Lógica modal computacional

Indecidibilidad y modelos infinitos

Carlos Areces

1er cuatrimestre de 2012

Temario de hoy

- Hasta ahora estuvimos viendo cuál era la complejidad de algunas lógicas modales.
- En esta clase vamos a ver algunos ejemplos de lógicas modales *indecidibles*
- Mostraremos algunas técnicas para probar indecidibilidad, y otras propiedades relacionadas.

La lógica $\mathcal{HL}(\downarrow)$

- En la primera clase vimos que se podía pensar en una versión modal de un *binder*, el operador $\downarrow$
- Este operador nombra el punto actual de evaluación, y permite referirse a él en el resto de la fórmula
- Por ejemplo, $\downarrow x.\Diamond x$ caracteriza a los puntos reflexivos de un modelo.

La lógica $\mathcal{HL}(\downarrow)$

- En la primera clase vimos que se podía pensar en una versión modal de un *binder*, el operador $\downarrow$
- Este operador nombra el punto actual de evaluación, y permite referirse a él en el resto de la fórmula
- Por ejemplo, $\downarrow x.\Diamond x$ caracteriza a los puntos reflexivos de un modelo.

A la signatura que habíamos definido para la lógica modal básica le vamos a agregar un conjunto infinito numerable $\textsc{var}$ de variables. Dado entonces una signatura $\langle \textsc{prop}, \textsc{rel}, \textsc{var} \rangle$, la sintaxis de $\mathcal{HL}(\downarrow)$ es:

Sintaxis de $\mathcal{HL}(\downarrow)$

$$\varphi ::= x \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle r \rangle \varphi \mid \downarrow x.\varphi$$

donde $p \in \textsc{prop}$, $r \in \textsc{rel}$, $x \in \textsc{var}$.

La lógica $\mathcal{HL}(\downarrow)$

La semántica de esta lógica se define sobre los modelos de Kripke $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ usuales, más una *función de asignación* $g : \textsc{var} \rightarrow W$ que asigna variables a elementos del dominio.

La lógica $\mathcal{HL}(\downarrow)$

La semántica de esta lógica se define sobre los modelos de Kripke $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ usuales, más una *función de asignación* $g : \text{VAR} \to W$ que asigna variables a elementos del dominio.

- Dado un modelo $\mathcal{M} = \langle W, \{R_i\}, V\rangle$ y una asignación g, la semántica es:

Semántica de $\mathcal{HL}(\downarrow)$

$$
\begin{array}{rcl}
(\mathcal{M}, g), w \models p & \text{sii} & w \in V(p) \\
(\mathcal{M}, g), w \models x & \text{sii} & g(x) = w \\
(\mathcal{M}, g), w \models \neg\varphi & \text{sii} & (\mathcal{M}, g), w \not\models \varphi \\
(\mathcal{M}, g), w \models \varphi \wedge \psi & \text{sii} & (\mathcal{M}, g), w \models \varphi \text{ y } (\mathcal{M}, g), w \models \psi \\
(\mathcal{M}, g), w \models \langle r_i\rangle\varphi & \text{sii} & \text{existe un } w' \text{ tal que } wR_iw' \text{ y } (\mathcal{M}, g), w' \models \varphi \\
(\mathcal{M}, g), w \models \downarrow x.\varphi & \text{sii} & (\mathcal{M}, g^x_w), w \models \varphi \text{ donde } g^x_w \text{ es identica a } g \\
 & & \text{salvo que } g^x_w(x) = w.
\end{array}
$$

Modelo infinito

- Ya vimos que la lógica modal básica (y otras extensiones) tienen la propiedad de modelo finito
- Esto nos ayudó a probar la decidibilidad de estas lógicas (sabiendo además una cota para el tamaño del modelo)
- Vamos a ver que $\mathcal{HL}(\downarrow)$ es capaz de *forzar* un modelo infinito
- Esto no prueba por sí mismo indecidibilidad, pero es un indicador en esa dirección

Modelo infinito

- Informalmente, vamos a intentar escribir una fórmula que diga que existe un conjunto no vacío B de elementos que forman un orden parcial estricto. Esto es:
  - Irreflexivo
  - Transitivo
- Y en donde todo elemento tiene un sucesor

Esto implica que B es infinito.

