

Examen Final de GRAU-IA

(14 de enero de 2021)

Duración: 2 horas 30 minutos

1. (5 puntos) Uno de los objetivos del telescopio espacial James Webb que se acaba de lanzar con éxito es el análisis de señales en el infrarrojo cercano y el infrarrojo medio para descubrir planetas extrasolares y analizar sus atmósferas. Se espera que, combinando los datos de este telescopio con los de otros observatorios (tanto terrestres como orbitales), se pueda estimar la habitabilidad de los planetas descubiertos. Por todo ello una organización que prefiere mantenerse anónima nos ha encargado un SBC que permita escoger los planetas con mayor posibilidad de albergar a la raza humana en caso necesario de tener que abandonar la Tierra.

Los datos de los que disponemos incluyen información sobre el planeta candidato, el resto de planetas y la estrella de su sistema planetario, y otros astros (estrellas, planetas o supernovas) y sistemas candidatos que pueda haber en la cercanía (unos 100 parsecs de distancia).

Cada sistema planetario tiene asociada su distancia a la Tierra (en años-luz) y está compuesto por una estrella y varios planetas (se descartan sistemas de más de una estrella, suelen ser inhabitables). De cada estrella guardamos su radio (medido en soles: por debajo de 0.4 soles es una estrella enana, de 0.4 a 20 soles es una estrella normal, por encima de 20 soles es una estrella gigante). Los planetas pueden ser rocosos o gigantes gaseosos. De los planetas rocosos se tiene su tamaño relativo a la Tierra (medido en tierras) y la distancia máxima y mínima de la órbita del planeta a su estrella (medida en unidades astronómicas -UA-). Del planeta candidato (siempre rocoso) se guarda otra información adicional (como el porcentaje de oxígeno y de hidrocarburos en la atmósfera), tal y como veremos a continuación. De los gigantes gaseosos se tiene su tamaño relativo a Júpiter (medido en júpiteres) y la distancia máxima y mínima de la órbita del planeta a su estrella (medida en unidades astronómicas -UA-).

También tenemos información de las supernovas cercanas a un sistema planetario (a máximo 10 parsecs de distancia), de ellas guardamos su tipo (Ia, Ib, Ic, II-b, II-L, II-P, II-n) y su distancia (en parsecs) al sistema planetario. Tenemos información de las estrellas gigantes cercanas (a máximo 10 parsecs de distancia) ya que son estrellas con alta probabilidad de convertirse en supernovas en el futuro, de ellas guardamos su masa (medida en masas solares) y su distancia (en parsecs) al sistema planetario candidato. Finalmente tenemos información de otros sistemas planetarios en la cercanía (a máximo 100 pársecs) que puedan ser candidatos para ser habitables, de ellos guardamos la misma información que para el sistema planetario candidato.

De toda esa cantidad de datos los expertos nos indican que hay información que es especialmente relevante a la hora de evaluar un planeta candidato:

- El tamaño del planeta candidato relativo al tamaño de la Tierra (entre 0.8 y 0.9 tierras es un planeta ligero, hasta 1.25 tierras es normal, entre 1.25 y 1.5 es pesado)
- La distancia máxima y mínima de la órbita del planeta candidato a su estrella (entre 0.7 y 0.8 UA es considerado calurosa, hasta 1.3 UA es considerado normal y entre de 1.3 y 1.5 es considerado fría)
- Porcentaje de oxígeno en la atmósfera del planeta candidato (un valor entre el 12 % y el 15 % se considera aceptable, un porcentaje superior al 25 % se considera una atmósfera reactiva, en otro caso es respirable)
- El porcentaje de hidrocarburos en la atmósfera del planeta candidato (un porcentaje por debajo del 1 % es una atmósfera ligera, hasta el 2 % se considera normal y hasta el 3 % es una atmósfera densa)
- El radio de la estrella del sistema planetario candidato relativo al sol (entre 0.7 y 0.8 soles es una estrella fría, hasta 1.2 soles es una estrella normal, hasta 1.5 soles es una estrella caliente)
- Número de gigantes gaseosos en el sistema (el rango normal es entre 4 y 7 gigantes gaseosos, por debajo es bajo, por encima es alto)
- Tamaño del mayor gigante gaseoso relativo a Júpiter (por debajo de 2 júpiteres es ideal, hasta 5 júpiteres es bueno y por encima molesto)
- La distancia mínima entre las órbitas de los planetas por encima y debajo del candidato (para el planeta en la órbita superior, una distancia por debajo de las 2 UA se considera malo, hasta las 3 UA se considera aceptable y por encima ideal; para el planeta en la órbita inferior una distancia por debajo de 0.3 UA se considera malo, hasta las 0.5 UA se considera aceptable y por encima es ideal)
- Cantidad de supernovas en un radio de 10 parsecs (cero o una es segura, hasta dos es peligrosa, más es muy peligrosa)
- Cantidad de estrellas gigantes en un radio de 10 parsecs (cero o una es segura, hasta dos es peligrosa, más es muy peligrosa)
- Cercanía a otros sistemas planetarios candidatos (si hay alguno más cerca de 20 parsecs es ideal, hasta 40 parsecs aceptable, mas lejos es aislado)

