estados y acciones que se expresan a través de lenguajes dedicados [11]. De acuerdo con esa expresión, STRIPS es un lenguaje formal de entradas para la generación de planes, es decir que describe el estado inicial

del mundo mediante un conjunto completo de literales básicos y las metas son definidas como una conjunción proposicional [12]. Por otra parte, está PDDL, sigla de Planification Domain Description Language, una familia de lenguajes de acción que permite representar problemas de planificación mediante la formalización proposicional [13]. En esta representación de problemas se definen objetos, los cuales serían seres o cosas que forman parte del "mundo" y también requieren de operadores, que son conocidos como predicados, o una acción que cambia el estado de uno o varios objetos [14]. Dentro de los posibles operadores o predicados que pueden usarse para la precondition, están:

- and / or / not
- imply <cond> <effect>
(imply (and (at ?r1 ?w1) (at ?r2 ?w2)) (= ?w1 ?w2))
- exists <variable> <literal>
(exists (?r - rover)(at ?r ?w2))
- forall <variables> <literal>
(forall (? - rover)(at ?r ?w2

Para los efectos, los posibles operadores son:

- and / not
- forall <variables> <effect>
- when <expression> <effect>

En un problema de planificación, un estado es una asignación particular al conjunto de variables, así que un conjunto de estados se define como una forma de describir el estado de creencias iniciales del agente [15], en este caso, el agente que está esperando indicación de qué hacer. Para ello se establecen dos estados: inicial y final. El primero presenta propiedades iniciales de los objetos de nuestro "mundo"; el segundo se refiere a las propiedades finales de nuestros objetos [14]. También para definir un problema en planificación se requieren acciones, las cuales son conjuntos representativos de transiciones de estado y están conformadas por predicados que funcionan como precondiciones o poscondiciones. Las primeras son necesarias para que las acciones puedan ser

ejecutadas y las últimas son el resultado de la efectiva realización de la acción [13]. Justamente en el dominio es donde se crean dichas acciones. Este es un archivo con extensión .pddl (domain.pddl). En él se incluyen los predicados, los cuales definen los tipos de valores proposicionales que pueden existir en el dominio. Los valores proposicionales son atómicos y bivalentes, es decir, están sometidos a un valor de "verdadero" o "falso"; además, pueden ser propios de un único objeto o pueden definirse entre varios objetos. Las instancias de todos los predicados para un posible problema conforman el conjunto de proposiciones denominado F (las metas del problema) [13].

Por otra parte, se encuentra el problema, que es un archivo con extensión .pdd (problema.pddl), en el cual se establece el estado inicial, esto es, los predicados que son ciertos al principio, y un estado final, los predicados que deben ser ciertos al final tras aplicar las acciones definidas en el dominio [13], es decir, busca generar una secuencia limitada de acciones (llamadas planes) para ser ejecutadas por agentes inteligentes. Estas acciones deben llevarse al agente desde un estado inicial hacia un estado final u objetivo dado, llamado meta [16]. La formalización de un problema de planificación (P) puede definirse como $P = (L, A, I, F)$, donde:

- L: Conjunto de literales que representan los hechos que tienen relevancia en el problema
- A: Conjunto de acciones que transforman un estado en otro
- I: El estado inicial del problema
- F: Las metas del problema [16]

Finalmente, está el plan, que consiste en una secuencia de uno o más pasos, cada uno de los cuales tiene un conjunto de atributos. Cada paso tiene el nombre del operador que debe ejecutarse y una obligación. En síntesis, el plan es el resultado de la ejecución del problema y el dominio [17].

Reforestación: "Establecimiento de plantaciones forestales en tierras temporalmente no arboladas, que son consideradas como bosques" [18].

III. Antecedentes y planteamiento del problema

La deforestación, definida como "el resultado de las actividades negativas hechas por el hombre, que simultáneamente generan una serie de amenazas para la vida misma, como lo es la pérdida de bosque en el planeta, disminución de la biodiversidad, afectación a suelos, entre otros" [19], es uno de los principales problemas ambientales que tiene Colombia, habiéndose llevado ya millones de ecosistemas a lo largo del tiempo, según el boletín del IDEAM, donde se informa que durante el tercer trimestre del 2019 se identificaron alrededor de 3.000 hectáreas deforestadas en los departamentos de Caquetá, Meta y Guaviare [1], sin ser estos los únicos departamentos del país que sufren esta problemática [2].

Por otra parte, EPM, con su programa Fomento a la Reforestación, ha sembrado 1,8 millones de árboles cada año; Emgesa, con el programa Bosque Renace, ha restaurado 690 hectáreas de bosque entre el 2012 y el 2018; Ecopetrol cuenta con 18 proyectos, recuperando y reforestando más de 1.100 hectáreas [3]. También el Gobierno nacional ha promovido actividades como la Gran Sembratón Nacional, que tiene como objetivo sembrar 5 millones de árboles en todo el país. Antioquia se suma a esta iniciativa, logrando sembrar cerca de 10 mil árboles en 13 hectáreas [4]. Estas iniciativas son fundamentales para la conservación de los bosques de nuestro país, ya que "la reforestación cumple un papel fundamental para el mantenimiento de los ecosistemas terrestres, puesto que contribuye con la recuperación de las cuencas hidrográficas, la estructura ecológica, la conservación de la biodiversidad y seguridad alimentaria" [2].

La presente propuesta busca definir una situación problema, a manera de suposición, que puede presentar una entidad dedicada a la reforestación de árboles, con el fin de tener una representación del estado inicial y el estado final, mediante el lenguaje PDDL, a través de la definición de un dominio de planificación automática que describe

cómo funcionan las acciones y, así, obtener un plan (conjunto de estados) que guiará a un robot inteligente o agente inteligente en la realización de actividades de reforestación de árboles.

Por lo tanto, se plantea la siguiente pregunta de investigación: ¿es posible representar un supuesto problema de siembra de semillas y reforestación de árboles a través de la definición de un dominio de planificación automática utilizando lenguaje PDDL?

Hipótesis de investigación: Si es posible representar un supuesto problema de siembra de semillas y reforestación de árboles a través de la definición de un dominio de planificación automática utilizando PDDL, se obtendrá un plan (conjunto de estado) que guiará la actividad de reforestación.