Modelo infinito

- Informalmente, vamos a intentar escribir una fórmula que diga que existe un conjunto no vacío B de elementos que forman un orden parcial estricto. Esto es:
  - Irreflexivo
  - Transitivo
- Y en donde todo elemento tiene un sucesor

Esto implica que B es infinito.

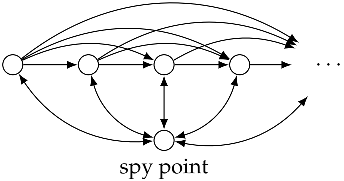
- Ahora, ¿cómo hablamos de un conjunto de puntos con una fórmula modal que se va a evaluar en un *único* punto?
- Vamos a usar una técnica que se conoce como *spy point*

Modelo infinito

- Informalmente, vamos a intentar escribir una fórmula que diga que existe un conjunto no vacío B de elementos que forman un orden parcial estricto. Esto es:
  - Irreflexivo
  - Transitivo
- Y en donde todo elemento tiene un sucesor

Esto implica que B es infinito.

- Ahora, ¿cómo hablamos de un conjunto de puntos con una fórmula modal que se va a evaluar en un *único* punto?
- Vamos a usar una técnica que se conoce como *spy point*



Forzando un modelo infinito

Cada s-sucesor tiene un eje hacia s

$(Back) \quad {\downarrow}s.([r]\neg s \wedge \langle r \rangle \top \wedge [r]\langle r \rangle s)$

Forzando un modelo infinito

Cada s-sucesor tiene un eje hacia s

$(Back) \quad \downarrow s.([r]\neg s \wedge \langle r\rangle\top \wedge [r]\langle r\rangle s)$

Los s-sucesores en dos pasos son s-sucesores en un paso

$(Spy) \quad \downarrow s.([r][r](\neg s \rightarrow (\downarrow x.\langle r\rangle(s \wedge \langle r\rangle x))))$

Forzando un modelo infinito

Cada s-sucesor tiene un eje hacia s

$(Back) \quad {\downarrow}s.([r]\neg s \wedge \langle r\rangle\top \wedge [r]\langle r\rangle s)$

Los s-sucesores en dos pasos son s-sucesores en un paso

$(Spy) \quad {\downarrow}s.([r][r](\neg s \to ({\downarrow}x.\langle r\rangle(s \wedge \langle r\rangle x))))$

La relación sobre los s-sucesores es irreflexiva

$(Irr) \quad [r]\neg({\downarrow}x.\langle r\rangle x)$

Forzando un modelo infinito

Cada s-sucesor tiene un eje hacia s

$(Back) \quad {\downarrow}s.([r]\neg s \wedge \langle r\rangle\top \wedge [r]\langle r\rangle s)$

Los s-sucesores en dos pasos son s-sucesores en un paso

$(Spy) \quad {\downarrow}s.([r][r](\neg s \rightarrow ({\downarrow}x.\langle r\rangle(s \wedge \langle r\rangle x))))$

La relación sobre los s-sucesores es irreflexiva

$(Irr) \quad [r]\neg({\downarrow}x.\langle r\rangle x)$

Todos los s-sucesores tienen un sucesor que no es s

$(Succ) \quad {\downarrow}s.([r]\langle r\rangle\neg s)$

Forzando un modelo infinito

Cada s-sucesor tiene un eje hacia s

$$(\mathit{Back}) \quad {\downarrow} s.([r]\neg s \wedge \langle r\rangle \top \wedge [r]\langle r\rangle s)$$

Los s-sucesores en dos pasos son s-sucesores en un paso

$$(\mathit{Spy}) \quad {\downarrow} s.([r][r](\neg s \rightarrow ({\downarrow} x.\langle r\rangle(s \wedge \langle r\rangle x))))$$

La relación sobre los s-sucesores es irreflexiva

$$(\mathit{Irr}) \quad [r]\neg({\downarrow} x.\langle r\rangle x)$$

Todos los s-sucesores tienen un sucesor que no es s

$$(\mathit{Succ}) \quad {\downarrow} s.([r]\langle r\rangle \neg s)$$

La relación sobre los s-sucesores es transitiva

$$(\mathit{Tran}) \quad {\downarrow} s.[r]{\downarrow} x.[r](\neg s \rightarrow [r](\neg s \rightarrow {\downarrow} z.\langle r\rangle(s \wedge \langle r\rangle(x \wedge \langle r\rangle z))))$$