Nuestro análisis nos ha llevado a determinar que existen tres criterios a partir de los cuales podemos dar un diagnóstico:

- Hospitalidad planetaria, que nos califica lo hospitalario que es el planeta candidato, valorado en inhabitable, muy dura, dura, soportable e ideal. Está asociada al tamaño del planeta, a la órbita, al porcentaje de oxígeno y al de hidrocarburos. Por ejemplo, si el tamaño relativo del planeta candidato es normal, la atmósfera es aceptable en oxígeno y ligera en hidrocarburos, y la distancia a la estrella es normal entonces la hospitalidad planetaria será ideal, pero si la atmósfera es reactiva en oxígeno o densa en hidrocarburos aunque el tamaño relativo y la distancia a la estrella sean normales entonces la hospitalidad planetaria será muy dura.
- Hospitalidad del sistema, que nos califica lo hospitalario que es el sistema en el que está el planeta, valorado en hostil, inestable, soportable y segura. Está asociada al radio de la estrella del sistema planetario, al número de gigantes gaseosos, el tamaño del mayor gigante gaseoso y las distancias a las órbitas de los planetas por encima y por debajo del candidato. Por ejemplo, si el radio de la estrella es normal, el número de gigantes gaseosos es bajo y la distancia mínima entre las órbitas de los planetas por encima y debajo del candidato son ideales entonces la hospitalidad del sistema será segura, pero si el número de gigantes gaseosos es alto o el mayor gigante gaseoso del sistema es molesto entonces la hospitalidad del sistema se considera hostil.
- Hospitalidad de la región, que nos califica lo hospitalaria que es la región que rodea al sistema planetario, valorada en peligrosa, aceptable y segura. Está asociada a la cantidad de supernovas, cantidad de estrellas gigantes y la cercanía de otros sistemas candidatos. Por ejemplo, si la cantidad de supernovas cercanas es segura y la cantidad de estrellas gigantes cercanas es segura o peligrosa y la cercanía de otros sistemas es aislado entonces la hospitalidad de la región es aceptable, pero si la cantidad de supernovas cercanas es peligrosa o muy peligrosa o bien la cantidad de estrellas gigantes cercanas es muy peligrosa entonces la hospitalidad de la región se considera peligrosa.

A partir de estas características queremos determinar la supervivencia de la población que se traslade a ese planeta en una escala de enteros del 0 al 10, donde 0 significa misión suicida y 10 un viaje al paraíso.