IV. Objetivos

Objetivo general

Definir un dominio de planificación automática de un robot reforestador de árboles en lenguaje PDDL que ayude a representar un problema de siembra de semillas y reforestación.

Objetivos específicos

- Establecer el marco de referencia acerca de propuestas de proceso de siembra de semillas o reforestación utilizando robot inteligente.
- Realizar un análisis comparativo de los trabajos relacionados encontrados.
- Desarrollar e implementar el dominio y problema en lenguaje PDDL.
- Realizar análisis de resultados obtenidos sobre el plan (conjunto de estados).

V. Metodología

Esta investigación es de tipo descriptivo, es decir que se preocupa por describir algunas características fundamentales de conjuntos homogéneos de fenómenos utilizando criterios sistemáticos que permitan poner de manifiesto su

estructura o comportamiento. Para el desarrollo de los objetivos dados, se siguió un modelo general para el proceso de investigación que consta de cuatro fases. El modelo no es lineal, lo que refleja el carácter dinámico y procesal de la investigación, de modo que no se conciba al proceso como algo que tiene un principio y un fin definitivos, sino más bien como un trabajo continuo [20]. En la Figura 1 se muestran estas fases.

Figura 1. Fases del proceso de investigación.

Fases del proceso de investigación

- Fase 1.** Definición del objetivo de investigación
- Fase 2.** Formulación de propuesta de investigación
- Fase 3.** Ejecución de la investigación
- Fase 4.** Síntesis de la investigación

Fuente: Elaboración propia (2021)

Fase 1. Definición del objeto de investigación: su propósito es hacer la elección del objeto, fenómeno o proceso para el cual se requiere generar conocimiento socialmente nuevo. En esta fase se seleccionó el área temática (planificación automática y reforestación) y se hizo la elección del objeto de investigación (dominio de planificación automática de un robot reforestador).

Fase 2. Formulación de la propuesta de investigación: su propósito es la organización del proyecto de investigación indicando claramente su alcance, su soporte teórico y la estrategia a seguir para el logro de los resultados planteados. En esta fase se realizó el planteamiento del problema (sección III), se hizo la delimitación de la investigación (objetivo general y objetivos específicos), se elaboró el marco teórico haciendo una revisión de la literatura, la cual se recuperó de las fuentes y herramientas proporcionadas por la Biblioteca Digital de la Universidad Politécnica de Cataluña (UPC), se tuvieron en cuenta motores de búsqueda de carácter académico o científico, como Google

Scholar y Microsoft Academic, y algunas bases de datos bibliográficas especializadas, como Web of Science y IEEE Xplore (sección II y trabajos relacionados acerca de propuestas de proceso de siembra de semillas o reforestación utilizando robot inteligente) y se realizó el diseño de la investigación (pregunta e hipótesis de investigación).

Fase 3. Ejecución de la investigación: su propósito es obtener, mediante observación o experimentación, los datos requeridos sobre el objeto de investigación elegido y efectuar el procesamiento de estos. En esta fase se diseñó e implementó el dominio y el problema en lenguaje PDDL (archivo dominio.pddl y archivo problema.pddl) y se realizó la ejecución del dominio y el problema en la herramienta PDDL Editor.

Fase 4. Síntesis de la investigación: su propósito es tener una elaboración teórica que refleje el nuevo conocimiento generado como resultado del proceso investigativo. En esta fase se realizó el análisis y síntesis de la información (análisis comparativo de trabajos relacionados encontrados y conclusiones).

VI. Trabajos relacionados

En la revisión crítica del estado del arte se obtuvieron los siguientes trabajos relacionados acerca de propuestas de siembra de semillas o reforestación utilizando robot inteligente.

Diseño de un robot móvil de dos hileras para trasplante de plántulas de arroz en sembríos de la costa norte peruana

El proyecto tuvo como objetivo el diseño de un robot para el trasplante de plántulas de arroz en Perú. El prototipo debía tener la suficiente potencia para avanzar y girar en movimiento

constante y la suficiente energía para un funcionamiento autónomo. Para el diseño electrónico se escogió la placa Arduino Mega y se programó en su interfaz Arduino IDE, además de usar sensores para controlar el sistema y evitar que se desvíe de la ruta programada [5].

Diseño en Autocad de robot sembrador de semillas de fresa: Fresabot III

El robot sembrador de semillas de fresa tiene como objetivo mejorar los tiempos de ejecución y la calidad en la siembra en invernaderos. El dispositivo está construido con un sistema de tracción de oruga, que consiste en un conjunto de eslabones modulares que permiten un desplazamiento estable en terrenos irregulares, además de servomotores y una broca para cavar la tierra y facilitar la siembra. La energía es suministrada por una batería de litio y está programado con el software Arduino IDE [6].

Diseño y construcción de un prototipo de robot tetrápodo sembrador de semillas para leguminosas

El proyecto es una herramienta didáctica y se encuentra en una fase temprana de ejecución. Presenta un prototipo de robot tetrápodo cuyo diseño le permite movilizarse fácilmente entre obstáculos en lugares de difícil acceso. Su función es dosificar, detectar e introducir semillas con una profundidad específica en tierras tratadas para cultivos de leguminosas, mejorando la eficiencia y productividad en la siembra [7].

Simulación de aplicación de un robot forestador

En este proyecto vemos cómo podría ser la actividad de plantar árboles, de acuerdo con una simulación realizada en el software Vensim, el cual, a diferencia de nuestro dominio, tiene un monitoreo total del proceso forestador,

implicando altos costos por ser un proceso prolongado en el tiempo, el cual puede ser útil para empresas encargadas de talar árboles para materia prima, ya que estas tienen grandes capitales para todo este seguimiento [8].

Ingeniería de detalle para un robot sembrador de semillas de maíz

Este proyecto se basa en el análisis e investigación del proceso de la siembra del maíz. Se buscó encontrar los mejores métodos y herramientas para optimizar el proceso. Los resultados obtenidos sirvieron de base para la elaboración de un prototipo que responde con eficiencia y soluciona las problemáticas planteadas [9].

Implementación de un prototipo de robot sembrador de papa en terrenos sin inclinación para pequeños productores

El robot sembrador de papa se creó en dos etapas. La primera fue la parte mecánica y estructural del robot, la cual tiene como objetivo transportar la semilla al lugar de la siembra. La segunda parte es la implementación electrónica, que se encarga de la potencia para movilizarse en el terreno y transportar la semilla. Los motores son programados con microprocesadores de Arduino que controlan la velocidad, y el sistema de control se realiza con una aplicación inalámbrica para realizar el proceso autónomo de siembra [10].

