

Planificación Automática – Práctica Final

Configurador de orígenes

Álvaro Guerrero Espinosa (100472294)
José Antonio Verde Jiménez (100472221)

Universidad Carlos III de Madrid

12 de enero de 2026

Índice

1. Dominio

2. Problema

3. Pruebas y resultados

4. Conclusiones

1.1 Descripción

- **Dominio**: configurador de orbamentos.
- Cada personaje tiene un **orbamento**.
- Cada orbamento tiene **6 ranuras**.
- Las ranuras están conectadas por **líneas**.
- En ranuras se pueden insertar **cuarzos**.
- Hay **restricciones** entre cuarzos.
- Hay **7 elementos**: **tierra** , **agua** , **fuego** , **viento** , **tiempo** , **espacio**  y **espejismo** .
- Cada **cuarzo** da un poder elemental (\mathbb{N}) para cada uno de los elementos.



Figura 1: Orbamento vacío

1.1 Descripción

Artes utilizables	Coste de PE
Lanza de tierra	30
Impacto pétreo	120
Protección de tierra	60
Emblema	30
Bola ígnea	10
Flecha ígnea	60
Fortaleza	30
Alma nublada	20
Acelerador	10

Valor	1	2	3	4	5	6	7
Línea 1	4	0	2	0	0	2	2
Línea 2	2	0	2	0	1	2	2

Figura 2: Ejemplo de artes que se pueden activar

- Una **arte** se activa en un **orbamento** si existe alguna **línea** para la que todos los **elementos** tienen valor suficiente.
- Cada **cuarzo** pertenece a una **categoría**.
 - Dos cuarzos de la misma categoría no pueden estar en el mismo orbamento.
 - Hay **ranuras** que solo permiten **cuarzos** de cierto **elemento**.
- El dominio es igual en los cinco primeros juegos de la saga (pequeñas variaciones).

1.2 Ejemplo

- Configuración
 - Escudo de topacio (+2 , +2 ).
 - Precisión 2 (+2 .
 - Rotura 2 (+2 .
 - Ataque 2 (+2 .
 - Hoja petrificadora (+2 .
 - PE 2 (+1 , +2 , +2 ).
- Total
 - Línea 1: 4 , 2 , 2 , 2 
 - Línea 2: 2 , 2 , 1 , 2 , 2 



Figura 3: Ejemplo de configuración.

1.2 Ejemplo

Se activan las artes:

- Martillo pétreo (≥ 1 
- Lanza de tierra (≥ 2 
- Impacto pétreo (≥ 4 
- Protección de tierra (≥ 2  $\wedge \geq 2$ 
- Emblema (≥ 1  $\wedge \geq 1$ 
- Bola ígnea (≥ 1 
- Flecha ígnea (≥ 2 
- Fortaleza (≥ 1  $\wedge \geq 1$ 
- Alma nublada (≥ 1 
- Acelerador (≥ 1  $\wedge \geq 1$ 

No se activaría por ejemplo:

- Pared de tierra (≥ 4  $\wedge \geq 3$ )
Porque falta uno de .
- Las líneas son independientes.
- Pueden compartir ranuras.
- La activación de artes es independiente.