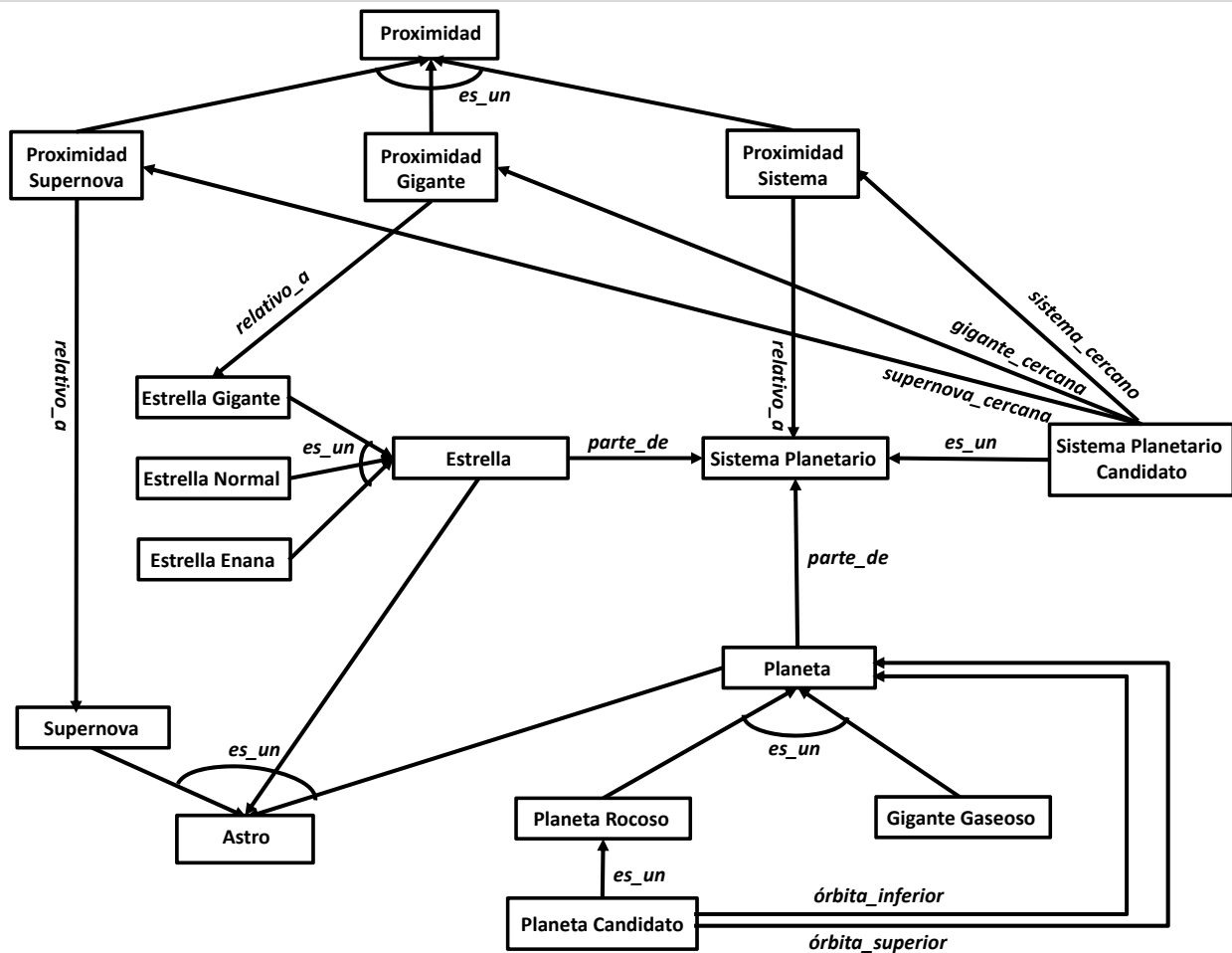
La *experiencia* en el dominio nos dice que:

- La hospitalidad de la región tiene un peso moderado en el éxito, e incluso una zona segura no aportaría más de tres puntos de la valoración. Lo que si sabemos es que una zona peligrosa no permite dar una valoración total superior a cinco.
- La hospitalidad del sistema tiene un peso superior y aportaría hasta cinco puntos de la valoración. Sabemos que una hospitalidad segura permite una valoración total de al menos cinco si la región es por lo menos aceptable.
- La hospitalidad planetaria aportaría hasta seis puntos de la valoración. Lo que sabemos es que un planeta inhabitable no permite una valoración total mayor que uno. Una valoración ideal aportaría el máximo de valoración posible para este criterio, si la hospitalidad del sistema es al menos soportable y la de la región es al menos aceptable.

El valor final de la supervivencia se calcula a partir de las puntuaciones obtenidas para los tres criterios, combinándolas según las restricciones que imponen para obtener un valor en la escala entera de 0 a 10.

- a) (2,5 puntos) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. Justifica las decisiones de diseño que has realizado. En el caso de las subclases de una clase, justifica con un texto breve la necesidad de crear dichas subclases. Para cada concepto de la ontología lista todos sus atributos (para cada atributo se ha de indicar su nombre y su tipo, y en el caso de las enumeraciones, sus valores posibles). [Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente.]

En este problema la ontología ha de incorporar, como mínimo, todos los conceptos que forman parte de la entrada del problema: el sistema planetario con su estrella y sus planetas (incluido el candidato), y la información de los otros astros y sistemas planetarios cercanos en la región del espacio alrededor del planeta. En este caso la salida esta compuesta por los conceptos **Planeta Candidato** y **Sistema Planetario Candidato**.