Tabla 1. Resumen de trabajos relacionados acerca de propuestas de siembra de semillas o reforestación de árboles utilizando robot inteligente.

Nombre	Año	Lugar	Lenguaje	Ventajas	Desventajas
Diseño de un robot móvil de dos hileras para trasplante de plántulas de arroz en sembríos de la costa norte peruana	2020	Lima, Perú	Arduino IDE	Para la elaboración del robot se utilizaron materiales resistentes a la corrosión y de acero inoxidable.	Los motores del robot no cuentan con la resistencia suficiente para soportar su propio peso y para moverse con fluidez en el terreno de siembra.
Diseño en Autocad de robot sembrador de semillas de fresa: Fresabot III	2019	Bogotá, Colombia	Arduino IDE	El robot está diseñado para ser creado con materiales reciclables, lo que disminuiría su costo de producción.	El robot podría verse afectado por la calidad de los suelos. Si estos no son óptimos para su funcionamiento, este podría presentar daños.
Diseño y construcción de un prototipo de robot tetrápodo sembrador de semillas para leguminosas	2018	Sangolquí, Ecuador	Arduino IDE	El robot tiene una interfaz amigable con el usuario. Las instrucciones las recibirá de forma inalámbrica por medio de <i>bluetooth</i> .	El movimiento del robot tiene menor rendimiento en áreas con obstáculos, en comparación con un terreno plano.
Simulación de aplicación de un robot forestador	2018	Argentina	N/A	El objetivo del robot es ayudar al trabajador en el proceso de siembra, logrando así mejorar la eficiencia y la productividad.	Los resultados obtenidos en el estudio se realizaron con base en supuestos. Para obtener datos más concretos, este debe realizarse en el campo, donde se llevará a cabo la siembra.

Nombre	Año	Lugar	Lenguaje	Ventajas	Desventajas
Ingeniería de detalle para un robot sembrador de semillas de maíz	2019	Bogotá, Colombia	Arduino IDE	En la simulación del robot se implementó un sistema de navegación que le permite una mejor siembra en un área determinada.	Dado que el robot aún es una simulación, puede haber inconveniente al momento de la ejecución.
Implementación de un prototipo de robot sembrador de papa en terrenos sin inclinación para pequeños productores	2019	Riobamba, Ecuador	Arduino IDE	El prototipo cuenta con un sistema de envío y recepción de datos en tiempo real.	El robot, por ser demasiado robusto, presenta problemas de movilidad en terrenos irregulares.

Fuente: Elaboración propia (2021)

Los proyectos anteriores tienen como objetivo común mejorar el proceso de siembra, ya sea de semillas o para la reforestación, y utilizaron la tecnología para optimizar procesos y reducir costos. La programación del robot se realizó de forma mecánica utilizando el programa Arduino y ninguno optó por programar al robot con inteligencia artificial. Uno de los desafíos en los proyectos fue el desplazamiento del robot al realizar la siembra en terrenos irregulares; para esto utilizaron diferentes métodos de movilidad y potencia, como sistema de tracción de oruga o robot tetrápodo, entre otros.

Los resultados obtenidos fueron satisfactorios, sin dejar de lado la interacción con los humanos. Una de las ventajas que posee la presente propuesta es que el dominio está programado con inteligencia artificial en lenguaje PDDL, permitiendo representar un problema de siembra de semillas y reforestación de árboles.

VII. Dominio y problema PDDL

Dominio PDDL

A continuación se presenta el dominio propuesto, el cual fue creado, desarrollado e implementado en lenguaje PDDL (archivo domain.pddl):

```
(define (domain treeworld)
  (:requirements :strips)
  (:predicates (tree ?tree)
               (robot-garra ?robot))
```

```
(location ?loc)
(loc-inicio ?ini)
(in-earth ?tree ?loc)
(in-forest ?loc ?forest)
(at ?robot ?ini)
(clear ?loc)
(forest ?forest)
(arm-empty)
(holding ?tree))

(:action pickup
  :parameters (?tree ?robot ?ini)
  :precondition (and (at ?tree ?ini)(arm-empty)
                    (at ?robot ?ini)(tree ?tree)
                    (robot-garra ?robot)(loc-inicio ?ini)
                    :effect (and (holding ?tree)(not (at ?tree ?ini))
                                (not (arm-empty))))))

(:action go
  :parameters (?tree ?robot ?loc ?ini ?forest)
  :precondition (and (holding ?tree)(at ?robot ?ini)(tree ?tree)
                    (robot-garra ?robot)(loc-inicio ?ini)(location ?loc)(forest ?forest)
                    (in-forest ?loc ?forest))
  :effect (and (at ?robot ?loc)(not(at ?robot ?ini))))))

(:action back
  :parameters(?tree ?robot ?loc ?ini ?forest)
  :precondition(and(arm-empty)(at ?robot ?loc)(tree ?tree)
                (robot-garra ?robot)(loc-inicio ?ini)(location ?loc)(forest ?forest)(in-earth ?tree ?loc))
  :effect(and(not(at ?robot ?loc))(at ?robot ?ini))))

(:action plant
  :parameters(?tree ?robot ?loc ?forest)
```



```
:precondition(and(at ?robot ?loc)(clear ?loc)
(tree ?tree)(forest ?forest)
(in-forest ?loc ?forest)(robot-garra ?robot)
(location ?loc)(holding ?tree))
:effect(and(not(holding ?tree))(in-earth ?tree
?loc)(arm-empty)(not(clear ?loc))))
```

Dominio: lista de predicados

- **(tree ?tree):** Es necesario un árbol (?tree) para poder plantarlo.
- **(robot-garra ?robot):** Es necesario un robot (?robot) para poder desplazar y sembrar el árbol.
- **(location ?loc):** Para poder sembrar un árbol necesitamos una coordenada (?loc).
- **(loc-inicio ?ini):** Es necesario una ubicación inicial (?ini) donde estarán los árboles y el robot al inicio del problema.
- **(in-earth ?tree ?loc):** Establece que ?tree (árbol) está sembrado en ?loc (coordenada).
- **(in-forest ?loc ?forest):** Establece que ?loc (coordenada) está en ?forest (bosque).
- **(at ?robot ?ini):** Establece que ?robot (robot) está en ?ini (coordenada inicial).
- **(clear ?loc):** Establece que ?loc (coordenada) está vacía.
- **(forest ?forest):** Para poder reforestar necesitamos un bosque (?forest).
- **(arm-empty):** El brazo está vacío.
- **(holding ?tree):** Al árbol (?tree) lo están sosteniendo.