Forzando un modelo infinito

Cada s-sucesor tiene un eje hacia s

$(\mathit{Back}) \quad \downarrow s.([r]\neg s \wedge \langle r\rangle\top \wedge [r]\langle r\rangle s)$

Los s-sucesores en dos pasos son s-sucesores en un paso

$(\mathit{Spy}) \quad \downarrow s.([r][r](\neg s \rightarrow (\downarrow x.\langle r\rangle(s \wedge \langle r\rangle x))))$

La relación sobre los s-sucesores es irreflexiva

$(\mathit{Irr}) \quad [r]\neg(\downarrow x.\langle r\rangle x)$

Todos los s-sucesores tienen un sucesor que no es s

$(\mathit{Succ}) \quad \downarrow s.([r]\langle r\rangle\neg s)$

La relación sobre los s-sucesores es transitiva

$(\mathit{Tran}) \quad \downarrow s.[r]\downarrow x.[r](\neg s \rightarrow [r](\neg s \rightarrow \downarrow z.\langle r\rangle(s \wedge \langle r\rangle(x \wedge \langle r\rangle z))))$

Sea la fórmula $\varphi = \mathit{Back} \wedge \mathit{Spy} \wedge \mathit{Irr} \wedge \mathit{Succ} \wedge \mathit{Tran}$

Forzando un modelo infinito

Teorema

Si $\mathcal{M}, w \models \varphi$ entonces $\mathcal{M}$ es infinito

Demostración. Por construcción de φ.

Forzando un modelo infinito

Teorema

Si $\mathcal{M}, w \models \varphi$ entonces $\mathcal{M}$ es infinito

Demostración. Por construcción de φ.

Es fácil ver que efectivamente hay modelos para φ

Teorema

Existe un modelo $\mathcal{M}$ y un punto $w \in \mathcal{M}$ tal que $\mathcal{M}, w \models \varphi$.

Demostración. Sea B un conjunto infinito de elementos y s un elemento tal que $s \notin B$. Sea R la mínima relación tal que

- R define un orden parcial estricto sobre B
- wRb y bRw para cada elemento $b \in B$

Luego el modelo $\mathcal{M} = \langle B \cup \{s\}, R, V\rangle$ (para cualquier V) verifica $\mathcal{M}, w \models \varphi$.

Indecidibilidad

- ¿Cómo mostramos que una lógica es indecidible?
- Si quisieramos mostrarlo de forma directa, deberíamos escribir una fórmula que codifique ejecuciones arbitrarias en una máquina de Turing
- El problema de *tiling*, que ya se demostró indecidible, nos va a ayudar para el caso modal

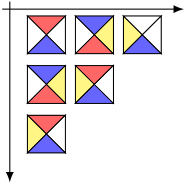
El problema de tiling

- Dado un conjunto finito de *tipos de tiles* $\mathcal{T}$

El problema de tiling

- Dado un conjunto finito de *tipos de tiles* $\mathcal{T}$

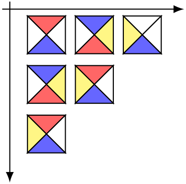
El *problema de tiling*: ¿Es posible colocar tiles de tipo $\mathcal{T}$ en $\mathbb{N} \times \mathbb{N}$ tal que cada par de tiles adyacentes tengan el mismo color?



El problema de tiling

- Dado un conjunto finito de *tipos de tiles* $\mathcal{T}$

El *problema de tiling*: ¿Es posible colocar tiles de tipo $\mathcal{T}$ en $\mathbb{N} \times \mathbb{N}$ tal que cada par de tiles adyacentes tengan el mismo color?

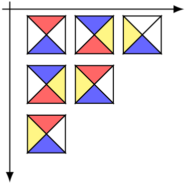


- El problema de tiling en $\mathbb{N} \times \mathbb{N}$ se sabe indecidible

El problema de tiling

- Dado un conjunto finito de *tipos de tiles* $\mathcal{T}$

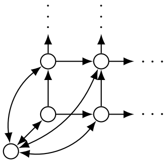
El *problema de tiling*: ¿Es posible colocar tiles de tipo $\mathcal{T}$ en $\mathbb{N} \times \mathbb{N}$ tal que cada par de tiles adyacentes tengan el mismo color?