De la ontología resultante cabe remarcar que se ha decidido no crear un concepto **Región** ya que tenía poca información asociada y no permitía modelar claramente la proximidad entre cuerpos celestes. Para para modelar la cercanía de otros astros y sistemas planetarios al **Sistema Planetario Candidato** no es suficiente crear una relación **cerca_de**, ya que esa relación no permite asociar una distancia a si misma, y poner la distancia al sistema candidato como un atributo más en las clases **Supernova**, **Estrella Gigante** y **Sistema Planetario** tampoco funcionaría en el caso de que estén cerca de más de un sistema candidato (hacer que sea un atributo multivaluado tampoco funcionaría, ya que no sabríamos cada distancia a que sistema candidato se refiere). Es por ello que se ha añadido el concepto **Proximidad** que tiene la distancia en parsecs como atributo, y del que se crean tres subclases para que puedan referenciar supernovas, estrellas gigantes o sistemas planetarios. En el caso de **Planeta** se han definido los tipos como subclases ya que tiene atributos con unidades diferentes (tierras y júpiteres) y además **Planeta Rocos** tiene una relación extra (la subclase **Planeta Candidato**, que debe su existencia a tener asociados muchos más atributos que el resto de planetas). En el caso de **Estrella** se han definido los tipos como subclases ya que, aunque tienen los mismo atributos, en el caso de **Estrella Gigante** tiene una relación extra (la proximidad a un sistema candidato) que no tienen las subclases **Estrella Normal** y **Estrella Enana**. También cabe destacar que se ha decidido no clasificar las **Supernova** como una subclase de **Estrella** ya que el enunciado solo considera tres tipos de estrella, pero conceptualmente se podría considerar una supernova como una estrella moribunda o un resto de estrella y se podría añadir como subclase. En el caso de **Supernova** se han definido los tipos como un atributo y no como subclases porque en el modelo actual ninguna de ellas tiene ni atributos especiales ni relaciones especiales con otros conceptos. La clase **Astro** se ha añadido porque el enunciado dice explícitamente que hay tres tipos de astros y porque nos permite relacionar mejor la clase **Supernova** con otras clases, no le hemos puesto atributos porque el enunciado no menciona ninguno pero si quisiéramos hacer esa ontología un poco más reusable se le podría asociar el nombre y número de catálogo.

Es importante remarcar que se ha decidido usar la relación taxonómica **parte_de** para relacionar **Sistema Planetario** con **Estrella** y **Planeta**, ya que son precisamente los elementos que componen un sistema planetario, son sus partes fundamentales, sin ellas no podemos hablar de un sistema planetario.

Atributos: a continuación se listan los atributos mínimos para representar la información mencionada

explícitamente en el enunciado y la necesaria para el apartado b del problema. Hay otros atributos que se deberían añadir (como el nombre de cada estrella y planeta) si quisiéramos una ontología lo más general y reutilizable posible.

- **Sistema Planetario:** distancia_Tierra (real positivo, en añosluz)
- **Sistema Planetario Candidato:** *cant_gigantes_gaseosos* (enumeración: {bajo, normal, alto}), *mayor_gigante_gaseoso* (enumeración: {ideal, bueno, molesto}), *cant_supernovas* (enumeración: {seguro, peligroso, muy peligroso}), *cant_estrellas_gigantes* (enumeración: {seguro, peligroso, muy peligroso}), *cercania_sist_planetarios* (enumeración: {ideal, aceptable, aislado}), *Hospitalidad_Sistema* (enumeración: {hostil, inestable, soportable, segura}), *Hospitalidad_Región* (enumeración: {peligrosa, aceptable, segura})
- **Planeta:** distancia_máxima_a_estrella (real positivo, en UA), distancia_mínima_a_estrella (real positivo, en UA)
- **Planeta Rocos:** tamaño (real positivo, en tierras)
- **Gigante Gaseoso:** tamaño (real positivo, en júpiteres)
- **Planeta Candidato:** %_Oxigeno (real positivo), %_Hidrocarburos (real positivo), *tamaño_relativo* (enumeración: {ligero, normal, pesado}), *distancia_a_estrella* (enumeración: {calurosa, normal, pesado}), *atmósfera_O* (enumeración: {respirable, aceptable, reactiva}), *atmósfera_HC* (enumeración: {ligera, normal, densa}), *dist_órbita_inferior* (enumeración: {malo, aceptable, ideal}), *dist_órbita_superior* (enumeración: {malo, aceptable, ideal}), *Hospitalidad_Planetaria* (enumeración: {inhabitable, muy dura, dura, soportable, ideal}), *SUPERVIVENCIA* (entero positivo, rango de 0 a 10)
- **Estrella:** radio (real positivo, en soles), *radio_relativo* (enumeración: {fria, normal, caliente})
- **Estrella Gigante:** masa (real positivo, en masas solares)
- **Supernova :** tipo (enumeración: {Ia, Ib, Ic, II-b, II-L, II-P, II-n})
- **Proximidad:** distancia (real positivo, en parsec)

Como se puede ver, hemos decidido también representar las características interpretadas y abstractas como atributos (son los que tienen el nombre en *cursiva*, *Cursiva* o *CURSIVA*). De esta manera todos los conceptos que aparecerán en las reglas están soportados por la ontología. Dependiendo de la solución propuesta en el apartado b) -con tres fases o con cuatro fases- estas características corresponderán a diferentes fases, como veremos en el siguiente apartado. En el caso de haber escogido tres fases, las características en *cursiva* y minúsculas corresponden a los problemas concretos y también deberían de ser representados en la ontología como parte de los datos de entrada.