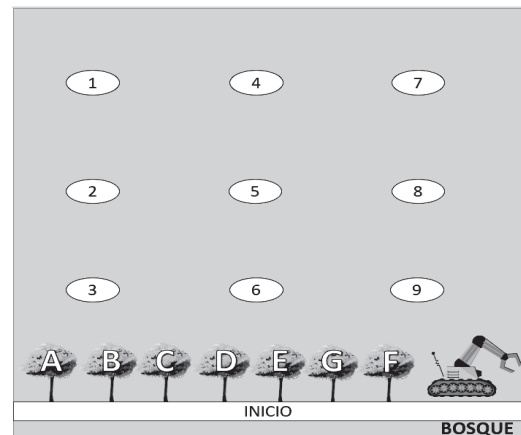
Dominio: lista de acciones

- **pickup:** Esta acción es utilizada para recoger árboles.
- **go:** Esta acción se utiliza cuando el robot se desplaza del inicio a una coordenada.
- **back:** Esta acción se utiliza cuando el robot se desplaza de una coordenada al inicio.
- **plant:** Esta acción se utiliza para plantar un árbol en una coordenada.

Descripción del problema

Para este problema se ha definido un ejercicio supuesto con el que se realizarán las pruebas del dominio planteado anteriormente. Cabe aclarar que este ejercicio se asemeja a un problema real que puede presentarse en el contexto de siembra de semillas y reforestación de árboles. El ejercicio supuesto es el siguiente: El plan de desarrollo destinado para el año 2022 en el territorio antioqueño tiene como finalidad desplegar un proyecto de reforestación en Caucasia. El proyecto consiste en la siembra de árboles de Guayacán en 2 hectáreas del territorio. Para su ejecución se cuenta con nueve coordenadas preestablecidas. La función del robot propuesto es plantar siete árboles distribuidos entre los distintos puntos del bosque, como se ve en las Figuras 2 y 3.

Figura 2. Estado inicial del problema.



Fuente: Elaboración propia (2021)

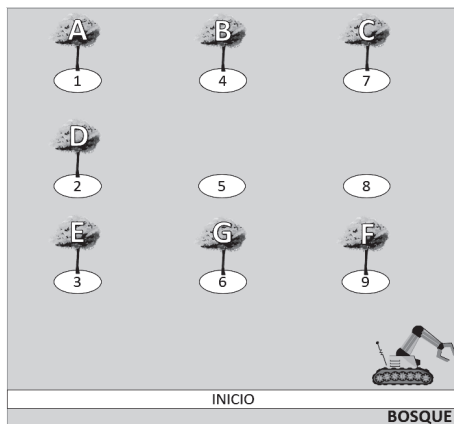
En la Figura 2 el robot y los siete árboles parten desde un punto "INICIO" y estos deben plantarse, según corresponda, en las coordenadas establecidas. Este proceso puede verse en la Tabla 2.

Tabla 2. Correspondencia de árbol, punto y coordenadas del estado final del problema.

Árbol	Punto	Coordenada
A	1	(A-1)
B	4	(B-4)
C	7	(C-7)
D	2	(D-2)
E	3	(E-3)
F	9	(F-9)
G	6	(G-6)

Fuente: Elaboración propia (2021)

Figura 3. Estado final del problema.



Fuente: Elaboración propia (2021)

La Figura 3 muestra el estado final del problema, indicando cómo quedarían plantados los siete árboles según el punto establecido. También muestra la ubicación final del robot, después de realizar las acciones necesarias para ubicar los árboles.

Tabla 3. Descripción de elementos de las Figuras 2 y 3.

Imagen	Descripción
	Bosque
	Punto
	Árbol
	Robot
INICIO	Punto de partida de los árboles y el robot

Fuente: Elaboración propia (2021)

El problema planteado anteriormente pudo modelarse en lenguaje PDDL, con sus respectivos estado inicial y estado final, según fue establecido. A continuación se detalla el archivo (problem.pddl):

Problema PDDL

```
(define (problem one)
  (:domain treeworld)
  (:objects treeA treeB treeC treeD treeE treeF
    treeG
    forest1 robot1 inicio forest1-1 forest1-2
    forest1-3 forest1-4
    forest1-5 forest1-6 forest1-7 forest1-8
    forest1-9
  )
  (:init (tree treeA)
    (tree treeB)
    (tree treeC)
    (tree treeD)
```

```
(tree treeE)
(tree treeF)
(tree treeG)
(forest forest1)
(location forest1-1)
(location forest1-2)
(location forest1-3)
(location forest1-4)
(location forest1-5)
(location forest1-6)
(location forest1-7)
(location forest1-8)
(location forest1-9)
(loc-inicio inicio)
(robot-garra robot1)
(at treeA inicio)
(at treeB inicio)
(at treeC inicio)
(at treeD inicio)
(at treeE inicio)
(at treeF inicio)
(at treeG inicio)
(at robot1 inicio)
(clear forest1-1)
(clear forest1-2)
(clear forest1-3)
(clear forest1-4)
(clear forest1-5)
(clear forest1-6)
(clear forest1-7)
(clear forest1-8)
(clear forest1-9)
(in-forest forest1-1 forest1)
(in-forest forest1-2 forest1)
(in-forest forest1-3 forest1)
(in-forest forest1-4 forest1)
(in-forest forest1-5 forest1)
(in-forest forest1-6 forest1)
(in-forest forest1-7 forest1)
(in-forest forest1-8 forest1)
(in-forest forest1-9 forest1)
(arm-empty)
)
```

```
(:goal (and(in-earth treeA forest1-1)
(in-earth treeB forest1-4)
(in-earth treeC forest1-7)
(in-earth treeD forest1-2)
(in-earth treeE forest1-3)
(in-earth treeF forest1-9)
(in-earth treeG forest1-6)
(at robot1 inicio))))
```

Problema: lista de objetos:

- Árboles A-G:

```
treeA
treeB
treeC
treeD
treeE
treeF
treeG
```

Bosque:

```
forest1
Robot:
robot1
```

- Posición de inicio:
inicio
- Coordenadas en el bosque 1-9:
forest1-1
forest1-2
forest1-3
forest1-4
forest1-5
forest1-6
forest1-7
forest1-8
forest1-9

VIII. Resultados

El resultado obtenido por el planificador en la herramienta web PDDL Editor fue un plan total de 28 acciones, mostradas en la Figura 4. El robot

realizó cuatro acciones por cada árbol a sembrar; por consiguiente, el robot finalizó su cometido de plantar siete árboles mediante 28 acciones o conjunto de estados. En la Tabla 4 se presenta una muestra detallada del plan obtenido en PDDL Editor.