1.3 Complicaciones y soluciones

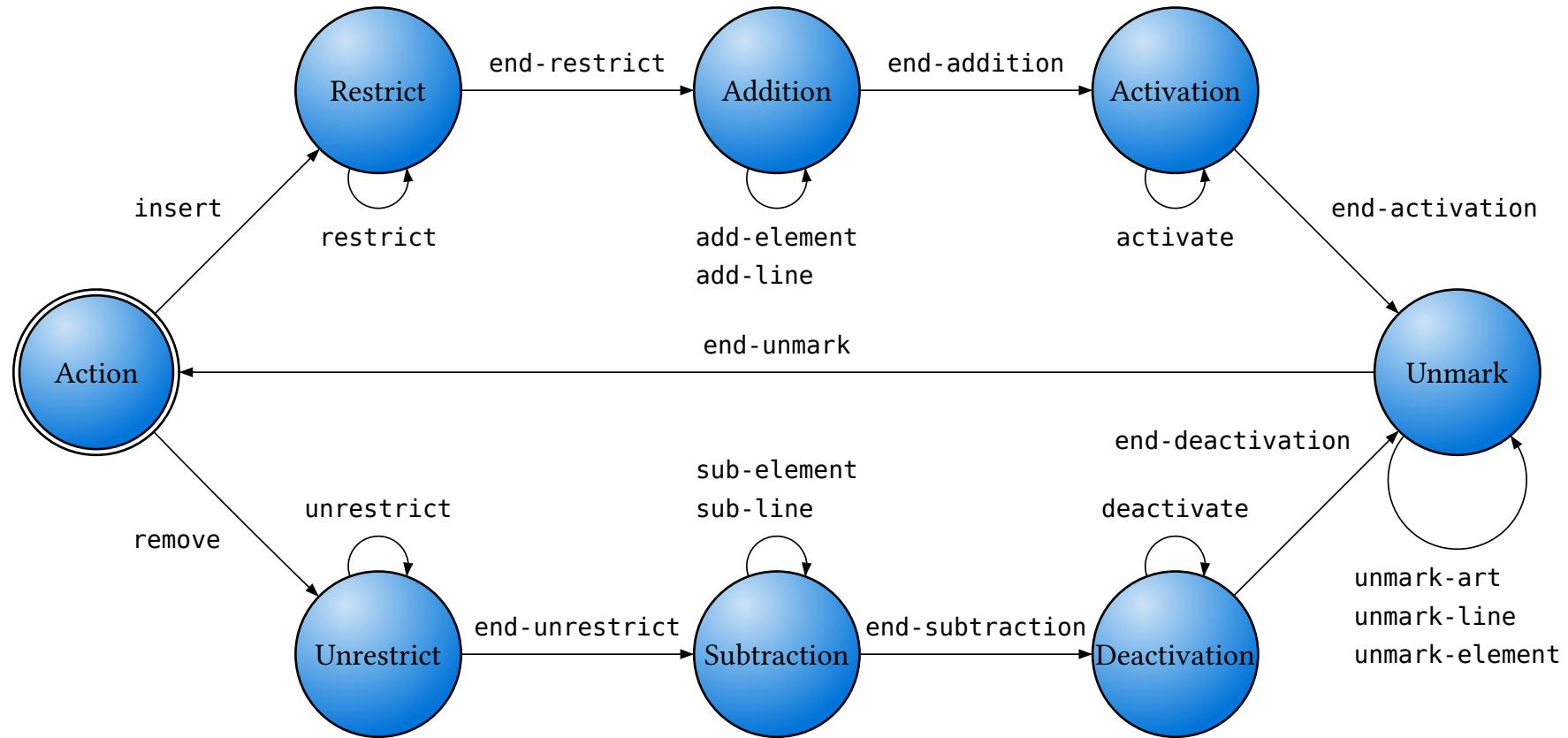
- Utilizamos **fast-downward**
 - Muchos competidores de *I.P.C. 2023 Classical Tracks* parten de o usan **fast-downward**.
 - **No soporta fluenta** (excepto **total-cost**).
 - El dominio usa mucho sumas y restas.
- No soporta cuantificadores universales en los efectos.
 - Insertar en una ranura puede modificar varias líneas.
 - Una inserción o eliminación puede activar y desactivar muchas aristas.

1.3 Complicaciones y soluciones

Definimos el tipo natural (\mathbb{N}).

- Los objetos n_0, n_1, n_2, \dots
- Relaciones entre ellos:
 - `(addition ?lhs ?rhs ?r - natural)`
 - `(addition n2 n3 n5)`
 - `(addition n0 n3 n3)`
 - Se puede usar para sumar y restar.
 - `(less-than ?lhs ?rhs)`
 - `(less-than n0 n1)`
 - `(less-than n3 n7)`
 - Con `not`, `and`, y cambiando el orden se puede definir el resto de operaciones: $\leq, <, =, >, \geq, \neq$.
- Necesitamos tres parámetros por operación.
- ¡Si queremos hacer sumas para los 7 elementos, necesitaríamos mínimo 21 parámetros!
- Si se tienen que instanciar todas las acciones, ignorando combinaciones imposibles, para 23 naturales habría $23^{21} \approx 3.95 \cdot 10^{28}$ objetos (95 bits).
- Las operaciones se tienen que hacer por pasos.