- b) (2,5 puntos) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística. Justifica si el tipo de solución que se pide requiere de refinamiento o no. Lista las variables que forman parte del Problema Abstracto (nombre y valores posibles). Lista las variables que forman parte de la Solución Abstracta (nombre y valores posibles). Da al menos 3 ejemplos de reglas (suficientemente variadas, utilizando diferentes conceptos) para cada una de las fases de esta metodología, usando los conceptos de la ontología desarrollada en el apartado anterior. Para las reglas podéis usar la notación de alto nivel que usamos en los ejercicios de problemas (si ... entonces ...) o una notación más próxima a la del lenguaje CLIPS (siempre que queden claro los elementos del antecedente de la regla y del consecuente).

Para resolverlo mediante clasificación heurística debemos identificar en el problema las diferentes fases y elementos de esta metodología. En este caso había dos posibles interpretaciones:

- considerar que *Hospitalidad_Planetaria*, *Hospitalidad_Sistema* y *Hospitalidad_Región* son características abstractas de la solución, y que *SUPERVIVENCIA* es una solución concreta → en este caso propondremos una solución con 4 fases, donde todos los atributos en *cursiva* y minúsculas corresponden a las características abstractas del problema que se obtienen por abstracción de los datos concretos (los atributos que no están en *cursiva* en la ontología). Es un planteamiento correcto si se tiene cuidado en aplicar metódicamente las 4 fases y no usar datos concretos en el proceso de asociación heurística, pero es discutible que sea el mejor para este problema, ya que no hace falta acceder a datos del problema concreto para seleccionar el valor de la supervivencia, ni hacer cálculos matemáticos complicados.
- considerar que *Hospitalidad_Planetaria*, *Hospitalidad_Sistema* y *Hospitalidad_Región* son características abstractas del problema, y que *SUPERVIVENCIA* es una solución abstracta → podemos ver la solución como una enumeración de valores discretos {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10} de los que se ha de escoger uno, y para ello no hace falta hacer un refinamiento para acceder a datos del problema concreto. En este caso propondremos una solución con 3 fases, donde todos los atributos en *cursiva* y minúsculas corresponden a datos del problema concreto que se obtienen por un paso de interpretación de los datos básicos previo al proceso de abstracción. Es un planteamiento mejor, el que se usa en la solución que se presentará a continuación.

El primer elemento son los problemas concretos que hay que tratar. En este dominio los problemas concretos están definidos por las características que se mencionan en el enunciado: el tamaño del planeta, la distancia de la órbita, los porcentajes de oxígeno e hidrocarburos, etc.

Para facilitar el paso de abstracción hacia los problemas abstractos, haremos primero un proceso de interpretación de los datos de bajo nivel ($\%_{\text{oxígeno}}$) en datos cualitativos (*atmósfera_O*) aprovechando la información de los expertos:

- si Estrella.radio $\in [0.7, 0.8]$ entonces Estrella.radio_relativo=fria;
- si PlanetaCandidato. $\%_{\text{oxígeno}} > 25\%$ entonces PlanetaCandidato.atmósfera_O=reactiva;
- si PlanetaCandidato.tamaño $\in [0.8, 0.9]$ entonces PlanetaCandidato.tamaño_relativo=ligero;
- si (PlanetaCandidato.distancia_mínima_a_estrella $\in [0.8, 1.3]$ y PlanetaCandidato.distancia_máxima_a_estrella $\in [1.3, 1.5]$) entonces PlanetaCandidato.distancia_a_estrella=normal;
- si contar(SistemaPlanetarioCandidato.supernova_cercana.distancia ≤ 10 parsec) < 2 entonces SistemaPlanetarioCandidato.num_supernovas=seguro;

El segundo elemento son los problemas abstractos, estos estarán definidos a partir de las tres características que menciona el enunciado (hospitalidad planetaria, del sistema y de la región), y los valores que se asignen a cada característica. Para conectar los problemas concretos con los abstractos necesitamos definir las reglas de abstracción de datos, por ejemplo:

- si PlanetaCandidato.tamaño_relativo=normal y PlanetaCandidato.distancia_a_estrella=normal y (PlanetaCandidato.atmósfera_O=soportable o PlanetaCandidato.atmósfera_HC=ligera) entonces PlanetaCandidato.Hospitalidad_Planetaria=soportable;
- si PlanetaCandidato.tamaño_relativo=normal y PlanetaCandidato.distancia_a_estrella=normal y PlanetaCandidato.atmósfera_O=reactiva o PlanetaCandidato.atmósfera_HC=densa) entonces PlanetaCandidato.Hospitalidad_Planetaria=muy dura;
- si PlanetaCandidato.tamaño_relativo=normal y PlanetaCandidato.distancia_a_estrella=calurosa y PlanetaCandidato.atmósfera_O=reactiva y PlanetaCandidato.atmósfera_HC=densa entonces PlanetaCandidato.Hospitalidad_Planetaria=inhabitable;
- si PlanetaCandidato.tamaño_relativo=pesado y PlanetaCandidato.distancia_a_estrella=frió y (PlanetaCandidato.atmósfera_O=reactiva o PlanetaCandidato.atmósfera_HC=densa) entonces PlanetaCandidato.Hospitalidad_Planetaria=inhabitable;
- si Estrella.radio_relativo=normal y SistemaPlanetarioCandidato.num_gigantes_gaseosos=bajo y PlanetaCandidato.dist_orbita_superior=ideal y PlanetaCandidato.dist_orbita_inferior=ideal entonces SistemaPlanetarioCandidato.Hospitalidad_Sistema=segura
- si SistemaPlanetarioCandidato.num_gigantes_gaseosos=alto y SistemaPlanetarioCandidato.mayor_gigante_gaseoso=molesto entonces SistemaPlanetarioCandidato.Hospitalidad_Sistema=hostil
- si SistemaPlanetarioCandidato.cant_supernovas = seguro y SistemaPlanetarioCandidato.cant_estrellas_gigantes=(seguro o peligroso) y SistemaPlanetarioCandidato.cercanía_sist_planetarios = aislado entonces SistemaPlanetarioCandidato.Hospitalidad_Región = aceptable;
- si SistemaPlanetarioCandidato.cant_supernovas = peligroso o SistemaPlanetarioCandidato.cant_estrellas_gigantes=peligroso entonces SistemaPlanetarioCandidato.Hospitalidad_Región = peligrosa;

El tercer elemento son las soluciones abstractas. En este caso las soluciones son directamente soluciones concretas, un valor entre 0 y 10. Tal como explica el enunciado, cada característica abstracta se puede usar para calcular una parte de la solución, aunque este calculo no es independiente, ya que algunas se influyen entre ellas y en el cálculo total.

Para ligar los problemas abstractos con las soluciones abstractas necesitaremos reglas de asociación heurística, como por ejemplo:

- si PlanetaCandidato.Hospitalidad_Planetaria = inhabitable y SistemaPlanetarioCandidato.Hospitalidad_Sistema=inestable y SistemaPlanetarioCandidato.Hospitalidad_Región=peligrosa entonces PlanetaCandidato.SUPERVIVENCIA=0;
- si PlanetaCandidato.Hospitalidad_Planetaria = inhabitable y SistemaPlanetarioCandidato.Hospitalidad_Sistema=segura y SistemaPlanetarioCandidato.Hospitalidad_Región=segura entonces PlanetaCandidato.SUPERVIVENCIA=1;
- si PlanetaCandidato.Hospitalidad_Planetaria = muy dura y SistemaPlanetarioCandidato.Hospitalidad_Sistema=segura y SistemaPlanetarioCandidato.Hospitalidad_Región=aceptable entonces PlanetaCandidato.SUPERVIVENCIA=5;

- si PlanetaCandidato.Hospitalidad_Planetaria = ideal
y SistemaPlanetarioCandidato.Hospitalidad_Sistema=segura
y SistemaPlanetarioCandidato.Hospitalidad_Región=peligrosa
entonces PlanetaCandidato.SUPERVIVENCIA=5;
- si PlanetaCandidato.Hospitalidad_Planetaria = ideal
y SistemaPlanetarioCandidato.Hospitalidad_Sistema=segura
y SistemaPlanetarioCandidato.Hospitalidad_Región=segura
entonces PlanetaCandidato.SUPERVIVENCIA=10;

2. (5 puntos) Una empresa de logística tiene que gestionar la entrega de paquetes voluminosos desde sus almacenes a una serie de domicilios en diversas localidades de una región. Para ello, disponen de un conjunto de vehículos de diversa capacidad, que pueden desplazarse libremente dentro de una localidad, o a través del conjunto de carreteras.

La empresa busca automatizar la organización de sus almacenes y vehículos para la entrega de todos los paquetes de una jornada. Previo al reparto local, la empresa puede enviar cualquier paquete desde su centralita a cada uno de sus almacenes mediante trenes de carga u otros medios, de forma que podemos escoger en qué almacén está cada paquete al inicio de la jornada. A su vez asumimos que, al inicio de la jornada, podemos tener cada uno de los vehículos en uno de nuestros almacenes a nuestra elección.

Una vez los paquetes se encuentran en un almacén en una ciudad, un vehículo puede pasar a recogerlos si tiene espacio. Se asume que todos los paquetes ocupan lo mismo, y que nuestros vehículos tienen capacidad para 2 (furgoneta) o 4 (camión) paquetes. De la misma forma, un vehículo puede entregar un paquete si se encuentra en la misma localidad donde debe entregarse. Tanto la acción de recoger como la acción de entregar toman el mismo tiempo. Asumimos que no tendremos que entregar más de un paquete a una misma dirección. También asumimos que el tiempo gastado en cargar un paquete es el mismo estemos ya en el almacén o teniendo que desplazarnos a él desde un sitio en la misma localidad, ya que con lo voluminosos que son los paquetes, la carga del paquete cuesta mucho más que un desplazamiento intra-localidad.