Figura 4. Resultado del plan obtenido en PDDL.

```

pickup treeG robot1 inicio
  go treeG robot1 forest1-6 inicio forest1
    plant treeG robot1 forest1-6 forest1
      back treeG robot1 forest1-6 inicio forest1
        pickup treeF robot1 inicio
          go treeF robot1 forest1-9 inicio forest1
            plant treeF robot1 forest1-9 forest1
              back treeF robot1 forest1-9 inicio forest1
                pickup treeE robot1 inicio
                  go treeE robot1 forest1-3 inicio forest1
                    plant treeE robot1 forest1-3 forest1
                      back treeE robot1 forest1-3 inicio forest1
                        pickup treeD robot1 inicio
                          go treeD robot1 forest1-2 inicio forest1
                            plant treeD robot1 forest1-2 forest1
                              back treeD robot1 forest1-2 inicio forest1
                                pickup treeC robot1 inicio
                                  go treeC robot1 forest1-7 inicio forest1
                                    plant treeC robot1 forest1-7 forest1
                                      back treeC robot1 forest1-7 inicio forest1
                                        pickup treeB robot1 inicio
                                          go treeB robot1 forest1-4 inicio forest1
                                            plant treeB robot1 forest1-4 forest1
                                              back treeB robot1 forest1-4 inicio forest1
                                                pickup treeA robot1 inicio
                                                  go treeA robot1 forest1-1 inicio forest1
                                                    plant treeA robot1 forest1-1 forest1
                                                      back treeA robot1 forest1-1 inicio forest1

```

Fuente: Visual Studio Code (Extensión PDDL v2.22.6, 2021)

Tabla 4. Muestra detallada de cuatro estados del plan obtenido en PDDL Editor.

1- (pickup treeg robot1 inicio)	2- (go treeg robot1 forest1-6 inicio forest1)
<pre>(:action pickup :parameters (treeg robot1 inicio) :precondition (and (at treeg inicio) (arm-empty) (at robot1 inicio) (tree treeg) (robot-garra robot1) (loc-inicio inicio)) :effect (and (holding treeg) (not (at treeg inicio)) (not (arm-empty))))</pre>	<pre>(:action go :parameters (treeg robot1 forest1-6 inicio forest1) :precondition (and (holding treeg) (at robot1 inicio) (tree treeg) (robot-garra robot1) (loc-inicio inicio) (location forest1-6) (forest forest1) (in-forest forest1-6 forest1)) :effect (and (at robot1 forest1-6) (not (at robot1 inicio))))</pre>

3- (plant treeg robot1 forest1-6 forest1)

```
(:action plant
  :parameters (treeg robot1 forest1-6 forest1)
  :precondition
    (and
      (at robot1 forest1-6)
      (clear forest1-6)
      (tree treeg)
      (forest forest1)
      (in-forest forest1-6 forest1)
      (robot-garra robot1)
      (location forest1-6)
      (holding treeg)
    )
  :effect
    (and
      (not
        (holding treeg)
      )
      (in-earth treeg forest1-6)
      (arm-empty)
      (not
        (clear forest1-6)
      )
    )
)
```

4- (back treeg robot1 forest1-6 inicio forest1)

```
(:action back
  :parameters (treeg robot1 forest1-6 inicio forest1)
  :precondition
    (and
      (arm-empty)
      (at robot1 forest1-6)
      (tree treeg)
      (robot-garra robot1)
      (loc-inicio inicio)
      (location forest1-6)
      (forest forest1)
      (in-earth treeg forest1-6)
    )
  :effect
    (and
      (not
        (at robot1 forest1-6)
      )
      (at robot1 inicio)
    )
)
```

Para mayor profundización sobre el plan obtenido, véase el apartado Anexos.

IX. Conclusiones y trabajo futuro

A partir de la investigación realizada, puede observarse que hemos definido un dominio en lenguaje PDDL que puede utilizarse de manera simple para representar un problema de un robot reforestador basado en planificación automática, en el contexto de siembra y reforestación. Para la definición del dominio se establecieron 11 predicados y 4 acciones. A partir de lo anterior, pudimos crear, diseñar e implementar en PDDL un problema supuesto de reforestación de árboles, enmarcado en el plan de desarrollo destinado para el año 2022 en el territorio antioqueño, que tiene como finalidad desplegar un proyecto de reforestación en Cauca. Logramos probar nuestro dominio con el problema de siembra de siete árboles de Guayacán en nueve puntos distribuidos en siete coordenadas. El robot inteligente, para asegurar la siembra de cada árbol, requirió las cuatro acciones propuestas en el dominio, en el siguiente orden: pickup, go, plant, back. De esta manera, se obtuvo un plan de 28 acciones que resuelven el problema de forma satisfactoria, puesto que se consiguió llegar al estado final con la menor cantidad de acciones posibles. Cabe resaltar que modelar un dominio en PDDL es una tarea compleja, sobre todo cuando existen detalles propios del dominio en cuestión, así que contar con personas expertas en el área a modelar facilitará la definición del dominio y el problema en lenguaje PDDL.

Por otro lado, se siguieron las cuatro fases del proceso de investigación, teniendo como resultado la selección del área temática, el planteamiento del problema, delimitación de la investigación, marco teórico, diseño e implementación del dominio y problema en lenguaje PDDL, análisis y, por último, síntesis de información.