1.3 Complicaciones y soluciones

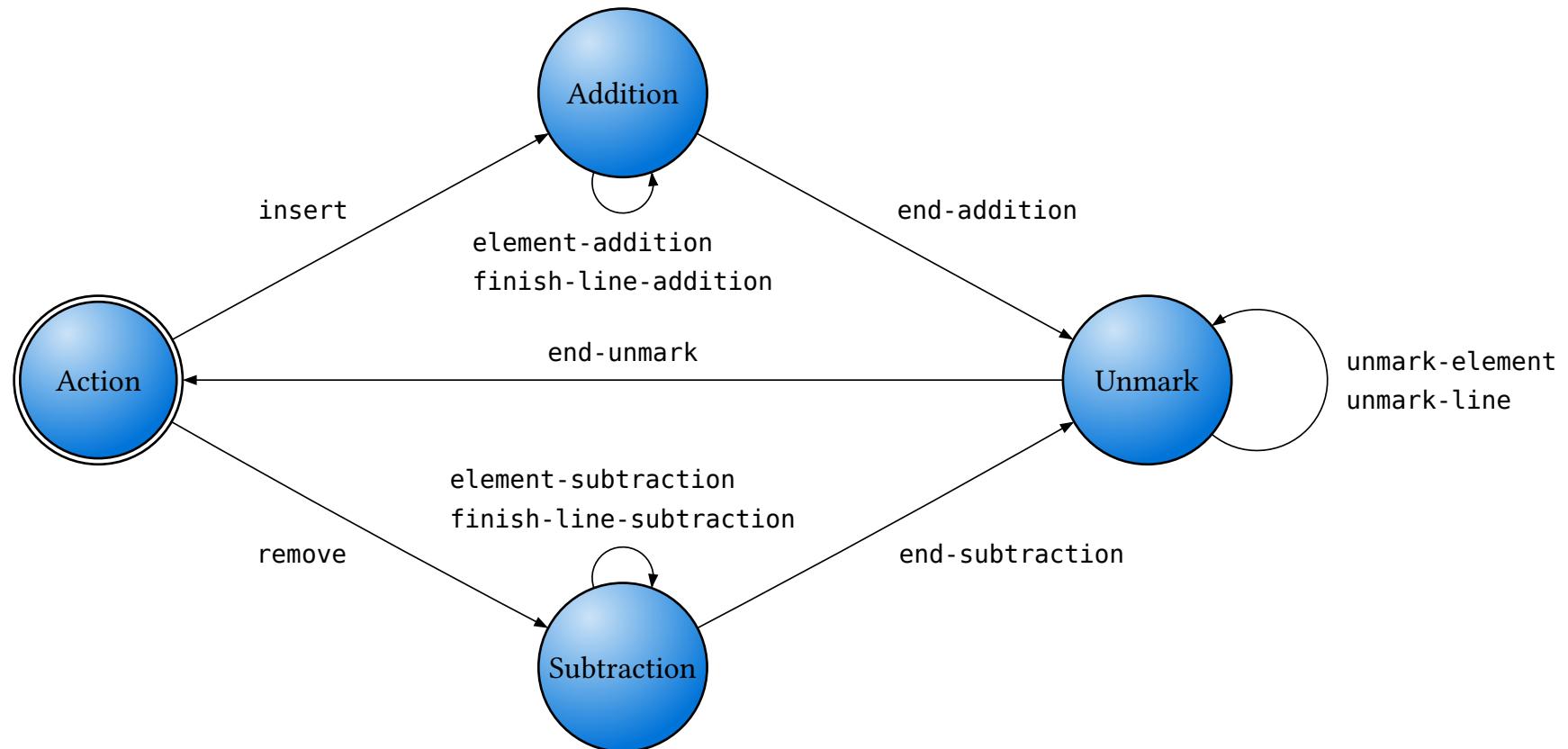


1.3 Complicaciones y soluciones

Solución: una máquina de estados

- Cada estado es un predicado de aridad cero.
- Se pueden colapsar el estado de activation y deactivation en uno solo, pero no mejora.
- En vez de la etapa de activación y desactivación decidimos utilizar **axiomas** o **derived**.
 - Da mejor rendimiento, del orden de pasar de días a segundos.
- No es necesario restringir las ranuras independientemente.
 - Se eliminan el estado *restrict* y *unrestrict*.

1.3 Complicaciones y soluciones



1.4 PDDL

- Tipos

- natural element - object
- quartz art category - object
- slot line orbament - object

- Axiomas / *Derived*

- (enough-power-for-element
?e - element ?l - line ?a - art)
- (line-active ?l - line ?a - art)
- (orbament-active ?o - orbament ?a - art)
- (any-active ?a - art)