- El problema de tiling en $\mathbb{N} \times \mathbb{N}$ se sabe indecidible
- Dado un conjunto de tipos de tiles $\mathcal{T}$, buscamos escribir una fórmula $\varphi_{\mathcal{T}}$ tal que $\varphi_{\mathcal{T}}$ es satisfacible sii hay un tiling para $\mathcal{T}$

El problema de tiling para $\mathcal{HL}(\downarrow)$

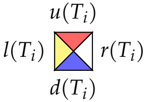
- La idea es intentar codificar este problema con una fórmula de $\mathcal{HL}(\downarrow)$
- Vamos a volver a usar la idea de definir un *spy point*



- Notar que, si bien puede parecer lo contrario, codificar el problema de tiling no implica forzar un modelo infinito (ni viceversa)

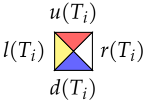
El problema de tiling para $\mathcal{HL}(\downarrow)$

- Sea $T = \{T_1, \ldots, T_n\}$ un conjunto de tipos de tiles
- Dado un tipo de tile T_i, $u(T_i)$, $r(T_i)$, $d(T_i)$, $l(T_i)$ van a representar los colores de T_i correspondientes a los lados arriba, derecha, abajo e izquierda.

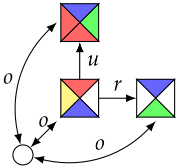


El problema de tiling para $\mathcal{HL}(\downarrow)$

- Sea $T = \{T_1, \ldots, T_n\}$ un conjunto de tipos de tiles
- Dado un tipo de tile T_i, $u(T_i)$, $r(T_i)$, $d(T_i)$, $l(T_i)$ van a representar los colores de T_i correspondientes a los lados arriba, derecha, abajo e izquierda.



- Supongamos también que tenemos una modalidad $\langle o \rangle$ que vamos a usar para movernos desde el spy point a cada tile
- Y modalidades $\langle u \rangle$ y $\langle r \rangle$ para movernos entre tiles hacia arriba y a la derecha respectivamente.



El problema de tiling para $\mathcal{HL}(\downarrow)$

Primero codificamos la grilla:

Cada s-sucesor tiene un eje hacia s

$(Back) \quad [o]\neg s \wedge \langle o \rangle \top \wedge [o]\langle o \rangle s$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Primero codificamos la grilla:

Cada s-sucesor tiene un eje hacia s

$$(Back) \quad [o]\neg s \wedge \langle o \rangle \top \wedge [o]\langle o \rangle s$$

El sucesor de un tile es un s-sucesor

$$(Spy) \quad [o][\dagger](\downarrow x.\langle o \rangle(s \wedge \langle o \rangle x)) \quad \dagger \in \{r, u\}$$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Primero codificamos la grilla:

Cada s-sucesor tiene un eje hacia s

$$(Back) \quad [o]\neg s \wedge \langle o \rangle \top \wedge [o]\langle o \rangle s$$

El sucesor de un tile es un s-sucesor

$$(Spy) \quad [o][\dagger](\downarrow x.\langle o \rangle(s \wedge \langle o \rangle x)) \quad \dagger \in \{r, u\}$$

Desde un tile no se llega a s por r y u

$$(Empty) \quad [o][\dagger]\neg s \quad \dagger \in \{r, u\}$$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Primero codificamos la grilla:

Cada s-sucesor tiene un eje hacia s

$(Back) \quad [o]\neg s \wedge \langle o \rangle \top \wedge [o]\langle o \rangle s$

El sucesor de un tile es un s-sucesor

$(Spy) \quad [o][\dagger](\downarrow x.\langle o \rangle (s \wedge \langle o \rangle x)) \quad \dagger \in \{r, u\}$

Desde un tile no se llega a s por r y u

$(Empty) \quad [o][\dagger]\neg s \quad \dagger \in \{r, u\}$

Todo tile tiene un tile arriba y a la derecha

$(Grid) \quad [o]\langle \dagger \rangle \top \quad \dagger \in \{r, u\}$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Primero codificamos la grilla:

Cada s-sucesor tiene un eje hacia s

$(Back) \quad [o]\neg s \wedge \langle o \rangle \top \wedge [o]\langle o \rangle s$

El sucesor de un tile es un s-sucesor

$(Spy) \quad [o][\dagger](\downarrow x.\langle o \rangle(s \wedge \langle o \rangle x)) \quad \dagger \in \{r, u\}$