En el futuro, definiremos un problema (caso de estudio) junto con Corantioquia y lo implementaremos en lenguaje PDDL, para observar el funcionamiento de nuestro dominio en el tratamiento de dicho problema y, así, contribuir al desarrollo sostenible dentro del Plan de Acción 2020-2023 "Protegiendo ecosistemas naturales", que tiene como objetivo la implementación

de acciones de restauración, rehabilitación y reforestación de áreas protegidas. Otras posibles líneas de investigación incluyen el uso de diferentes planificadores para estudiar el funcionamiento de nuestro dominio, así como incluir funciones de coste que complementen al dominio y permitan tener una métrica relacionada con reducir el coste del plan obtenido.

X. Referencias

- [1] Ministerio de Ambiente y Desarrollo Sostenible e Instituto de Hidrología, Meteorología y Estudios Ambientales, *Resultados de monitoreo de reforestación 2019*. 9 de julio de 2020 [En línea]. Disponible en: <http://www.ideam.gov.co/documents/10182/105413996/presentacionbalancedeforestacion2019/7c9323fc-d0a1-4c95-b1a1-1892b162c067#:~:text=Para%20el%20año%202019%20la,2018%20cuando%20fue%20de%2012%25.&text=En%20la%20Orinoquía%20se%20consolidan,el%20curso%20del%20río%20Vichada>
- [2] A. R. López Guevara y D. C. López Guevara, "Desarrollo de una herramienta de educación ambiental enfocado en técnicas básicas de reforestación en Colombia", Trabajo de grado. Bogotá, D.C., 2021. Repositorio Institucional UCC [En línea]. Disponible en: <https://repository.ucc.edu.co/handle/20.500.12494/34984>
- [3] M. I. Gómez López y L. Calle Páez, "Colombia y sus bosques frente a la acción climática", Tesis de especialización. Bogotá, D.C., 2018. Repositorio Institucional Universidad Militar Nueva Granada [En línea]. Disponible en: <http://hdl.handle.net/10654/17936>
- [4] "Colombia: el camino a la reforestación de los 180 millones de árboles antes de 2022", *CONtexto Ganadero*, 9 de marzo de 2020 [En línea]. Disponible en: <https://www.contextoganadero.com/politica/colombia-el-camino-la-reforestacion-de-los-180-millones-de-arboles-antes-de-2022>
- [5] A. A. Sosa Navarro y M. A. Puellas Palacios, "Diseño de un robot móvil de dos hileras para trasplante de plántulas de arroz en sembríos de la costa norte peruana", Trabajo de grado. Lima, 2020. Repositorio Institucional Universidad Ricardo Palma [En línea]. Disponible en: <https://hdl.handle.net/20.500.14138/3580>

- [6] J. C. Flórez González y V. M. Castellanos Pabón, "Diseño en Autocad de robot sembrador de semillas de fresa: Fresabot III", Tesis de especialización. Bogotá D.C., 2019. Repositorio Institucional Universidad Distrital Francisco José de Caldas [En línea]. Disponible en: <http://hdl.handle.net/11349/16257>
- [7] R. D. Díaz Araque, "Diseño y construcción de un prototipo de robot tetrápodo sembrador de semillas para leguminosas", Trabajo de grado. Sangolquí, 2018. Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE [En línea]. Disponible en: <http://repositorio.espe.edu.ec/handle/21000/15131>
- [8] A. A. Bukovits y S. D. Núñez Romagnoli, "Simulación de aplicación de un robot forestador", Trabajo de grado. Argentina, 2018. Repositorio Institucional UADE [En línea]. Disponible en: <https://repositorio.uade.edu.ar/xmlui/handle/123456789/8137>
- [9] D. J. Duarte Gordillo y A. D. Rincón Rodríguez, "Ingeniería de detalle para un robot sembrador de semillas de maíz", Trabajo de grado. Bogotá, D.C., 2019. Repositorio Institucional Universidad Distrital Francisco José de Caldas [En línea]. Disponible en: <http://hdl.handle.net/11349/23182>
- [10] H. F. González Gavilanes y M. G. Carrillo Trujillo, "Implementación de un prototipo de robot sembrador de papa en terrenos sin inclinación para pequeños productores", Trabajo de grado. Riobamba, 2019. DBRAI Escuela Superior Politécnica de Chimborazo [En línea]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/13686>
- [11] A. Marrella, "Automated Planning for Business Process Management", *Journal on Data Semantics*, vol. 8, no. 2, pp. 79-98, noviembre de 2019 [En línea]. Disponible en: <https://doi.org/10.1007/s13740-018-0096-0>
- [12] C. A. Cocosco, "A review of STRIPS: A new approach to the application of theorem proving to problem solving by R.E. Fikes, N.J. Nilsson, 1971", *Artificial Intelligence*, no. 4, pp. 304-526B, 1998.
- [13] J. Ortiz Pérez-Jaraiz, "Planificación en sistemas robotizados mediante PDDL y ROS", Trabajo de grado. 2021. Biblioteca Digital Universidad de Alcalá [En línea]. Disponible en: <https://ebuah.uah.es/dspace/handle/10017/49634>
- [14] J. L. Usero Vílchez, "PLAGUE: Un editor visual para PDDL", Trabajo de grado. Málaga, 2015. Repositorio Institucional Universidad de Málaga [En línea]. Disponible en: <http://hdl.handle.net/10630/11652>
- [15] S. J. Russell y P. Norvig, *Inteligencia artificial. Un enfoque moderno*, 2ª ed. Madrid: Pearson Educación, 2004.
- [16] M. Gelfond y Y. Kahl, *Knowledge Representation, Reasoning, and the Design of Intelligent Agents. The Answer-Set Programming Approach*. New York: Cambridge University Press, 2014 [En línea]. Disponible en: <https://doi.org/10.1017/CBO9781139342124>
- [17] F. Caro Herranz, "Sistemas de construcción de mapas en PDDL para la planificación automática", Trabajo de grado. Madrid, 2016. Biblioteca Universidad Carlos III de Madrid [En línea]. Disponible en: <http://hdl.handle.net/10016/23310>
- [18] Organización de las Naciones Unidas para la Alimentación y la Agricultura, *Global Forest Resources Assessment 2000*. Roma: FAO, 2001 [En línea]. Disponible en: <https://www.fao.org/forestry/fra/86624/en/>
- [19] J. P. Aragón Rodríguez, "El deber que tiene el Estado Colombiano de realizar y fomentar acciones de reforestación, para cumplir con la conservación y protección del medioambiente", Trabajo de grado. Bogotá, 2019. Repositorio Institucional Universidad Católica de Colombia – RIUCaC [En línea]. Disponible en: <https://hdl.handle.net/10983/23456>
- [20] C. Sabino, *El proceso de investigación*. Buenos Aires: Lumen, 1996.