- Predicados

- (addition ?lhs ?rhs ?res - natural)
- (less-than ?lhs ?rhs - natural)
- (inventory-count ?q - quartz ?n - natural)
- (power ?e - element ?q - quartz ?power - natural)
- (belongs ?q - quartz ?c - category)
- (requirement ?e - element ?a - art ?n - natural)
- (contains-slot ?o - orbament ?s - slot)
- (contains-line ?o - orbament ?l - line)
- (connects ?l - line ?s - slot)
- (quartz-element ?q - quartz ?e - element)
- (slot-can-hold ?s - slot ?e - element)
- (value ?e - element ?l - line ?n - natural)
- (contains-quartz ?s - slot ?q - quartz)
- (restricted ?o - orbament ?c - category)
- (filled ?s - slot)
- (quartz-to-be-modified ?q - quartz)
- (category-to-be-modified ?c - category)
- (slot-to-be-modified ?s - slot)
- (orbament-to-be-modified ?o - orbament)

Índice

1. Dominio

2. Problema

3. Pruebas y resultados

4. Conclusiones

2.1 Generación

- Definición compleja del dominio
 - Muchos objetos compartidos (naturales, orbamentos, cuarzos, artes, ...)
 - Muchos predicados base (`addition`, `less-than`).
 - Muchos predicados compartidos para definir los orbamentos, cuarzos, artes, restricciones...
- **Solución: metaprogramación de los problemas**
 - Problema definido en un JSON que se transforma a PDDL.
 - El JSON define los elementos y orbamentos permitidos, el estado inicial del inventario y los orbamentos, y las metas.
 - Durante la transformación, se añaden los objetos y predicados comunes a todos los problemas.

2.2 Pruebas y casos de uso

1. Insertar un cuarzo.
2. Activar todas las artes de un tipo dado.
3. Activar todas las artes.
4. Activar todas las artes con un solo orbamento.
5. Activar todas las artes, empezando con todos los orbamentos llenos.
6. Activar todas las artes, empezando con todos los orbamentos llenos y sin usar los mejores cuarzos.

Índice

1. Dominio

2. Problema

3. Pruebas y resultados

4. Conclusiones

3.1 Planificadores

- Fast-downward
 - Lama-first
 - Seq-sat-lama-2011
- Otros planificadores intentados
 - Fast-Forward
 - Odin (planner2)
 - SymDB (planner14)
 - DecStar-2023 (planner15)
 - Scorpion 2023 (planner25) Sat
 - Approximate Novelty (planner29)
 - Problema: no soportan axiomas, no compilan, no funcionan, requieren dependencias propietarias, o falla el wrapper de python.

3.2 Resultados

Tabla 1: Problema sencillo que pretende activar el arte de agua «Lágrima» (≥ 1). Los orbamentos comienzan vacíos, así que la solución óptima es única y es introducir un cuarzo que de al menos 1 en cualquier ranura de cualquier orbamento. En el inventario solo hay un cuarzo «PV 1» (1). seq-sat-lama-2011 a su vez explora todo el espacio de búsqueda (es diminuto) y demuestra que el plan es óptimo.

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	0.5896 "	0.55 "	0.0396 "	1	19	80	60.4 MB	13.9 MB
FD (seq-sat-lama-2011)	0.5891 "	0.52 "	0.0691 "	1	19	80	60.4 MB	15.3 MB

Tabla 2: Problema que intentan que se activen todas las artes de tipo espejismo (⌚), que como son dos, debería ser más sencillo que el resto de alternativas. Las artes son «Santificación» (≥ 2 ⌚, ≥ 1 ⚡, ≥ 1 💚, ≥ 1 🎪) y «Marca del caos (≥ 3 ⌚). En el inventario hay un cuarzo de cada tipo y los orbamentos empiezan vacíos.

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	5' 41.073 "	1' 26.42 "	4' 14.6526 "	2	38	5141	3359.7 MB	554.8 MB
FD (seq-sat-lama-2011)	14' 27.376 "	4' 06.54 "	10' 20.8363 "	2	38	5141	3366.7 MB	772.7 MB

3.2 Resultados

Tabla 3: El problema es similar al de la Tabla 2. En este caso con las mismas condiciones iniciales, se intenta activar todas las artes de tipo tierra (⼟), entre todos los personajes. En total hay ocho artes de tipo tierra distintas.

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	7' 14.649 "	1' 27.49 "	5' 47.1592 "	3	57	6873	3366.7 MB	557.6 MB
FD (seq-sat-lama-2011)	8' 33.468 "	1' 47.22 "	6' 46.2483 "	3	57	6873	3314.4 MB	775.8 MB

Tabla 4: El problema es similar al de la Tabla 2 y de la Tabla 3. En este caso con las artes de tipo agua (水), entre todos los personajes. En total hay once artes de tipo agua. Sin embargo, se necesitan menos cuarzos para activar que en las de tipo tierra (⼟).