Los vehículos pueden moverse libremente entre los domicilios de una misma localidad. Sin embargo, para el desplazamiento entre dos localidades, deberá usar una serie de carreteras para desplazarse. Conocemos el mapa de carreteras de la región y la conectividad entre ciudades. Como dificultad añadida, las carreteras son de diferente longitud, y no es lo mismo recorrer una carretera larga que una corta. Para ello, un conjunto de expertos evalúa la longitud de cada carretera, y las ha dividido en 'tramos', de forma que el tiempo que se tarda en entregar un paquete en un domicilio, recogerlo de un almacén, y entre tramos (o entre un tramo y una localidad) es el mismo.

- a) (3,5 puntos) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a un número arbitrario de paquetes, vehículos, localidades y carreteras.

Existen muchas posibles formas de modelar este dominio en PDDL (con más o menos predicados, con más o menos tipos, con más o menos operadores...) y por ello tomaremos decisiones que vayan encaminadas a crear un modelo que no contenga ineficiencias innecesarias.

Algunas cosas que tendremos en cuenta en la solución propuesta:

- intentaremos minimizar el número de operadores y el factor de ramificación de los mismos, de forma que se reduzca la exploración de que operadores son aplicables en cada momento;
- evitaremos operadores con parámetros similares, de forma que la existencia de unos objetos u otros dirija la instanciación de operadores;
- usaremos tipos en las variables para reducir en lo posible la cantidad de objetos que el planificador comprobará para cada parámetro del operador;
- intentaremos evitar el uso de exists y forall en las precondiciones y efectos de los operadores, ya que tienen un impacto negativo en el tiempo de cómputo y, en la práctica, aumentan el factor de ramificación por la existencia de variables ocultas (variables a instanciar que no son parámetros) dentro del operador;
- será más importante que la ejecución sea eficiente, aunque la representación sea más compleja (es decir, añadiremos predicados, operadores y tipos si eso puede facilitar la labor del planificador).
- nos aseguraremos de que los elementos que se nos pide que registremos de manera explícita sean parte de la solución del problema, preferentemente como instanciaciones de los operadores que forman parte del plan resultado.

A priori este problema se puede atacar de varias maneras, todas válidas siempre y cuando tengamos la expresividad que se pide en el enunciado. Tal como se comenta en el enunciado, no necesitamos manejar

viajes intra-ciudad explícitamente. Deberíamos, sin embargo, explicitar tanto el acto de recoger y entregar paquetes, así como el desplazamiento entre localidades y tramos. También, dado que el programa debería permitir escoger dónde empieza cada vehículo y paquete, sería buena idea explicitar los desplazamientos desde centralita a cada lugar (aunque dada la ambigüedad del enunciado esto se ha penalizado poco). En la solución propuesta realizamos las siguientes simplificaciones, que hacen que la representación sea muy eficiente:

- Ignoramos la posición de un vehículo dentro de una localidad. El coste de cargar o descargar es el mismo independientemente de la localización del vehículo, no hace falta tomar nota de la posición exacta de cada vehículo.
- Agrupamos localidades y secciones como 'posiciones' donde puede haber un vehículo.
- Agrupamos vehículo y paquete como 'cosas que salen de centralita' para simplificar la acción de llevarlo a un almacén.
- Ignoramos que existen 'almacenes' como algo explícito, ya que solo necesitamos saber si hay un almacén en una localidad para llevar vehículos o paquetes a esta desde centralita. Solo modelaremos si una localidad tiene almacén.
- Podríamos ignorar que existen direcciones explícitas, y hacer que el predicado de 'entregar en' junte paquetes y localidades. En la solución propuesta no se ha hecho, pero también sería válido. Una posible justificación de tener domicilios es que las entregas a domicilios aparecen explícitamente dentro del enunciado.
- Para modelar conectividad, podríamos usar predicados duplicados (*accessible_from ?a ?b* y *accessible_from ?b ?a*) o manejarlo en la precondition con un or.Cuál es preferible depende del grafo de conexiones del problema concreto en sí, ambas son igual de válidas.

```
(define (domain BigDeliveries)
  (:requirements :strips :adl :typing)
  (:types position deployable address slot - object
    locality section - position ; sitios donde puede haber un vehiculo
                                ; (aparte de centralita)
    vehicle package - deployable ; Cosas que salen de centralita,
                                ; usado para simplificar 'deploy'
  )

  (:predicates
    (at_central_storage ?d - deployable) ; Centralita, pre-deploy
    (at ?d - deployable ?p - position)
    (warehouse_at ?l - locality)

    (slot_of ?s - slot ?v - vehicle)
    (empty ?s - slot)
    (stored_at ?p - package ?s - slot)

    (address_at ?a - address ?l - locality)
    (deliver_at ?p - package ?a - address)
    (served ?p - package)

    (accessible_from ?p1 - position ?p2 - position)
  )

  (:action deploy
    :parameters (?d - deployable ?l - locality)
    :precondition (and (at_central_storage ?d)
                       (warehouse_at ?l))
    :effect (and
      (not (at_central_storage ?d))
      (at ?d ?l))
  )

  (:action load
    :parameters (?p - package ?s - slot ?v - vehicle ?l - locality)
    :precondition (and (at ?p ?l) (at ?v ?l)
                       (slot_of ?s ?v) (empty ?s) )
    :effect (and
```

```

(not (empty ?s)) (not (at ?p ?l))
(stored_at ?p ?s))
)

(:action deliver
:parameters (?p - package ?s - slot ?v - vehicle ?l - locality ?a - address)
:precondition (and (at ?v ?l) (address_at ?a ?l)
                  (slot_of ?s ?v) (stored_at ?p ?s)
                  (deliver_at ?p ?a))
)
:effect (and
        (empty ?s) (not (stored_at ?p ?s))
        (served ?p))
)

(:action move
:parameters (?v - vehicle ?orig - position ?dest -position)
:precondition (and (at ?v ?orig) (accessible ?orig ?dest))
:effect (and
        (not (at ?v ?orig)) (at ?v ?dest))
)
)