ANEXOS

PLAN COMPLETO EN LENGUAJE PDDL (HERRAMIENTA: PDDL EDITOR)

1- (pickup treeg robot1 inicio)

```
(:action pickup
:parameters (treeg robot1 inicio)
:precondition
(and
(at treeg inicio)
(arm-empty)
(at robot1 inicio)
(tree treeg)
(robot-garra robot1)
(loc-inicio inicio)
)
:effect
(and
(holding treeg)
(not
(at treeg inicio)
)
(not
(arm-empty)
)
)
)
```

2- (go treeg robot1 forest1-6 inicio forest1)

```
(:action go
:parameters (treeg robot1 forest1-6 inicio forest1)
:precondition
(and
(holding treeg)
(at robot1 inicio)
(tree treeg)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-6)
(forest forest1)
(in-forest forest1-6 forest1)
)
:effect
(and
(at robot1 forest1-6)
(not
(at robot1 inicio)
)
)
)
```

3- (plant treeg robot1 forest1-6 forest1)

```
(:action plant
:parameters (treeg robot1 forest1-6 forest1)
:precondition
(and
(at robot1 forest1-6)
(clear forest1-6)
(tree treeg)
(forest forest1)
(in-forest forest1-6 forest1)
(robot-garra robot1)
(location forest1-6)
(holding treeg)
)
:effect
(and
(not
(holding treeg)
)
(in-earth treeg forest1-6)
(arm-empty)
(not
(clear forest1-6)
)
)
)
```

4- (back treeg robot1 forest1-6 inicio forest1)

```
(:action back
:parameters (treeg robot1 forest1-6 inicio forest1)
:precondition
(and
(arm-empty)
(at robot1 forest1-6)
(tree treeg)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-6)
(forest forest1)
(in-earth treeg forest1-6)
)
:effect
(and
(not
(at robot1 forest1-6)
)
(at robot1 inicio)
)
)
```

5- (pickup treef robot1 inicio)

```
(:action pickup
:parameters (treef robot1 inicio)
:precondition
(and
(at treef inicio)
(arm-empty)
(at robot1 inicio)
(tree treef)
(robot-garra robot1)
(loc-inicio inicio)
)
:effect
(and
(holding treef)
(not
(at treef inicio)
)
(not
(arm-empty)
)
)
)
)
```

6- (go treef robot1 forest1-9 inicio forest1)

```
(:action go
:parameters (treef robot1 forest1-9 inicio forest1)
:precondition
(and
(holding treef)
(at robot1 inicio)
(tree treef)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-9)
(forest forest1)
(in-forest forest1-9 forest1)
)
:effect
(and
(at robot1 forest1-9)
(not
(at robot1 inicio)
)
)
)
```

7- (plant treef robot1 forest1-9 forest1)

```
(:action plant
:parameters (treef robot1 forest1-9 forest1)
:precondition
(and
(at robot1 forest1-9)
(clear forest1-9)
(tree treef)
(forest forest1)
(in-forest forest1-9 forest1)
(robot-garra robot1)
(location forest1-9)
(holding treef)
)
:effect
(and
(not
(holding treef)
)
(in-earth treef forest1-9)
(arm-empty)
(not
(clear forest1-9)
)
)
)
```

8- (back treef robot1 forest1-9 inicio forest1)

```
(:action back
:parameters (treef robot1 forest1-9 inicio forest1)
:precondition
(and
(arm-empty)
(at robot1 forest1-9)
(tree treef)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-9)
(forest forest1)
(in-earth treef forest1-9)
)
:effect
(and
(not
(at robot1 forest1-9)
)
(at robot1 inicio)
)
)
```

9- (pickup treee robot1 inicio)

```
(:action pickup
:parameters (treee robot1 inicio)
:precondition
(and
(at treee inicio)
(arm-empty)
(at robot1 inicio)
(tree treee)
(robot-garra robot1)
(loc-inicio inicio)
)
:effect
(and
(holding treee)
(not
(at treee inicio)
)
(not
(arm-empty)
)
)
)
)
```

10- (go treee robot1 forest1-3 inicio forest1)

```
(:action go
:parameters (treee robot1 forest1-3 inicio forest1)
:precondition
(and
(holding treee)
(at robot1 inicio)
(tree treee)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-3)
(forest forest1)
(in-forest forest1-3 forest1)
)
:effect
(and
(at robot1 forest1-3)
(not
(at robot1 inicio)
)
)
)
```

11- (plant treee robot1 forest1-3 forest1)

```
(:action plant
:parameters (treee robot1 forest1-3 forest1)
:precondition
(and
(at robot1 forest1-3)
(clear forest1-3)
(tree treee)
(forest forest1)
(in-forest forest1-3 forest1)
(robot-garra robot1)
(location forest1-3)
(holding treee)
)
:effect
(and
(not
(holding treee)
)
(in-earth treee forest1-3)
(arm-empty)
(not
(clear forest1-3)
)
)
)
```

12- (back treee robot1 forest1-3 inicio forest1)

```
(:action back
:parameters (treee robot1 forest1-3 inicio forest1)
:precondition
(and
(arm-empty)
(at robot1 forest1-3)
(tree treee)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-3)
(forest forest1)
(in-earth treee forest1-3)
)
:effect
(and
(not
(at robot1 forest1-3)
)
(at robot1 inicio)
)
)
```

13- (pickup treed robot1 inicio)

```
(:action pickup
:parameters (treed robot1 inicio)
:precondition
(and
(at treed inicio)
(arm-empty)
(at robot1 inicio)
(tree treed)
(robot-garra robot1)
(loc-inicio inicio)
)
:effect
(and
(holding treed)
(not
(at treed inicio)
)
(not
(arm-empty)
)
)
)
)
```

14- (go treed robot1 forest1-2 inicio forest1)

```
(:action go
:parameters (treed robot1 forest1-2 inicio forest1)
:precondition
(and
(holding treed)
(at robot1 inicio)
(tree treed)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-2)
(forest forest1)
(in-forest forest1-2 forest1)
)
:effect
(and
(at robot1 forest1-2)
(not
(at robot1 inicio)
)
)
)
```