Desde un tile no se llega a s por r y u

$(Empty) \quad [o][\dagger]\neg s \quad \dagger \in \{r, u\}$

Todo tile tiene un tile arriba y a la derecha

$(Grid) \quad [o]\langle \dagger \rangle \top \quad \dagger \in \{r, u\}$

Cada tile tiene un único tile arriba y un único tile a la derecha

$(Func) \quad [o]\downarrow x.([\dagger]\downarrow y.(\langle o \rangle \langle o \rangle (x \wedge [\dagger]y))) \quad \dagger \in \{r, u\}$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Primero codificamos la grilla:

Cada s-sucesor tiene un eje hacia s

$$(Back) \quad [o]\neg s \wedge \langle o\rangle\top \wedge [o]\langle o\rangle s$$

El sucesor de un tile es un s-sucesor

$$(Spy) \quad [o][\dagger](\downarrow x.\langle o\rangle(s \wedge \langle o\rangle x)) \quad \dagger \in \{r, u\}$$

Desde un tile no se llega a s por r y u

$$(Empty) \quad [o][\dagger]\neg s \quad \dagger \in \{r, u\}$$

Todo tile tiene un tile arriba y a la derecha

$$(Grid) \quad [o]\langle\dagger\rangle\top \quad \dagger \in \{r, u\}$$

Cada tile tiene un único tile arriba y un único tile a la derecha

$$(Func) \quad [o]\downarrow x.([\dagger]\downarrow y.(\langle o\rangle\langle o\rangle(x \wedge [\dagger]y))) \quad \dagger \in \{r, u\}$$

Hay confluencia entre arriba-derecha y derecha-arriba

$$(Conf) \quad [o]\downarrow x.(\langle u\rangle\langle r\rangle\downarrow y.(\langle o\rangle\langle o\rangle(x \wedge \langle r\rangle\langle u\rangle y)))$$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Y finalmente que en cada punto de la grilla hay un tile y que los colores coinciden:

Cada punto tiene un único tipo de tile

$$(\mathit{Unique}) \quad [o]\left(\bigvee_{1\leq i\leq n} t_i \wedge \bigwedge_{1\leq i<j\leq n}(t_i \rightarrow \neg t_j)\right)$$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Y finalmente que en cada punto de la grilla hay un tile y que los colores coinciden:

Cada punto tiene un único tipo de tile

$$(\mathit{Unique}) \quad [o]\left(\bigvee_{1\leq i\leq n} t_i \wedge \bigwedge_{1\leq i<j\leq n}(t_i \to \neg t_j)\right)$$

Cada tile tiene a la derecha un adyacente del color apropiado

$$(\mathit{Vert}) \quad [o]\bigwedge_{1\leq i\leq n}\left(t_i \to \langle u\rangle \bigvee_{1\leq j\leq n, u(T_i)=d(T_j)} t_j\right)$$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Y finalmente que en cada punto de la grilla hay un tile y que los colores coinciden:

Cada punto tiene un único tipo de tile

$$(Unique) \quad [o]\left(\bigvee_{1\leq i\leq n} t_i \wedge \bigwedge_{1\leq i<j\leq n}(t_i \rightarrow \neg t_j)\right)$$

Cada tile tiene a la derecha un adyacente del color apropiado

$$(Vert) \quad [o]\bigwedge_{1\leq i\leq n}\left(t_i \rightarrow \langle u\rangle \bigvee_{1\leq j\leq n, u(T_i)=d(T_j)} t_j\right)$$

Cada tile tiene a arriba un adyacente del color apropiado

$$(Horiz) \quad [o]\bigwedge_{1\leq i\leq n}\left(t_i \rightarrow \langle r\rangle \bigvee_{1\leq j\leq n, r(T_i)=l(T_j)} t_j\right)$$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Y finalmente que en cada punto de la grilla hay un tile y que los colores coinciden:

Cada punto tiene un único tipo de tile

$$(Unique) \quad [o]\left(\bigvee_{1\leq i\leq n} t_i \wedge \bigwedge_{1\leq i<j\leq n}(t_i \rightarrow \neg t_j)\right)$$

Cada tile tiene a la derecha un adyacente del color apropiado

$$(Vert) \quad [o]\bigwedge_{1\leq i\leq n}\left(t_i \rightarrow \langle u\rangle \bigvee_{1\leq j\leq n, u(T_i)=d(T_j)} t_j\right)$$

Cada tile tiene a arriba un adyacente del color apropiado

$$(Horiz) \quad [o]\bigwedge_{1\leq i\leq n}\left(t_i \rightarrow \langle r\rangle \bigvee_{1\leq j\leq n, r(T_i)=l(T_j)} t_j\right)$$

Sea

$$\varphi_T = \downarrow s.(Back \wedge Empty \wedge Spy \wedge Grid \wedge Func \wedge Conf \wedge Unique \wedge Vert \wedge Horiz)$$

El problema de tiling para $\mathcal{HL}(\downarrow)$

Teorema

Sea T un conjunto de tipos de tiles. Luego φ_T es satisfacible sii con T se pueder armar un tiling en $\mathbb{N} \times \mathbb{N}$.