```

- b) (1,5 puntos) La empresa nos proporciona un ejemplo de prueba para saber como es la entrada del sistema. Describe el problema usando PDDL. Da una breve explicación de cómo modelas el problema. No es necesario escribir el init completo siempre que sea obvio qué falta, dando al menos un ejemplo de uso de cada predicado.

Paquete	Domicilio, localidad
P1	Av. Diagonal 3, Barcelona
P2	Av. Paral·lel 26, Barcelona
P3	C/ Gavà 8, Barcelona
P4	P. Olot 4, Girona
P5	C/ Joan Alsina 7, Girona
P6	C/ Torelló 2, Vic
P7	Av. Can Sucarrats, Rubí

Vehículo	Tipo
C1	Camión
F1	Furgoneta

Almacén	Localidad
A1	Barcelona
A2	Girona

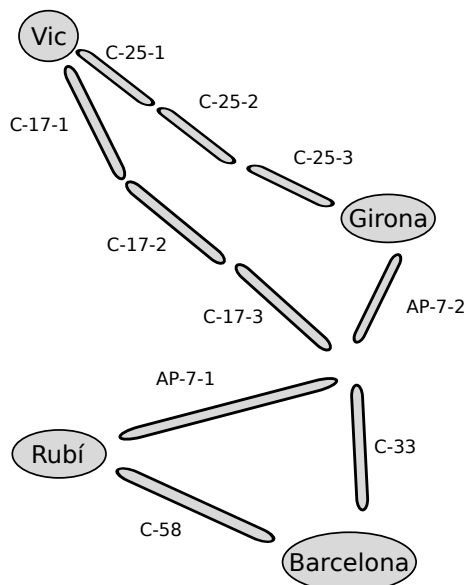


Figura 1: Mapa de carreteras de la región, dividido en tramos

```

(define (problem test-suite1)
(:domain BigDeliveries)
(:objects
BCN VIC GIR RUB - locality
C58 AP71 AP72 C33 C171 C172 C173 C251 C252 C253 - section
C1 F1- vehicle

```



```

P1 P2 P3 P4 P5 P6 P7 - package
Diag3 Par26 Gava8 Olot4 Jals7 Tor2 Sucar - address
C11 C12 C13 C14 F11 F12 - slot
)
(:init
(at_central_storage C1) (at_central_storage F1)
(at_central_storage P1) (at_central_storage P2) ... (at_central_storage P7)
(warehouse_at BCN) (warehouse_at GIR)
(slot_of C11 C1) ... (slot_of C14 C1) (slot_of F11 F1) (slot_of F12 F1)
(empty C11) ... (empty F12)
(address_at Diag3 BCN) (address_at Par26 BCN) ... (address_at Sucar RUB)
  (deliver_at P1 Diag3) (deliver_at P2 Par26) ... (deliver_at P7 Sucar)
  (accessible_from BCN C58) (accessible_from C58 BCN)
  (accessible_from BCN C33) (accessible_from C33 BCN)
  (accessible_from AP71 AP72) (accessible_from AP72 AP71)
  ...
)
(:goal (forall (?x - package) (served ?x))
)

```

Las notas se publicaran el día **26 de enero**.