15- (plant treed robot1 forest1-2 forest1)

```
(:action plant
:parameters (treed robot1 forest1-2 forest1)
:precondition
(and
(at robot1 forest1-2)
(clear forest1-2)
(tree treed)
(forest forest1)
(in-forest forest1-2 forest1)
(robot-garra robot1)
(location forest1-2)
(holding treed)
)
:effect
(and
(not
(holding treed)
)
(in-earth treed forest1-2)
(arm-empty)
(not
(clear forest1-2)
)
)
)
```

16- (back treed robot1 forest1-2 inicio forest1)

```
(:action back
:parameters (treed robot1 forest1-2 inicio forest1)
:precondition
(and
(arm-empty)
(at robot1 forest1-2)
(tree treed)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-2)
(forest forest1)
(in-earth treed forest1-2)
)
:effect
(and
(not
(at robot1 forest1-2)
)
(at robot1 inicio)
)
)
```

17- (pickup treec robot1 inicio)

```
(:action pickup
:parameters (treec robot1 inicio)
:precondition
(and
(at treec inicio)
(arm-empty)
(at robot1 inicio)
(tree treec)
(robot-garra robot1)
(loc-inicio inicio)
)
:effect
(and
(holding treec)
(not
(at treec inicio)
)
(not
(arm-empty)
)
)
)
)
```

18- (go treec robot1 forest1-7 inicio forest1)

```
(:action go
:parameters (treec robot1 forest1-7 inicio forest1)
:precondition
(and
(holding treec)
(at robot1 inicio)
(tree treec)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-7)
(forest forest1)
(in-forest forest1-7 forest1)
)
:effect
(and
(at robot1 forest1-7)
(not
(at robot1 inicio)
)
)
)
```

19- (plant treec robot1 forest1-7 forest1)

```
(:action plant
:parameters (treec robot1 forest1-7 forest1)
:precondition
(and
(at robot1 forest1-7)
(clear forest1-7)
(tree treec)
(forest forest1)
(in-forest forest1-7 forest1)
(robot-garra robot1)
(location forest1-7)
(holding treec)
)
:effect
(and
(not
(holding treec)
)
(in-earth treec forest1-7)
(arm-empty)
(not
(clear forest1-7)
)
)
)
```

20- (back treec robot1 forest1-7 inicio forest1)

```
(:action back
:parameters (treec robot1 forest1-7 inicio forest1)
:precondition
(and
(arm-empty)
(at robot1 forest1-7)
(tree treec)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-7)
(forest forest1)
(in-earth treec forest1-7)
)
:effect
(and
(not
(at robot1 forest1-7)
)
(at robot1 inicio)
)
)
```

21- (pickup treeb robot1 inicio)

```
(:action pickup
:parameters (treeb robot1 inicio)
:precondition
(and
(at treeb inicio)
(arm-empty)
(at robot1 inicio)
(tree treeb)
(robot-garra robot1)
(loc-inicio inicio)
)
:effect
(and
(holding treeb)
(not
(at treeb inicio)
)
(not
(arm-empty)
)
)
)
)
```

22- (go treeb robot1 forest1-4 inicio forest1)

```
(:action go
:parameters (treeb robot1 forest1-4 inicio forest1)
:precondition
(and
(holding treeb)
(at robot1 inicio)
(tree treeb)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-4)
(forest forest1)
(in-forest forest1-4 forest1)
)
:effect
(and
(at robot1 forest1-4)
(not
(at robot1 inicio)
)
)
)
```

23- (plant treeb robot1 forest1-4 forest1)

```
(:action plant
:parameters (treeb robot1 forest1-4 forest1)
:precondition
(and
(at robot1 forest1-4)
(clear forest1-4)
(tree treeb)
(forest forest1)
(in-forest forest1-4 forest1)
(robot-garra robot1)
(location forest1-4)
(holding treeb)
)
:effect
(and
(not
(holding treeb)
)
(in-earth treeb forest1-4)
(arm-empty)
(not
(clear forest1-4)
)
)
)
```

24- (back treeb robot1 forest1-4 inicio forest1)

```
(:action back
:parameters (treeb robot1 forest1-4 inicio forest1)
:precondition
(and
(arm-empty)
(at robot1 forest1-4)
(tree treeb)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-4)
(forest forest1)
(in-earth treeb forest1-4)
)
:effect
(and
(not
(at robot1 forest1-4)
)
(at robot1 inicio)
)
)
```

25- (pickup treea robot1 inicio)

```
(:action pickup
:parameters (treea robot1 inicio)
:precondition
(and
(at treea inicio)
(arm-empty)
(at robot1 inicio)
(tree treea)
(robot-garra robot1)
(loc-inicio inicio)
)
:effect
(and
(holding treea)
(not
(at treea inicio)
)
(not
(arm-empty)
)
)
)
)
```

26- (go treea robot1 forest1-1 inicio forest1)

```
(:action go
:parameters (treea robot1 forest1-1 inicio forest1)
:precondition
(and
(holding treea)
(at robot1 inicio)
(tree treea)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-1)
(forest forest1)
(in-forest forest1-1 forest1)
)
:effect
(and
(at robot1 forest1-1)
(not
(at robot1 inicio)
)
)
)
)
```

27- (plant treea robot1 forest1-1 forest1)

```
(:action plant
:parameters (treea robot1 forest1-1 forest1)
:precondition
(and
(at robot1 forest1-1)
(clear forest1-1)
(tree treea)
(forest forest1)
(in-forest forest1-1 forest1)
(robot-garra robot1)
(location forest1-1)
(holding treea)
)
:effect
(and
(not
(holding treea)
)
(in-earth treea forest1-1)
(arm-empty)
(not
(clear forest1-1)
)
)
)
)
```

28- (back treea robot1 forest1-1 inicio forest1)

```
(:action back
:parameters (treea robot1 forest1-1 inicio forest1)
:precondition
(and
(arm-empty)
(at robot1 forest1-1)
(tree treea)
(robot-garra robot1)
(loc-inicio inicio)
(location forest1-1)
(forest forest1)
(in-earth treea forest1-1)
)
:effect
(and
(not
(at robot1 forest1-1)
)
(at robot1 inicio)
)
)
)
```