El problema de tiling para $\mathcal{HL}(\downarrow)$

Teorema

Sea T un conjunto de tipos de tiles. Luego φ_T es satisfacible sii con T se pueder armar un tiling en $\mathbb{N} \times \mathbb{N}$.

Demostración:

($\Rightarrow$) Supongamos que $\mathcal{M}, w \models \varphi_T$. Por construcción, $\mathcal{M}$ representa un tiling de $\mathbb{N} \times \mathbb{N}$.

El problema de tiling para $\mathcal{HL}(\downarrow)$

Teorema

Sea T un conjunto de tipos de tiles. Luego φ_T es satisfacible sii con T se pueder armar un tiling en $\mathbb{N} \times \mathbb{N}$.

Demostración:

($\Rightarrow$) Supongamos que $\mathcal{M}, w \models \varphi_T$. Por construcción, $\mathcal{M}$ representa un tiling de $\mathbb{N} \times \mathbb{N}$.

($\Leftarrow$) Supongamos que $f : \mathbb{N} \times \mathbb{N} \to T$ es un tiling de $\mathbb{N} \times \mathbb{N}$. Definimos el modelo $\mathcal{M} = \langle W, \{R_o, R_u, R_r\}, V\rangle$:

- $W = \mathbb{N} \times \mathbb{N} \cup \{w\}$
- $R_o = \{(w, v), (v, w) \mid v \in \mathbb{N} \times \mathbb{N}\}$
- $R_u = \{(x, y), (x, y + 1) \mid x, y \in \mathbb{N}\}$
- $R_r = \{(x, y), (x + 1, y) \mid x, y \in \mathbb{N}\}$
- $V(t_i) = \{x \mid x \in \mathbb{N} \times \mathbb{N}, f(x) = T_i\}$

No es difícil ver que $\mathcal{M}, w \models \varphi_T$

Un tiling para cada necesidad

- Codificar el problema de tiling no sólo sirve para probar indecidibilidad

Un tiling para cada necesidad

- Codificar el problema de tiling no sólo sirve para probar indecidibilidad
- Variaciones de la definición de este problema también sirven para probar cotas de complejidad y otros grados de indecidibilidad. Por ejemplo:

Un tiling para cada necesidad

- Codificar el problema de tiling no sólo sirve para probar indecidibilidad
- Variaciones de la definición de este problema también sirven para probar cotas de complejidad y otros grados de indecidibilidad. Por ejemplo:
  - El problema de tiling que acabamos de ver es Π_1^0-completo

Un tiling para cada necesidad

- Codificar el problema de tiling no sólo sirve para probar indecidibilidad
- Variaciones de la definición de este problema también sirven para probar cotas de complejidad y otros grados de indecidibilidad. Por ejemplo:
  - El problema de tiling que acabamos de ver es Π^0_1-completo
  - Si distinguimos T_1, un tipo de tile en particular, el problema de tiling de $\mathbb{N} \times \mathbb{N}$ donde T_1 ocurre infinitamente seguido en la primera fila del tiling es Σ^1_1-completo

Un tiling para cada necesidad

- Codificar el problema de tiling no sólo sirve para probar indecidibilidad
- Variaciones de la definición de este problema también sirven para probar cotas de complejidad y otros grados de indecidibilidad. Por ejemplo:
  - El problema de tiling que acabamos de ver es Π^0_1-completo
  - Si distinguimos T_1, un tipo de tile en particular, el problema de tiling de $\mathbb{N} \times \mathbb{N}$ donde T_1 ocurre infinitamente seguido en la primera fila del tiling es Σ^1_1-completo
  - El "two person corridor tiling" es EXPTIME-completo

Two person corridor tiling

- Este tiling tiene 3 jugadores: Spoiler, Duplicator y un réferi.