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	5' 34.371 "	1' 27.03 "	4' 07.3415 "	2	38	5130	3323.4 MB	558.9 MB
FD (seq-sat-lama-2011)	13' 05.541 "	4' 35.51 "	8' 30.031 "	2	38	5130	3320.4 MB	751.9 MB

3.2 Resultados

Tabla 5: En este problema el objetivo es activar todas entre todos los personajes. Los orbamentos comienzan vacíos y hay cuatro objetos de cada en el inventario. Las dos filas de seq-sat-lama-2011 corresponden a la misma ejecución, pero a planes sucesivos que ha ido generando.

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	1 h 11' 22.43 "	4' 34.2 "	1 h 06' 48.2261 "	7	149	25157	3334.6 MB	575.2 MB
FD (seq-sat-lama-2011)	1 h 28' 47.38 "	5' 04.1 "	1 h 23' 43.2792 "	7	149	25157	3370.7 MB	769.2 MB
FD (seq-sat-lama-2011)	2 h 55' 44.9 "	5' 04.1 "	2 h 50' 40.7719 "	6	130	36451	3370.7 MB	769.2 MB

Tabla 6: En este problema se empieza con un elemento de cada en el inventario y un único orbamento vacío. El objetivo es activar **todas** las artes en un único personaje. Se puede ver que las soluciones subóptimas generar acciones de más, en este caso, de eliminación de cuarzos. Las dos filas de seq-sat-lama-2011 corresponden a la misma ejecución, pero a planes sucesivos que ha ido generando.

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	4 h 21' 33.8 "	1' 04.76 "	4 h 20' 29.0521 "	7	149	483661	902.1 MB	361.1 MB
FD (seq-sat-lama-2011)	4 h 14' 57.3 "	1' 18.76 "	4 h 13' 38.5653 "	7	149	483661	902.1 MB	417.1 MB
FD (seq-sat-lama-2011)	4 h 20' 07.9 "	1' 18.76 "	4 h 18' 49.136 "	4	92	17789	902.1 MB	417.1 MB

3.2 Resultados

Tabla 7: En este problema se tienen cuatro orígenes que empiezan completamente llenos con cuarzos. En el inventario hay cuatro de cada tipo de cuarzo excepto los mejores cuarzos (dan más potencia elemental) que solo hay uno de cada. El objetivo es el mismo, activar todas las artes a la vez.

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	6 h 54' 42.7"	3' 15.29"	6 h 51' 27.4487"	10	190	195359	3373.2 MB	654 MB
FD (seq-sat-lama-2011)	5 h 22' 43.5"	1' 49.27"	5 h 20' 54.2014"	10	190	195359	3367.2 MB	848.3 MB

Tabla 8: Este problema es el más difícil, las condiciones iniciales son las mismas que el de la Tabla 7, pero sin ninguno de los mejores cuarzos (antes uno de cada, ahora cero de cada). El objetivo es el mismo. En este caso se pasó del límite de tiempo impuesto de ocho horas.

* Una mención especial a que cuando estábamos trabajando con un dominio más relajado, donde no había restricciones de elemento por ranura, encontraba la solución de este problema en solo cuatro horas.

Planificador	Tiempo total	Tiempo de traducción	Tiempo de búsqueda	Coste del plan	Longitud del plan	Nodos expandidos	Memoria de traducción	Memoria de búsqueda
FD (lama-first)	8 h 02' 17.88"	2' 21.04"	7 h 59' 56.8"	∞	∞	219685	3063.2 MB	628.6 MB
FD (seq-sat-lama-2011)	8 h 01' 42.63"	1' 46.52"	7 h 59' 56.1"	∞	∞	219685	3060.2 MB	808.9 MB
FD (lama-first)*	4 h 00' 52.8"	1' 08.91"	3 h 59' 43.903"	16	304	622075	3067.6 MB	953.1 MB

Índice

1. Dominio

2. Problema

3. Pruebas y resultados

4. Conclusiones

4.1 Conclusiones

- Estado actual de los planificadores caótico
 - Ha sido necesario corregir errores de compilación en muchos de los planificadores probados.
 - Cada planificador soporta un subconjunto distinto de PDDL.
 - Algunos planificadores utilizan una sintaxis de PDDL un poco distinta.
- Son capaces de resolver problemas complejos de forma eficiente a pesar de ser independientes del dominio.
 - Se han podido encontrar planes con ~ 200 pasos en un periodo de tiempo razonable.
 - La mayor parte del consumo de memoria ocurre durante la traducción. La búsqueda utiliza muy poca memoria aún expandiendo muchos estados.