Two person corridor tiling

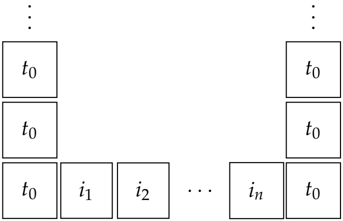
- Este tiling tiene 3 jugadores: Spoiler, Duplicator y un réferi.
- Del conjunto finito de tipos de tiles $T = \{t_0, t_1, \ldots, t_{s+1}\}$ vamos a distinguir t_0 y t_{s+1}

Two person corridor tiling

- Este tiling tiene 3 jugadores: Spoiler, Duplicator y un réferi.
- Del conjunto finito de tipos de tiles $T = \{t_0, t_1, \ldots, t_{s+1}\}$ vamos a distinguir t_0 y t_{s+1}
- Como parámetro de entrada también nos dan un $n \in \mathbb{N}$, que define el ancho del corredor

Two person corridor tiling

- Este tiling tiene 3 jugadores: Spoiler, Duplicator y un réferi.
- Del conjunto finito de tipos de tiles $T = \{t_0, t_1, \ldots, t_{s+1}\}$ vamos a distinguir t_0 y t_{s+1}
- Como parámetro de entrada también nos dan un $n \in \mathbb{N}$, que define el ancho del corredor
- El juego comienza con el réferi poniendo tiles de la siguiente manera



Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha

Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha
- Los jugadores no pueden elegir dónde poner un tile, sólo pueden elegir el tipo del tile

Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha
- Los jugadores no pueden elegir dónde poner un tile, sólo pueden elegir el tipo del tile
- Las reglas de coincidencia de color son las de siempre (e incluyen los “bordes” del corredor)

Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha
- Los jugadores no pueden elegir dónde poner un tile, sólo pueden elegir el tipo del tile
- Las reglas de coincidencia de color son las de siempre (e incluyen los “bordes” del corredor)
- Los jugadores juegan alternativamente, y Spoiler juega primero.

Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha
- Los jugadores no pueden elegir dónde poner un tile, sólo pueden elegir el tipo del tile
- Las reglas de coincidencia de color son las de siempre (e incluyen los "bordes" del corredor)
- Los jugadores juegan alternativamente, y Spoiler juega primero.
- Cuándo los jugadores ganan o pierden?

Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha
- Los jugadores no pueden elegir dónde poner un tile, sólo pueden elegir el tipo del tile
- Las reglas de coincidencia de color son las de siempre (e incluyen los “bordes” del corredor)
- Los jugadores juegan alternativamente, y Spoiler juega primero.
- Cuándo los jugadores ganan o pierden?
  - Si después de finitas jugadas un tile de tipo t_{s+1} es puesto en la primera columna, Spoiler gana

Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha
- Los jugadores no pueden elegir dónde poner un tile, sólo pueden elegir el tipo del tile
- Las reglas de coincidencia de color son las de siempre (e incluyen los "bordes" del corredor)
- Los jugadores juegan alternativamente, y Spoiler juega primero.
- Cuándo los jugadores ganan o pierden?
  - Si después de finitas jugadas un tile de tipo t_{s+1} es puesto en la primera columna, Spoiler gana
  - En otro caso (si alguno de los jugadores no puede hacer ningún movimiento válido, t_{s+1} no está en la columna 1, o el juego se prolonga infinitamente) gana Duplicator

Two person corridor tiling

- Las reglas para ir llenando el corredor son estrictas: desde abajo y de izquierda a derecha
- Los jugadores no pueden elegir dónde poner un tile, sólo pueden elegir el tipo del tile
- Las reglas de coincidencia de color son las de siempre (e incluyen los "bordes" del corredor)
- Los jugadores juegan alternativamente, y Spoiler juega primero.
- Cuándo los jugadores ganan o pierden?
  - Si después de finitas jugadas un tile de tipo t_{s+1} es puesto en la primera columna, Spoiler gana
  - En otro caso (si alguno de los jugadores no puede hacer ningún movimiento válido, t_{s+1} no está en la columna 1, o el juego se prolonga infinitamente) gana Duplicator
- El problema de determinar si Spoiler tiene una estrategia ganadora se sabe EXPTIME-completo.

Two person corridor tiling

- Usando el “two person corridor tiling” se puede demostrar que PDL es EXPTIME-hard, codificando el árbol de posibles jugadas entre Spoiler y Duplicator.
- También se puede usar para demostrar que $K + A$ es EXPTIME-hard.